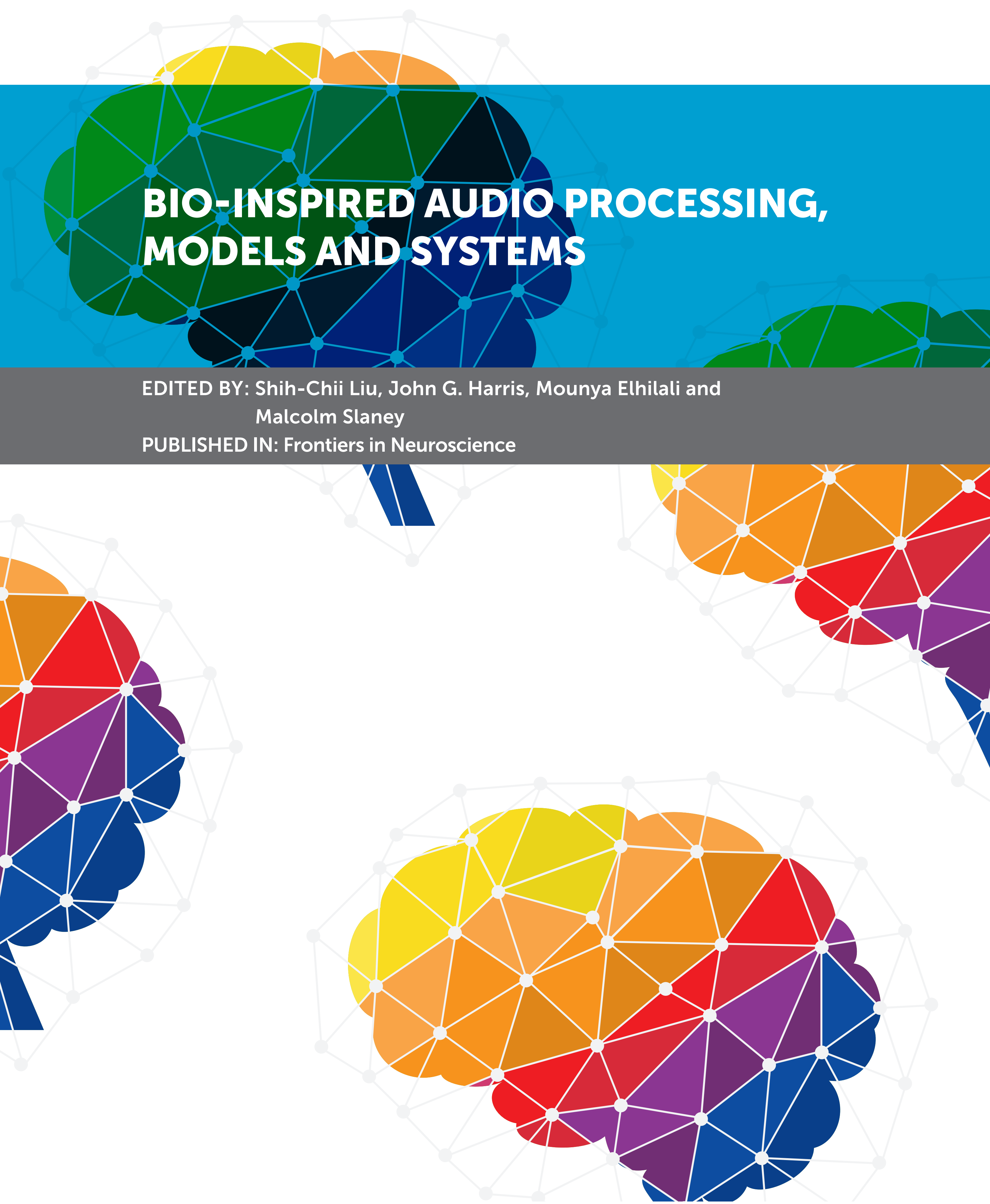


BIO-INSPIRED AUDIO PROCESSING, MODELS AND SYSTEMS

EDITED BY: Shih-Chii Liu, John G. Harris, Mounya Elhilali and
Malcolm Slaney

PUBLISHED IN: Frontiers in Neuroscience

frontiers Research Topics

ISSN 1664-8714
ISBN 978-2-88963-232-9
DOI 10.3389/978-2-88963-232-9

About Frontiers

Frontiers is more than just an open-access publisher of scholarly articles: it is a pioneering approach to the world of academia, radically improving the way scholarly research is managed. The grand vision of Frontiers is a world where all people have an equal opportunity to seek, share and generate knowledge. Frontiers provides immediate and permanent online open access to all its publications, but this alone is not enough to realize our grand goals.

Frontiers Journal Series

The Frontiers Journal Series is a multi-tier and interdisciplinary set of open-access, online journals, promising a paradigm shift from the current review, selection and dissemination processes in academic publishing. All Frontiers journals are driven by researchers for researchers; therefore, they constitute a service to the scholarly community. At the same time, the Frontiers Journal Series operates on a revolutionary invention, the tiered publishing system, initially addressing specific communities of scholars, and gradually climbing up to broader public understanding, thus serving the interests of the lay society, too.

Dedication to Quality

Each Frontiers article is a landmark of the highest quality, thanks to genuinely collaborative interactions between authors and review editors, who include some of the world's best academicians. Research must be certified by peers before entering a stream of knowledge that may eventually reach the public - and shape society; therefore, Frontiers only applies the most rigorous and unbiased reviews.
Frontiers revolutionizes research publishing by freely delivering the most outstanding research, evaluated with no bias from both the academic and social point of view. By applying the most advanced information technologies, Frontiers is catapulting scholarly publishing into a new generation.

What are Frontiers Research Topics?

Frontiers Research Topics are very popular trademarks of the Frontiers Journals Series: they are collections of at least ten articles, all centered on a particular subject. With their unique mix of varied contributions from Original Research to Review Articles, Frontiers Research Topics unify the most influential researchers, the latest key findings and historical advances in a hot research area! Find out more on how to host your own Frontiers Research Topic or contribute to one as an author by contacting the Frontiers Editorial Office: researchtopics@frontiersin.org

BIO-INSPIRED AUDIO PROCESSING, MODELS AND SYSTEMS

Topic Editors:
Shih-Chii Liu, University of Zurich and ETH Zurich, Switzerland
John G. Harris, University of Florida, United States
Mounya Elhilali, Johns Hopkins University, United States
Malcolm Slaney, Google, United States

Neurophysiology and biology provide useful starting points to help us understand and build better audio processing systems. The papers in this special issue address hardware implementations, spiking networks, sound identification, and attention decoding.

Citation: Liu, S.-C., Harris, J. G., Elhilali, M., Slaney, M., eds. (2019). Bio-inspired Audio Processing, Models and Systems. Lausanne: Frontiers Media SA.
doi: 10.3389/978-2-88963-232-9

Table of Contents

HARDWARE IMPLEMENTATIONS

SPIKING NETWORKS

SOUND IDENTIFICATION

ATTENTION DECODING

frontiers in Neuroscience

EDITORIAL
published: 13 September 2019
doi: 10.3389/fnins.2019.00978

Editorial: Bio-inspired Audio Processing, Models and Systems

*Shih-Chii Liu[1], John G. Harris[2], Mounya Elhilali[3] and Malcolm Slaney[4]**

[1] Institute of Neuroinformatics, University of Zurich and ETH Zurich, Zurich, Switzerland, [2] Department of Electrical & Computer Engineering, University of Florida, Gainesville, FL, United States, [3] Department of Electrical and Computer Engineering, Whiting School of Engineering, Johns Hopkins University, Baltimore, MD, United States, [4] Google, Mountain View, CA, United States

Keywords: neuromorphic audio sensors, spatial hearing, auditory perception, inter-aural time delay, spike-based computation, spatial-temporal response fields, attention decoding, saliency

Editorial on the Research Topic

Bio-inspired Audio Processing, Models and Systems

OPEN ACCESS

Edited by:
Timothy K. Horiuchi,
University of Maryland, College Park,
United States

Reviewed by:
Soumyajit Mandal,
Case Western Reserve University,
United States
Chetan Singh Thakur,
Indian Institute of Science (IISc), India

****Correspondence:***
Malcolm Slaney
malcolm@ieee.org

Specialty section:
This article was submitted to
Neuromorphic Engineering,
a section of the journal
Frontiers in Neuroscience

Received: *02 June 2019*
Accepted: *30 August 2019*
Published: *13 September 2019*

Citation:
Liu S-C, Harris JG, Elhilali M and Slaney M (2019) Editorial: Bio-inspired Audio Processing, Models and Systems. Front. Neurosci. 13:978.
doi: 10.3389/fnins.2019.00978

INTRODUCTION

Bio-inspired systems look at biology to inspire engineering solutions that help explain, emulate and complement the intricate processes that take place in a biological system. As such, they operate at the intersection of biology and engineering and leverage advantages from both disciplines. When applied to brain sciences, bio-inspired systems often use non-conventional approaches to solve complex sensory and cognitive tasks.

Recent developments in sensor design, algorithmic configurations, and network-level processing show the promise and efficacy of brain-like systems in solving complex tasks. While vision systems are widely explored in neuromorphic engineering design, audio systems offer unique challenges. These include careful handling of the time and space dimensions, issues related to temporal sampling and signal representation in both time and frequency, leveraging the redundancy in audio signals for complex detection and recognition tasks, as well as robust processing against noise and other interferers and maskers.

Our auditory systems have evolved highly efficient solutions to audio scene analysis, spatial understanding, and sound recognition. We wish to better understand the biological solutions that allow the brain to process sounds in unknown and highly distorted conditions; in order to help advance state-of-art audio systems that often operate well under well-controlled environments but fail to generalize, adapt and efficiently process unknown conditions. Furthermore, we want to apply engineering methods to better understand biological processes, using non-invasive methods. By leveraging both our knowledge of the biology in building better systems, as well as new technological advantages to unravel secrets of the brain, we hope to enrich the conversation across both disciplines in order to advance our understanding of the brain function and help improve technologies that impact our lives in a wide range of domains.

OVERVIEW

This special topic issue describes the latest advances in research on sensors, models, networks, and hardware for audio processing, hearing systems, and speech technologies. Broadly speaking, the papers in this special issue fall into four broad classes:

1. Bio-inspired implementations
2. Models based on spikes
3. Sound recognition
4. Attention decoding.

Bio-inspired systems often start with hardware designed to mimic and/or capitalize on the advantages of biological systems. With regards to processing acoustic cues, a paper by Xu et al. describes a digital hardware FPGA implementation of a well-known CAR-FAC cochlear model that mimics the auditory physiology seen in the biological cochlea. Similarly, our auditory system is exquisitely sensitive to the differences in signals received between the two ears. The paper by Isbell and Horiuchi explores how the auditory system might change the timing of pulses in an echo-location system. Finally a paper by Encke and Hemmert introduces a spiking neuron model based on recent physiological findings in mammals for the detection of interaural time differences for sound localization.

The most obvious difference between conventional solutions to auditory processing and biological systems is the way that our biology depends on discrete spikes to represent the sensory signal. Toward this end, papers by Anumula et al. and Acharya et al. investigate different ways to represent the spiking information in ways amenable to conventional machine-learning methods. The paper by Wu et al. takes these approaches to feature discovery a step further by using a self-organizing network to design the best feature representation. Then, the paper by Li and Príncipe looks at ways to extend the temporal information using kernel methods that can choose the optimal representation.

An important task for the auditory system is to understand and identify the sounds around us. The paper by McWalter and Dau considers high-level features that combine information across time and frequency for synthesizing and perceiving auditory textures. A paper by Zuk et al. looks at how we perceive musical beats, comparing the information from bottom-up (sensory) processes vs. top-down (cognitive) expectations. Finally a paper by Huang et al. looks at ways to build models of what makes a sound salient in its environment.

To conclude this special issue, much effort recently has gone toward finding methods that allow us to monitor the attention of a user. In the visual world, the eyes provide an important clue, but no such obvious signal exists for the auditory world. The paper by Alickovic et al. summarizes several approaches based on regression and correlation analysis that allow us to match the audio signal and brain's response. Wong et al.'s paper adds further details on regularization methods for regression-based methods, which are needed to make the computations stable. To put it all together, a paper by Miran et al. builds an end-to-end solution that considers the statistics of the input signal and the output decision to build an optimal decoder of a user's attentional state.

We hope you find these 13 papers illuminating. They represent the state of the art in bio-inspired audio-processing models and systems.

AUTHOR CONTRIBUTIONS

This editorial was written and edited by MS, S-CL, ME, and JH.

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Neuroscience

ORIGINAL RESEARCH
published: 10 April 2018
doi: 10.3389/fnins.2018.00198

A FPGA Implementation of the CAR-FAC Cochlear Model

*Ying Xu, Chetan S. Thakur†, Ram K. Singh, Tara Julia Hamilton, Runchun M. Wang and André van Schaik**

MARCS Institute, Western Sydney University, Sydney, NSW, Australia

This paper presents a digital implementation of the Cascade of Asymmetric Resonators with Fast-Acting Compression (CAR-FAC) cochlear model. The CAR part simulates the basilar membrane's (BM) response to sound. The FAC part models the outer hair cell (OHC), the inner hair cell (IHC), and the medial olivocochlear efferent system functions. The FAC feeds back to the CAR by moving the poles and zeros of the CAR resonators automatically. We have implemented a 70-section, 44.1 kHz sampling rate CAR-FAC system on an Altera Cyclone V Field Programmable Gate Array (FPGA) with 18% ALM utilization by using time-multiplexing and pipeline parallelizing techniques and present measurement results here. The fully digital reconfigurable CAR-FAC system is stable, scalable, easy to use, and provides an excellent input stage to more complex machine hearing tasks such as sound localization, sound segregation, speech recognition, and so on.

Keywords: neuromorphic engineering, electronic cochlea, basilar membrane, inner hair cell, outer hair cell, automatic gain control, medial olivocochlear efferent, FPGAs

OPEN ACCESS

Edited by:
Malcolm Slaney,
Google, United States

Reviewed by:
Guillaume Garreau,
IBM Research Almaden, United States
Amin Saremi,
University of Oldenburg, Germany

****Correspondence:***
André van Schaik
a.vanschaik@westernsydney.edu.au

†Present Address:
Chetan S. Thakur,
Department of Electronic Systems Engineering, Indian Institute of Science, Bangalore, India

Specialty section:
This article was submitted to Neuromorphic Engineering, a section of the journal Frontiers in Neuroscience

Received: *31 October 2017*
Accepted: *12 March 2018*
Published: *10 April 2018*

Citation:
Xu Y, Thakur CS, Singh RK, Hamilton TJ, Wang RM and van Schaik A (2018) A FPGA Implementation of the CAR-FAC Cochlear Model. Front. Neurosci. 12:198. doi: 10.3389/fnins.2018.00198

INTRODUCTION

The human auditory system is superior to any machine-hearing system in efficiency of perceiving sound. As the input structure for the auditory pathway, the tonotopically-organized cochlea decomposes, converts and amplifies sound waves nonlinearly into electrical signals, and delivers the results to the nervous system. The cochlea is characterized by a remarkably wide dynamic range (0-120 dB SPL) (Fettiplace and Hackney, 2006), and a high frequency selectivity (∼3 Hz at the characteristic frequency of 1 kHz; Glasberg and Moore, 1990). Over the past decades, efforts have been made to engineer a hearing machine that is able to emulate the function and efficiency of the human auditory system. As a first step toward this target, cochlear models have been proposed, developed, and implemented in a number of ways with a varying degree of complexities.

Auditory Filter Models

Cochlear models can be divided into two classes: transmission-lines (TL) and auditory filterbanks (Duifhuis, 2004). The TL models represent the cochlea partition as a coupled mass-spring-damper system to model wave propagation on the Basilar Membrane (BM) (Zweig et al., 1976). TL models are faithful to the physiology and are accurate in simulating wave propagation on the BM. However, they are more computationally challenging as they have complicated differential equations in the time domain (Altoè and Pulkki, 2014).

Auditory filterbank models use either parallel or cascade filters to model wave propagation on the BM. Parallel filterbank models use independent filters, such as rounded-exponential (roex) filters (Glasberg et al., 1984), the gammatone filter family (including gammachirp; Patterson et al., 2003), or pole-zero filters (Lyon et al., 2010), that connect to a single input signal in parallel. Cascade filterbank models, for example the CAR-FAC model (Lyon, 2017) or biophysical models of (Liu and Neely, 2010; Saremi and Stenfelt, 2013), use a cascade of filters instead.

Parallel filterbank models are mostly concerned with reproducing the observed mechanical and pay little attention to the biological structure of the cochlea. For example, Wang et al. implemented a parallel ultra-steep roll-off filter model on a 0.35μm CMOS chip (Wang et al., 2015), and Yang et al. implemented a parallel source-follower-based bandpass filterbank on a 0.18 μm CMOS analog IC (Yang et al., 2016). Some parallel filterbank models include an automatic gain control (AGC) mechanism to model some couplings between channels. For example, Yang et al. implemented a parallel filter bank of 4th-order one-zero gammatone filters (OZGF) with across channels AGC on a 0.35 μm CMOS chip (Yang et al., 2015). Another parallel form, the 2-D parallel filterbank, models the fluid within the cochlear duct as well as the BM taking both the longitudinal and vertical wave propagation into account. Examples of silicon cochleae of 2-D models include (van Schaik and Fragniere, 2001; Hamilton et al., 2008; Nouri et al., 2015).

Cascade filterbank models take advantage of the way sound propagates in the forward direction as traveling waves in the cochlea. In the cascade of filters, each filter stage models a segment of the nonuniform distributed wave system and its output becomes the input of the next section (Lyon, 1998). The cascade form thus provides a natural model of coupling in the forward direction. For example, Chan et al. implemented a 2nd-order low pass filter with address event interface (Chan et al., 2007), Liu et al. implemented a cascade 64-stage model on a 0.35μm CMOS chip (Liu et al., 2014), Thakur et al. implemented a CAR model on a FPGA (Thakur et al., 2014), and Jimenez-Fernandez et al. implemented a cascade spike band pass filer model on a FPGA (Jimenez-Fernandez et al., 2016). For some cascade filterbank models, such as Lyon's pole-zero filter cascade (PZFC) model and CAR-FAC model, an AGC feedback loop is included to model some couplings between channels in both directions. We describe the hardware implementation of the CAR-FAC model in this paper.

Cochlea Nonlinearity

The biological cochlea is a causal, active, and nonlinear system. **Figure 1** shows the nonlinearity and frequency tuning measured from a biological cochlea for various sound pressure levels measured in dB SPL adapted from (Ruggero, 1992). The gain is measured by the BM displacement (or velocity) relative to the stapes motion. In the biological cochlea, responses at frequencies near the characteristic frequency (CF) (9 kHz) vary nonlinearly with input level. Additionally, the responses show steeper high-frequency roll-off slope at lower SPLs, and the peak gain shifts toward lower frequencies with increasing input level.

In auditory filterbank models, the nonlinearities can be described as linear filters with parameters depending on signal level. For example, the parallel and cascade gammachirp filter models (PrlGC and CasGC) (Irino and Patterson, 2001; Unoki et al., 2006), the all-pole gammatone filter (APGF) models and PZFC models (Lyon, 1997; Katsiamis et al., 2007) show a forward compressive nonlinear response via the movement of the poles and/or zeros. For AGC-based models, the output level is fed back to modify filter parameters, to result in a compressive input-output function (Lyon, 2011). Such a feedback nonlinearity mechanism is inspired by the OHCs function of the mammalian cochlea (Kim, 1986). The PZFC analog cochlear model (Lyon and Mead, 1988) and the CAR-FAC model (Lyon, 2017) are such examples.

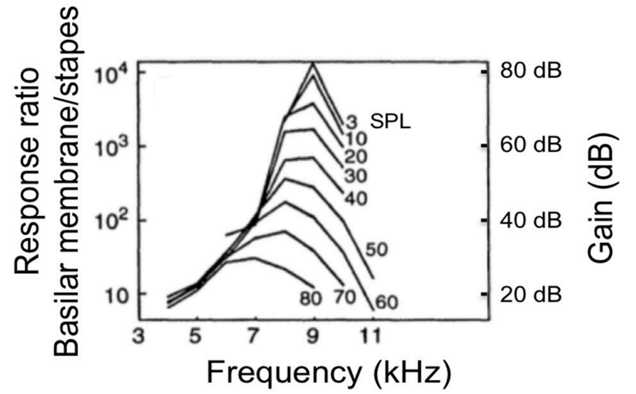


FIGURE 1 | The frequency response measured from a chinchilla cochlea for various levels input strength measured in dB of sound pressure level (SPL) adapted from (Ruggero, 1992). The gain is measured by the BM displacement (or velocity) relative to the stapes motion.

Motivations

The CAR-FAC model is a digital cascade auditory filter model proposed by Richard Lyon and described in detail in (Lyon, 2017). It closely approximates the physiological elements that consist of the human cochlea and mimics its qualitative behavior. The CAR part models the BM function that translates the cochlear fluid pressure wave (converted from the sound wave by the middle ear) into positions of maximal displacement along its length. Its pole-zero cascade form uses fewer parameters in the z domain than other filters, such as the gammatone and the gammachrip filters (impulse response) to provide an excellent fit to data on human detection tones in masking noise (Lyon, 2011). The FAC part models the OHC, the IHC and the medial olivocochlear efferent system functions that transduce the cochlear mechanic vibrations into electronic signals and exert a nonlinear gain control feedback on the BM through the OHC. The FAC nonlinear effects include a fast wide-dynamic-range compression and frequency distortions such as cubic difference tones (CDTs) and quadratic difference tones (QDTs) and are realized by moving the positions of the poles and zeros of the CAR resonators in the z plane.

Saremi et al. compared seven computational cochlear models including one cascade filterbank model (CAR-FAC), one transmission-line model, one biophysical model, and four parallel filterbank models (Saremi et al., 2016) in response to a set of common stimuli, which are used in the clinical assessment of human hearing to study their performance. The results show that the CAR-FAC exhibits an outstanding agreement with the biological data recordings at a reasonably low computational cost. These factors formed our basis of developing the CAR-FAC model and investigating its characteristics and possible applications.

We target a digital ASIC implementation of the CAR-FAC model for machine hearing applications since it is small, more energy efficient and more stable than analog implementations (Sarpeshkar, 2006). For the validation and prototype stage, we choose to implement it on a small FPGA board, the Altera Cyclone V starter kit. We previously introduced the CAR-FAC system on FPGA in (Xu et al., 2016), and here we present the complete system and measurement results.

MATERIALS AND METHODS

The CAR-FAC Model

The CAR-FAC model consists of a cascade of asymmetric resonators, a digital OHC (DOHC) model, a digital IHC (DIHC) model and an AGC loop, as shown in **Figure 2.** At each stage, the resonator H_i is connected to its next stage and the DIHC. It also gives an intermediate variable, velocity, to the DOHC. The DIHC feeds back to the DOHC through the AGC loop. The DOHC combines the AGC loop output and the velocity and feeds back to the resonator. The CAR-FAC output includes a multi-channel BM out y_i and a DIHC out, which can be transformed into the neural activity patterns r_i. The details of each model are described hereafter:

CAR

In the CAR, the asymmetric resonator is a coupled form two-pole-two-zero filter, as shown in **Figure 3**. The transfer function of the filter in the z domain is:

$$
\begin{aligned}
H(z) &= \frac{Y}{X} = g\left[\frac{(z - z_{zero})\left(z - z_{zero}^{*}\right)}{\left(z - z_{pole}\right)\left(z - z_{pole}^{*}\right)}\right] \\
&= g\left[\frac{z^2 + \left(-2a_0 + hc_0\right) rz + r^2}{z^2 - 2a_0 rz + r^2}\right] \qquad (1)
\end{aligned}
$$

The two-pole coupled form has a pair of conjugate poles (z_{pole} *and* z_{pole}^{*}):

$$
\begin{aligned}
z_{pole}, z_{pole}^{*} &= \frac{2a_0 r \pm \sqrt{(2a_0 r)^2 - 4r^2}}{2} \qquad (2)\\
&= rcos\left(\theta_R\right) \pm irsin\left(\theta_R\right) \\
a_0 &= \cos\left(\theta_R\right) \qquad (3)
\end{aligned}
$$

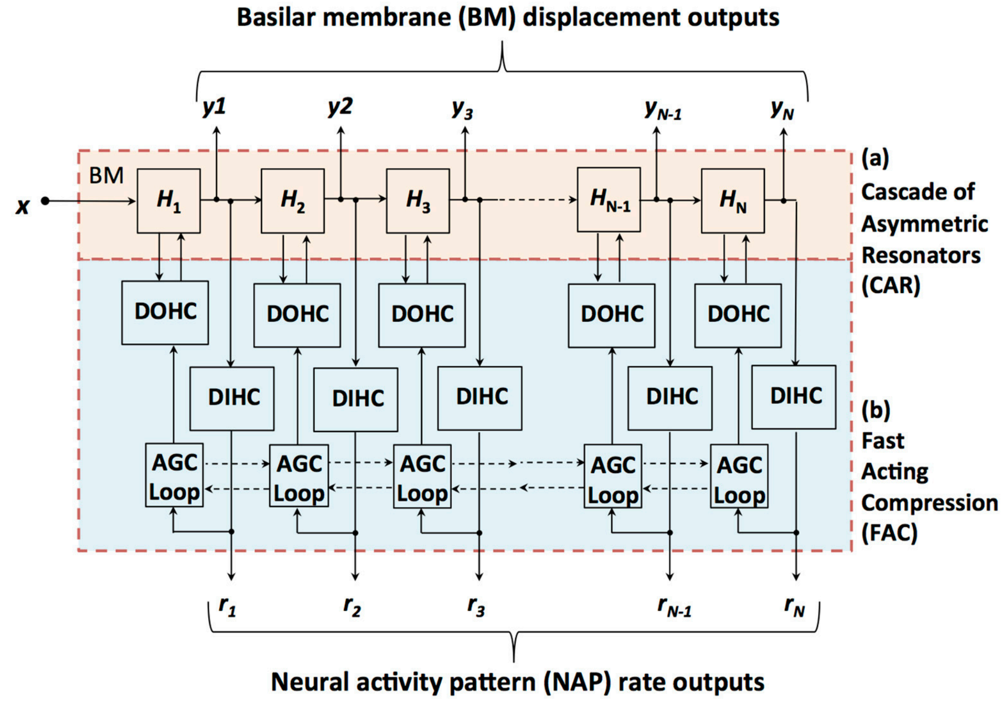


FIGURE 2 | Structure of the CAR-FAC model. x is the input sound, H_1 to H_N are the transfer functions of the CAR part, and y_1 to y_N represent the CAR-FAC output. The CFs of the CAR resonators decrease from left to right. The DOHC, the DIHC and the AGC loop comprise the FAC part. The neural activity pattern (NAP) rate outputs, r_1 to r_N, are estimations of average instantaneous nerve firing rates.

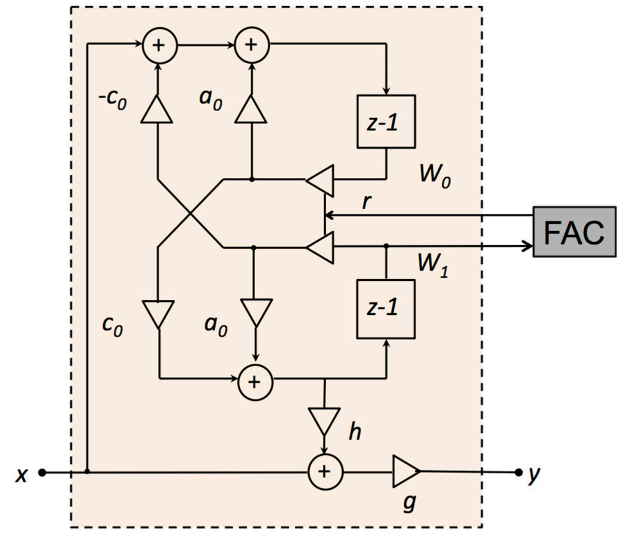


FIGURE 3 | Structure of the two-pole-two-zero resonator. a_0, c_0, and h are the resonator coefficients, r is the pole/zero radius in the z plane, g is the DC gain factor, W_0 and W_1 are the intermediate variables, x is the input, and y is the output.

where θ_R is the pole angle in the z plane. The conjugate zeros (z_{zero} and z^*_{zero}) are:

$$z_{zero},\ z^*_{zero} = \frac{-\left(-2a_0 + hc_0\right) r \pm \sqrt{\left(\left(-2a_0 + hc_0\right) r\right)^2 - 4r^2}}{2} \tag{4}$$

$$= rcos\left(\theta_z\right) \pm irsin\left(\theta_z\right)$$

$$a_0 - hc_0/2 = \cos\left(\theta_Z\right) \tag{5}$$

where θ_Z is the zero angle in the z plane. The zero radius is the same as the pole radius, r. The condition for complex zeros becomes relevant for high-frequency channels, where $\cos\left(\theta_R\right) < 0$:

$$a_0 - \frac{hc_0}{2} > -1 \tag{6}$$

$$h < \frac{2 + 2a_0}{c_0} \tag{7}$$

Coefficient g controls the stage DC gain. Here, g is set to maintain a unit DC gain for each stage of the filterbank:

$$g = \frac{1}{H(1)} = \frac{1 - 2a_0 r + r^2}{1 - 2\left(a_0 - hc_0\right) r + r^2} \tag{8}$$

In this structure, the zeros can be moved together with the poles by changing r while keeping h constant. The two zeros are placed slightly above the poles in frequency, and the distance between the zeros and the poles are set by the coefficient h. For lower h, the zeros are close to the poles, forming a steeper roll-off (asymmetric). For higher h, the zeros are further away from the poles, which results in a gradual roll-off at the higher frequency end. The steeper roll-off fits the auditory filtering characteristic and provides better frequency selectivity. Here, h is set to c_0 to keep the zero frequency at half an octave above the pole frequency.

Additionally, changing the poles and the zeros of the filter, via r leaves the zero-crossing times of the filter's impulse response nearly unchanged in time. The unchanged zero crossing characteristic satisfies the physiologically observed condition that the impulse response zero crossings are very nearly unchanged with variation in stimulus level (Lyon, 2017).

The zeros and poles are set initially for each cascade stage. The poles of the two-pole-two-zero resonator are chosen to be equally spaced along the normalized length of the cochlea according to the Greenwood map function (Greenwood, 1990):

$$f = 165.4(10^{2.1x} - 1) \tag{9}$$

Here, coefficient x is the normalized position along the cochlea, varying from 0 at the apex of the BM, to 1 at the basal end, and coefficient f is the pole frequency.

In the CAR-FAC model, the FAC effects are achieved by moving the initial CAR poles and zeros positions by varying their radius r. The details of each element in the FAC part are presented in the next three sections.

DOHC

The DOHC models the OHCs function, actively and nonlinearly amplifying the wave propagation in the cochlea. In the CAR-FAC model, the DOHC gain control mechanism integrates a local instantaneous nonlinearity and a multi-time-scale nonlinearity, as shown in **Figure 4**. The instantaneous nonlinearity is based on the BM velocity, taken as the rate of change of W_1. The multi-time-scale nonlinearity comes from the DIHC feedback through the AGC loop filter. Both combine to change the pole (zero) radius r:

$$r = r_1 + d_{rz} \times (1 - b) \times NLF(v) \tag{10}$$

where coefficient r_1 is the minimum radius, corresponding to the maximum damping of the resonator. In a digital implementation, r_1 is given by:

$$r_1 = 1 - damping \times \left(\frac{2\pi f}{f_s}\right) \tag{11}$$

where the coefficient *damping* controls the damping factor, f is the CF from Equation (9), and f_s is the sampling frequency. r_1 keeps the damping away from zero, thereby keeping the system away from the Hopf bifurcation of the resonators. r_1 also makes the damping bounded. The increment of r above r_1 is the relative undamping. It is the product of the nonlinear function (*NLF*) of the CAR velocity, and the AGC loop, b. The coefficient *d_rz* controls the rate at which the velocity and the AGC loop affects the damping. Here, *d_rz* is set to 0.7×(1-r_1) (Lyon, 2017).

The *NLF* function in the DOHC is given by:

$$NLF(v) = \frac{1}{1 + (v \times scale + offset)^2} \tag{12}$$

where v is the CAR velocity, *scale* is 0.1, and *offset* is 0.04 (Lyon, 2017). At high velocities, the velocity-squared function grows very rapidly and saturates the *NLF* toward zero, thus making the damping saturate toward a high-level limit.

The level dependence of the damping mechanism introduces frequency distortions. The velocity-squared function includes a double-frequency term that interacts with the CAR coefficients (a_0r and c_0r) to generate a CDT. For example, if there are two tones, f_1 and f_2 (where $f_1 < f_2$), then a third tone, at the frequency ($2f_1$–f_2) will appear and propagate through the cascade of filters. The *offset* in the NLF function introduces a first order damping factor, which will interact with the CAR coefficients to generate a QDT, (f_2–f_1) (Lyon, 2017).

DIHC

The DIHC models the IHC function. It comprises a high-pass filter (HPF), a transduction nonlinearity unit, a transducer unit and two LPFs. The IHCs are mechano-electrical transducers that sense the BM vibration, convert the mechanical motion into electrical signals, and deliver the results to the nervous system. The DIHC model is shown in **Figure 5**. The HPF suppresses the CAR output frequencies below 20 Hz. The transduction nonlinearity includes a half wave rectifier (HWR), and a rational sigmoid function:

$$u = HWR\left(BM_{hpf} + 0.175\right) \tag{13}$$

$$n = \frac{u^3}{u^3 + u^2 + 0.1} \tag{14}$$

where BM_{hpf} is the high pass filtered CAR output, u is the intermediate variable, and n is the transduction nonlinearity output. The HWR mimics directional sensitivity of the IHC transduction which response mainly in one direction. The constant 0.175 (Lyon, 2017) keeps the nonlinearity at a fixed value at zero response. The rational sigmoid function (14)

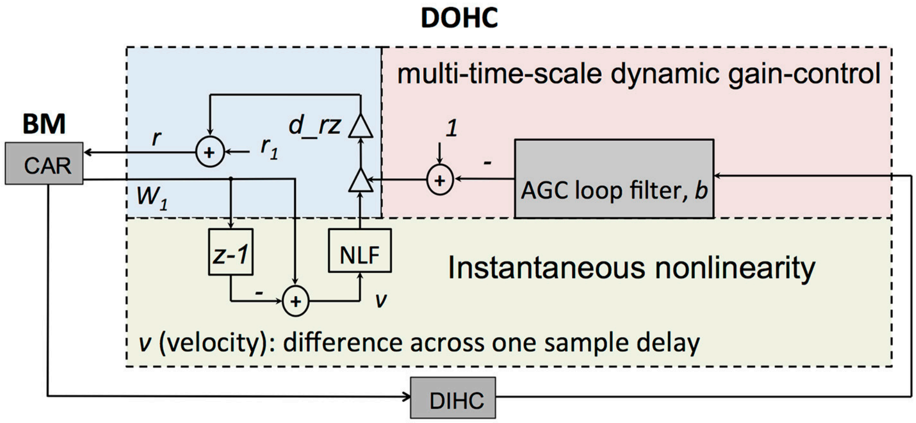

DOHC
multi-time-scale dynamic gain-control
BM
CAR
r
r_1
d_rz
1
-
AGC loop filter, b
W_1
z-1
NLF
Instantaneous nonlinearity
-
+
v
v (velocity): difference across one sample delay
DIHC

FIGURE 4 | Structure of the DOHC model. The instantaneous nonlinearity performs a nonlinear gain control (NLF) on the CAR velocity, which is calculated from the BM coefficient W_1. The multi-time-multi-scale dynamic gain-control factor, *b*, is obtained from the AGC loop. Both gain control factors are combined to change *r* through Equation (10).

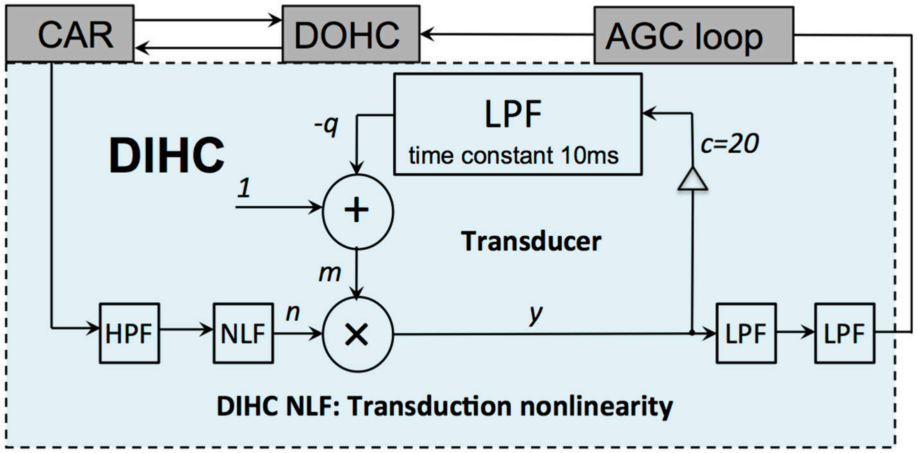


FIGURE 5 | Structure of the DIHC model. It comprises a HPF, a transduction nonlinearity unit, a transducer unit and two LPFs.

provides a nearly linear response at low amplitudes and a saturating response at higher amplitudes.

The transducer unit detects and amplifies the signal onset, then compresses and reduces its response gain quickly after the signal onset. It is implemented by:

$$m = 1 - q \tag{15}$$

$$y = nm \tag{16}$$

$$q_{new} = (1 - a)\, q + a(cy) \tag{17}$$

where m is the adaptive gain of its input, n, c is set to 20, and q is the LPF state. The time constant of the first order FIR LPF is set to 10 ms. The final two FIR LPFs smooth the output using a time constant of 80 µs each.

AGC Loop

The AGC loop consists of a four-stage cascade FIR LPF, with each stage coupled with its left and right neighbors to form a three-stage spatial LPF. It feeds the DIHC signal back to the DOHC at a much lower update rate than other parts of the CAR-FAC model. The AGC loop models the medial olivocochlear system's efferent feedback that exerts an AGC on the BM vibration through the OHCs. The AGC loop filter is shown in **Figure 6.** Each AGC smoothing filter (SF) stage includes a temporal linear LPF with a defined coefficient *c_t* and a three-tap spatial LPF. The three-tap spatial LPF coefficients [s_1, *1*-s_1-s_2, s_2] apply weight s_1 to the left neighbor value, s_2 to the right neighbor value, and *1*-s_1-s_2 to the current channel value to keep the total mixing gain equal to 1. For a 44.1 kHz signal, in the fastest and most local stage, AGC-SF4, *c_t* is set to 0.09, s_1 is 0.14 and s_2 is 0.2 (Lyon, 2017). The input of each AGC-SF comes from a respective accumulation of the DIHC and its lower stage. The AGC-SF4 output *b* feeds back to the DOHC.

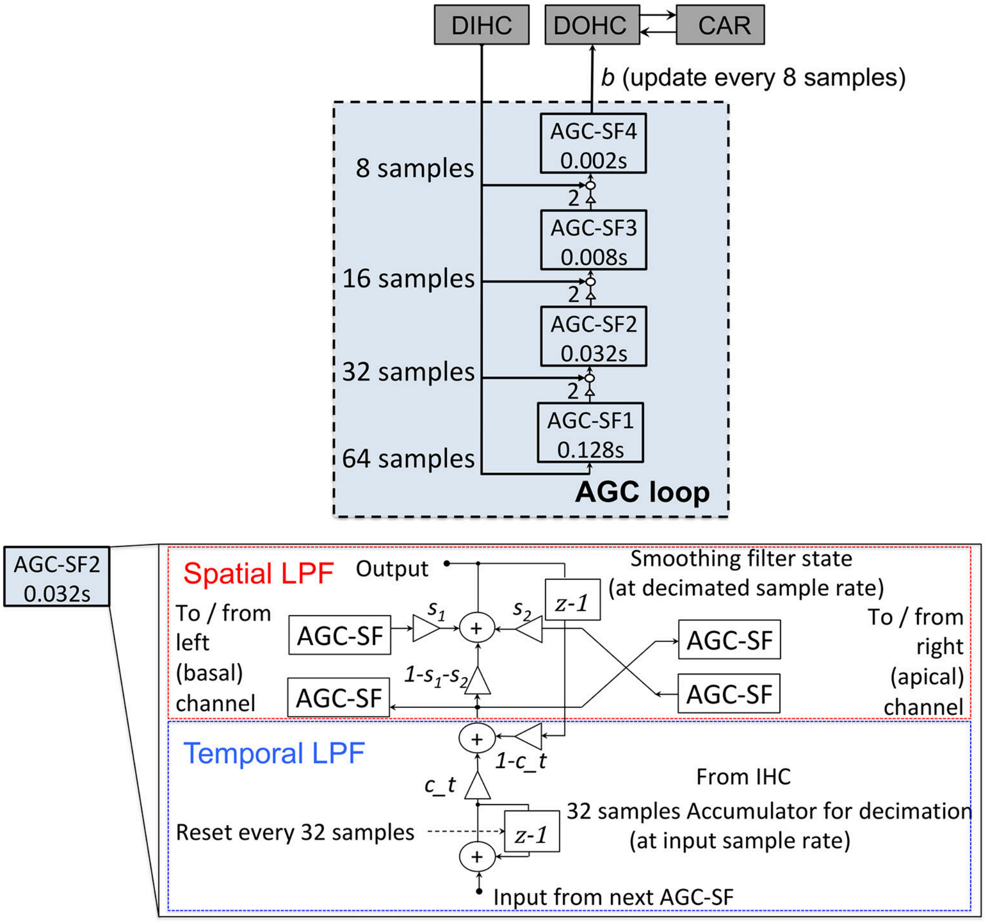


FIGURE 6 | Structure of the AGC loop. Four stages of the temporal smoothing filters (SF) (Upper). Each stage consists of a temporal LPF with a defined time constant (0.002, 0.008, 0.032, and 0.128 s) and a three-tap spatial smoothing filter. The internal structure of an AGC-SF (Lower), the input of the AGC-SF comes from the lower filter stage with the smaller time constant as well as the accumulation of the DIHC. The output goes to the next stage of the temporal filter. The spatial smoothing filter is a three-tap smoothing filter coupled with lateral channels. s_1, s_2, and *1*-s_1-s_2 are the spatial filter coefficients. *c_t* is the temporal LPF coefficient calculated from the time constant.

FPGA Implementation

The CAR-FAC system can be efficiently implemented on FPGA, and the system is configurable in filter parameters and channel numbers **Figure 7** shows the architecture of the system. It comprises an audio codec, a CAR-FAC module, a controller module and an interface module. The system provides two ways of sound input. One way is through the SSM2603 audio codec on the FPGA board. It also supports recorded audio file input from the PC host through a USB 3.0 interface.

The CAR-FAC module implements the components described in section The CAR-FAC Model. Additionally, the CAR module can operate independently: when the FAC function is turned off, the DOHC and AGC loop function will be switched off, and all the CAR coefficients (a_0, c_0, g, h, and r) remain fixed at their initial values. The system then operates as a linear CAR system.

The controller module controls the system data flow, including writing the initial coefficients, and/or the audio file input to the CAR-FAC module, as well as the CAR-FAC module output to the interface module. Additionally, the output of the system is selectable: we can choose either the BM output or the DIHC output as the system output.

The interface module consists of a data synchronization module, an external memory, and a USB interface. The data synchronization circuit synchronizes data between different clock domains. There exist two clock domains in the system: a system clock domain (250 MHz) and an interface clock domain (100 MHz). The system clock domain includes the controller module and the CAR-FAC module. The interface clock domain is unique to the interface module. The external memory is a 1 GB DDR3 SDRAM on the FPGA board: it stores the CAR-FAC output data. The USB interface communicates between the FPGA board and the PC, and transmits the system's initial coefficients (a_0, c_0, g, h, r, r_1, b, and d_rz), and, if required, the input audio file from the PC to the FPGA board. It also transmits the system's output from the external memory to the PC.

We first simulated the CAR-FAC model in Python with floating-point numbers. Next, we verified the model using the fixed-point numbers to determine the required word length for the FPGA implementation. We use 20-bit BM variables, 20-bit DOHC variables, 14-bit DIHC variables and 14-bit AGC variables to approximate the floating-point CAR-FAC performance and to meet the input, output and internal variables range to achieve a 70 dB input dynamic range. We use the pipeline technique to parallel the CAR module, the DOHC module, and the DIHC_AGC module, and the time-multiplexing approach to reuse single CAR, DOHC, and DIHC_AGC hardware module to implement a compact reconfigurable CAR-FAC system. The system design diagram is shown in **Figure 8**.

In digital audio, 44.1 kHz is a common sampling frequency, and the digital hardware of the CAR module (the two-zero-two-pole resonator) and the FAC module (the DOHC module and the DIHC-AGC module) can operate much faster than the audio sample interval (22.68 µs). Hence, in this system, a single CAR-FAC hardware module is reused multiple times to implement the multiple-channel multi-level pipeline CAR-FAC system. At 44.1 kHz sampling frequency, with a single CAR-FAC module, we were able to implement up to 70 filter channels real-time CAR-FAC system.

For each CAR-FAC module, there exist four state machines in the system. The controller state machine determines the cochlear channel to be processed at a particular time and controls the CAR-FAC coefficients and data for that channel. The CAR state machine calculates the transfer function of Equation (1). The DOHC state machine calculates Equation (10–12), and feeds back an updated r to the CAR. The DIHC-AGC state machine calculates Equation (13–17), as well as the AGC_loop function shown in **Figure 6**. The AGC output b feeds back to the DOHC module via Equation (10).

The BM_start signal controls the start of the system through the controller and is triggered by the Audio_in_ready signal. If

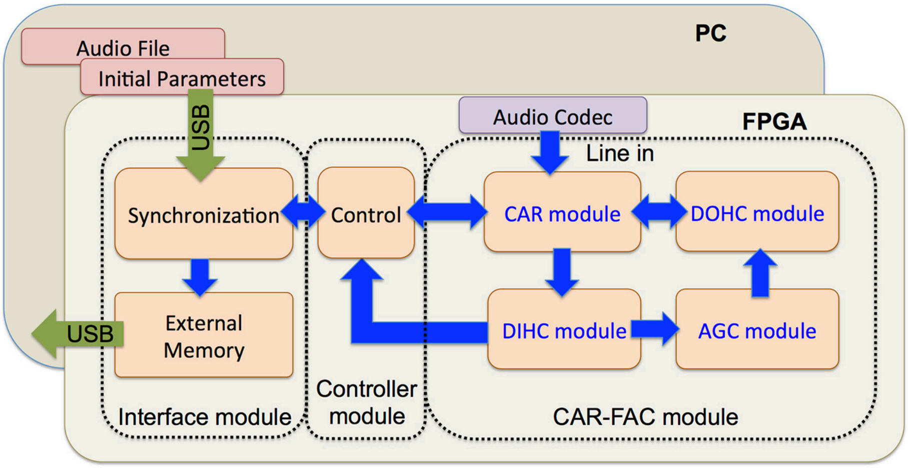


FIGURE 7 | Architecture of the CAR-FAC FPGA system. The system consists of an audio codec, a CAR-FAC module, a controller module and an interface module. The FPGA board is hosted by a PC through a USB interface.

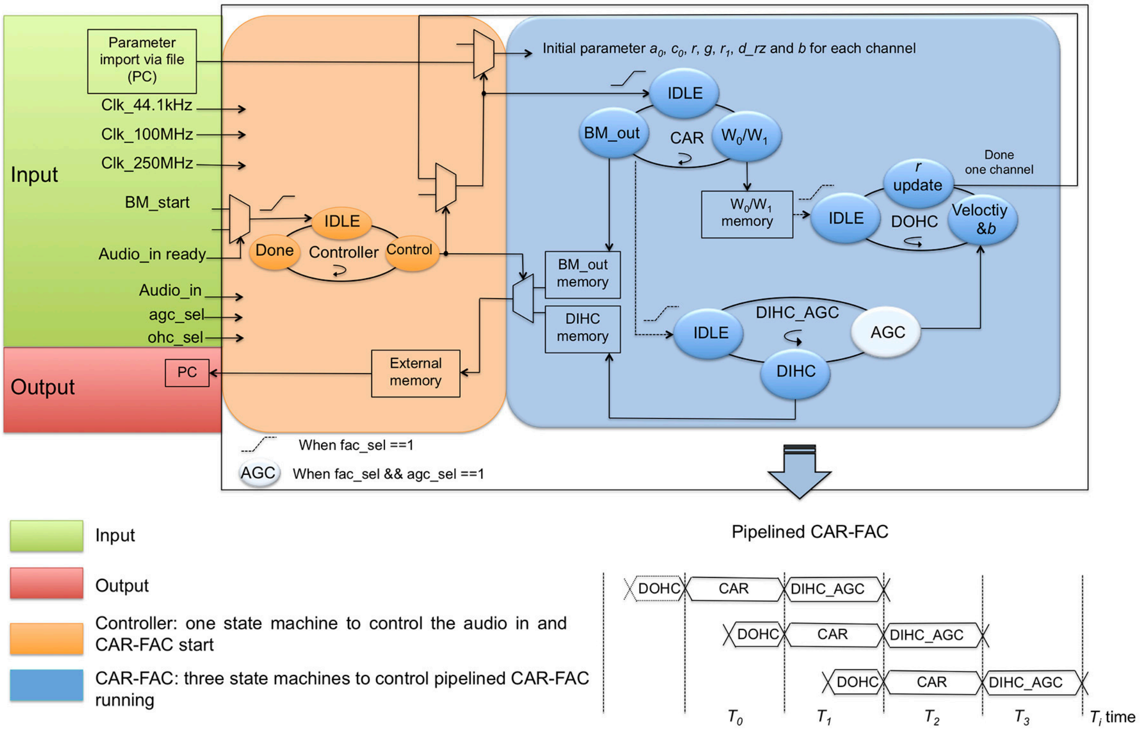


FIGURE 8 | CAR-FAC system design diagram. The CAR-FAC system is implemented with 20-bit word length for the design coefficients, BM output, and DOHC output, and 14-bit for the DIHC output and the AGC output. The controller state machine determines the cochlear channel to be processed at any particular time and controls the CAR-FAC coefficients and data for that channel. The BM_start signal controls the start of the system through the controller, and it is triggered by the Audio_in_ready signal. The ohc_sel is a selector switch for the CAR/CAR-FAC function. The agc_sel is a switch for the AGC loop function. The CAR state machine calculates the transfer function of Equation (1) and controls the DOHC and DIHC_AGC start in the system. The DOHC state machine calculates Equation (10–12) and feeds back an updated r to the CAR. The DIHC-AGC calculates Equation (13–17), as well as the AGC_loop function shown in **Figure 6**. The AGC output b feeds back to the DOHC module via Equation (10). The pipelined CAR-FAC timing diagram is shown in lower right.

there exists an audio input (Audio_in) from either the PC or the audio codec, the BM_start signal will be sent to the CAR through the controller, and the CAR will start to run. The ohc_sel is a selector switch for the CAR/CAR-FAC function, and the agc_sel is a switch for the AGC loop function. When the ohc_sel is low, the DOHC function is switched off, and the CAR-FAC operates as a linear CAR system, and we can choose either the CAR or the DIHC as the output. When both the ohc_sel and the agc_sel are high, the whole CAR-FAC function is switched on. When the ohc_sel is high and the agc_sel is low, the AGC loop function is switched off, leaving only the instantaneous nonlinearity in the CAR-FAC system.

The CAR state machine controls the DOHC and DIHC_AGC start in the system. It will send a start signal to the DOHC and the DIHC-AGC module separately at a particular time to start the DOHC and the DIHC-AGC function if both the ohc_sel and the agc_sel are high. The DOHC state machine starts when the CAR module finishes updating the internal variables W_0/W_1. The DIHC-AGC state machine starts when the BM output calculation is finished. The pipelined CAR, DOHC, and DIHC_AGC structure is shown in **Figure 8** bottom right. Each filter channel output, BM_out or DIHC_out, is moved to the external memory in the interface module and sent to the PC through the USB interface.

The device utilization for a single CAR-FAC module is shown in **Table 1**. Given the size of a Cyclone V FPGA and the low hardware resource utilization of a single CAR-FAC hardware module, this FPGA board can accommodate up to a total of 210 cochlear channels (using three CAR-FAC hardware modules).

RESULTS

CAR-FAC Transfer Function

We have implemented a real-time digital CAR-FAC system at a 44.1 kHz sampling rate on a Cyclone V FPGA board covering an input frequency range up to 22.05 kHz. The number of channels in the system is reconfigurable, and more channels will result in more overlap among filters if the frequency range is kept the same. For machine hearing applications, about 50% overlap in

TABLE 1 | Device utilization summary.

	Used	Available	Utilization (%)
ALM	5,235	29,080	18
Memory (bits)	1,082,812	4,567,040	24
DSPs	49	150	33

items of equivalent rectangular bandwidth (ERB) is considered to provide a well-behaved representation of a sound (Lyon, 2011). Psychophysical experiments (Glasberg and Moore, 1990; Moore, 1995) show that each ERB at moderate sound level corresponds to about 0.89 mm on the BM. Therefore, for the total length of the human BM (about 35 mm), this would correspond to 78 channels with 50% overlap, or 11 channels per octave according to the Greenwood function map in Equation (9). Machine hearing models typically use 60 to 100 channels in total (Lyon, 2011), here we implemented a 70-channel CAR-FAC system and investigated the system characteristics.

The measured system transfer function in response to a -40 dB full scale (FS), 1 s sine tone sweep from 20 Hz to 22.05 kHz (squared-cosine rise and decay time of 0.1 s to minimize the influence of the spectral splatter) is shown in **Figure 9.** Note that we express the intensity of input signals in dB FS relative to a maximum amplitude of FFFFF (20-bit unsigned number), and the input amplitude is normalized to 1.0 in the figures in this paper. The upper set of curves shows the linear CAR response of all the 70 channels when the FAC function is switched off. The lower set shows the CAR-FAC response. Both the CAR and the CAR-FAC show an increased gain in the lower and moderate frequency range and a reduced gain in the higher frequency range. Additionally, the FAC function shows a global gain compression effect on the system response.

Figure 10 shows the CAR and the CAR-FAC output in the time domain in response to 0.5, 1, 2, and 4 kHz tones (squared-cosine rise and decay time of 10 ms) at channels of CFs corresponding to the input tones. The CAR amplifies the amplitude of the input tones linearly, whereas the CAR-FAC responses exhibit a gradually compressed gain control.

CAR-FAC Excitation Patterns and Nonlinear Growth

Excitation patterns show the vibration amplitude across the BM to a single sound. Here, the excitation patterns were calculated as the root-mean-square (RMS) signal at the output of all the CAR-FAC channels (Ren, 2002). The Greenwood function in Equation (9) was used as the position-frequency map.

Figures 11A–E show excitation patterns in response to 100 ms tones at 0.5, 1, 2, 4, and 8 kHz (squared-cosine rise and decay time of 10 ms) with intensities ranging from -65 dB FS to -15 dB FS in steps of 10 dB FS. The peak locations of all excitation patterns correspond to the input tones through the position-frequency map, demonstrating that the system captures the human frequency-position map well.

Additionally, we calculated the BM input/output (I/O) function to evaluate the nonlinear and compression effects of the system. The I/O function is the ratio between the RMS output at the CF channel corresponding to the stimulus frequency and the RMS of the stimulus (Saremi et al., 2016). **Figure 11F** shows the I/O function curves of the system to 100 ms pure tones of 0.5, 1, 2, 4 and 8 kHz (squared-cosine rise and decay time of 10 ms) with intensities between -65 dB FS to -15 dB FS in steps of 10 dB FS. The I/O curves were normalized with respect to the -65 dB FS I/O point. The output shows a compressed intensity range (15 dB FS) comparing to the input (50 dB FS), and the I/O curves were generally more compressive at moderate CFs, such as 1, 2, and 4 kHz, than the lower and higher CFs (0.5 and 8 kHz).

CAR-FAC Frequency Selectivity and *Q* Tuning

The CAR-FAC frequency selectivity was evaluated from the system frequency responses. The frequency response was calculated using the FFT from the system impulse responses at the channels of CFs corresponding to 0.5, 1, 2, 4, and 8 kHz.

Furthermore, in the CAR-FAC system, quality factor (*Q* factor) tuning is achieved by tuning of the damping factor [*damping* in Equation (11)]. Here, to investigate the system's *Q* tuning effects, we used different damping factors and calculated the corresponding *Q* factors associated with the ERB, Q_{ERB} (de Boer and Nuttall, 2000):

$$Q_{ERB} = \frac{CF}{ERB} \tag{18}$$

The ERB was evaluated from the system's impulse response power spectral density (PSD).

Figures 12A–E shows the system's frequency responses at output channels of CFs corresponding to 0.5, 1, 2, 4, and 8 kHz to -20 dB FS, 40 μs condensation clicks. The *damping* in the system was set as 0.4, 0.5, and 0.7, respectively. The smaller damping corresponds to higher gain at all CFs. **Figure 12F** shows the calculated Q_{ERB} under different damping factors. The smaller Q_{ERB} corresponds to higher damping, and at higher damping (0.5 and 0.7), Q_{ERB} is higher at moderate CFs than lower and higher CFs.

The relation between dB FS and Sound Pressure Level, expressed in dB SPL, depends on the *damping* set-point used in the CAR-FAC model [r_1 in Equation (10)]. Comparing the peak gain at moderate frequencies (1, 2, and 4 kHz) with the measured biological cochlea frequency response in **Figure 1**, we can see that using a *damping* factor of 0.4, the -20 dB FS input has ∼60 dB peak gain, which fits the 30 dB SPL input intensity curve in **Figure 1.** Accordingly, at 0.5 *damping*, the -20 dB FS corresponds to 60 dB SPL, and at 0.7 *damping*, the -20 dB FS corresponds to 70 dB SPL.

We also investigated the system's impulse response characteristics in the time domain and the intensity dependence of the Q_{ERB} factors. **Figure 13** (Left) shows the CAR-FAC impulse responses at CFs corresponding to 1 kHz to a condensation click with -50 dB FS, -30 dB FS, and -10 dB FS intensity respectively. It shows the CAR-FAC filter characteristic that the shape and the amplitude of the impulse responses varied while the zero-crossing timing remains the same across the stimulus levels. **Figure 13** (Right) shows the calculated Q_{ERB} factor for clicks with intensities between -60 dB FS and -10 dB FS in steps of 10 dB FS at the CF corresponding to 1 kHz. The Q_{ERB} factor decreases as the stimulus intensity increases. The sharpness of the frequency response thus decreases as the stimulus intensity increases.

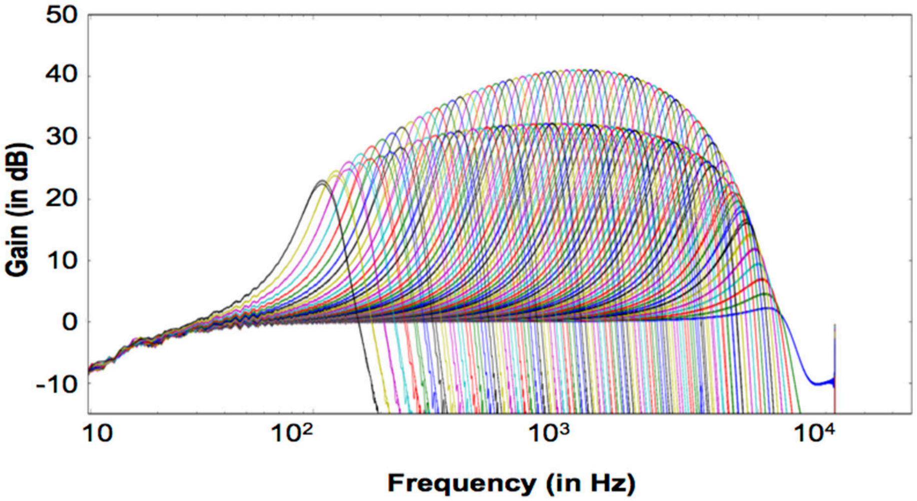


FIGURE 9 | Transfer function of the 70-channel CAR-FAC system to a -40 dB FS, 1 s sine tone sweep from 20 Hz to 22.05 kHz (squared-cosine rise and decay time of 0.1 s to minimize the influence of the spectral splatter). The CAR response (Upper) when the FAC function is switched off; The CAR-FAC response (Lower).

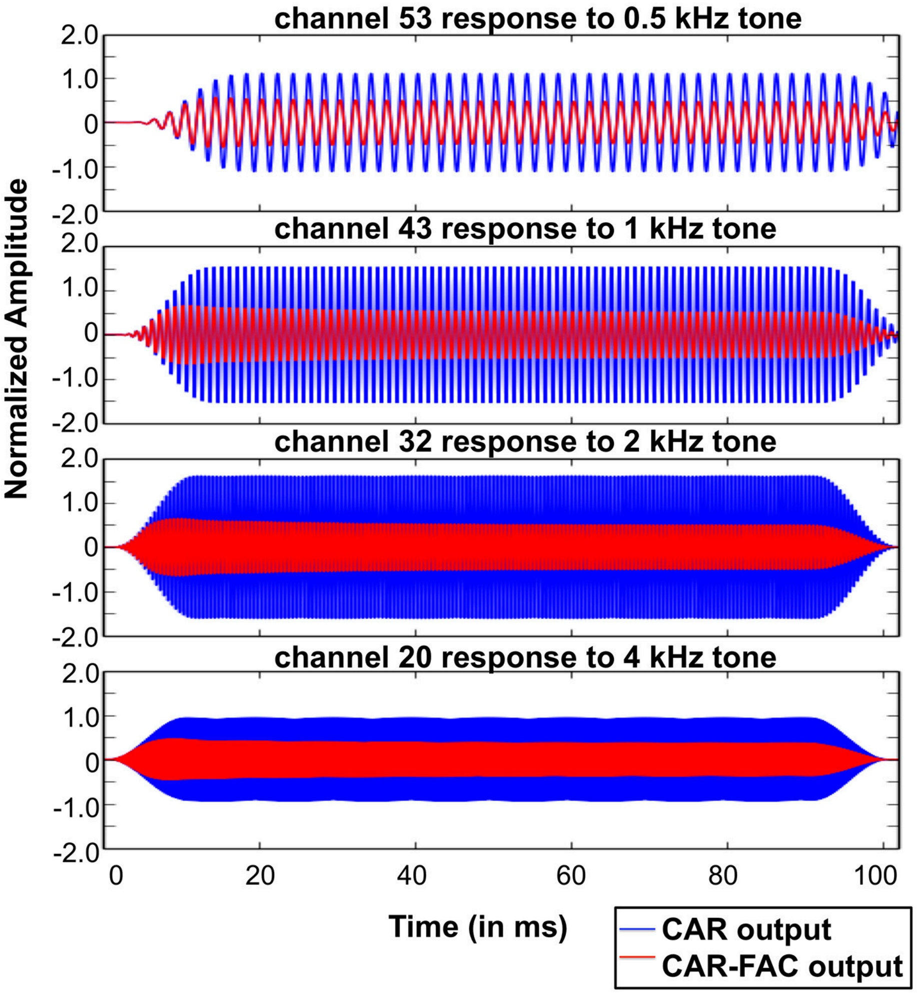


FIGURE 10 | CAR and CAR-FAC output in response to 0.5, 1, 2, and 4 kHz tones with an amplitude of -40 dB FS at the channels of CFs corresponding to the input frequencies.

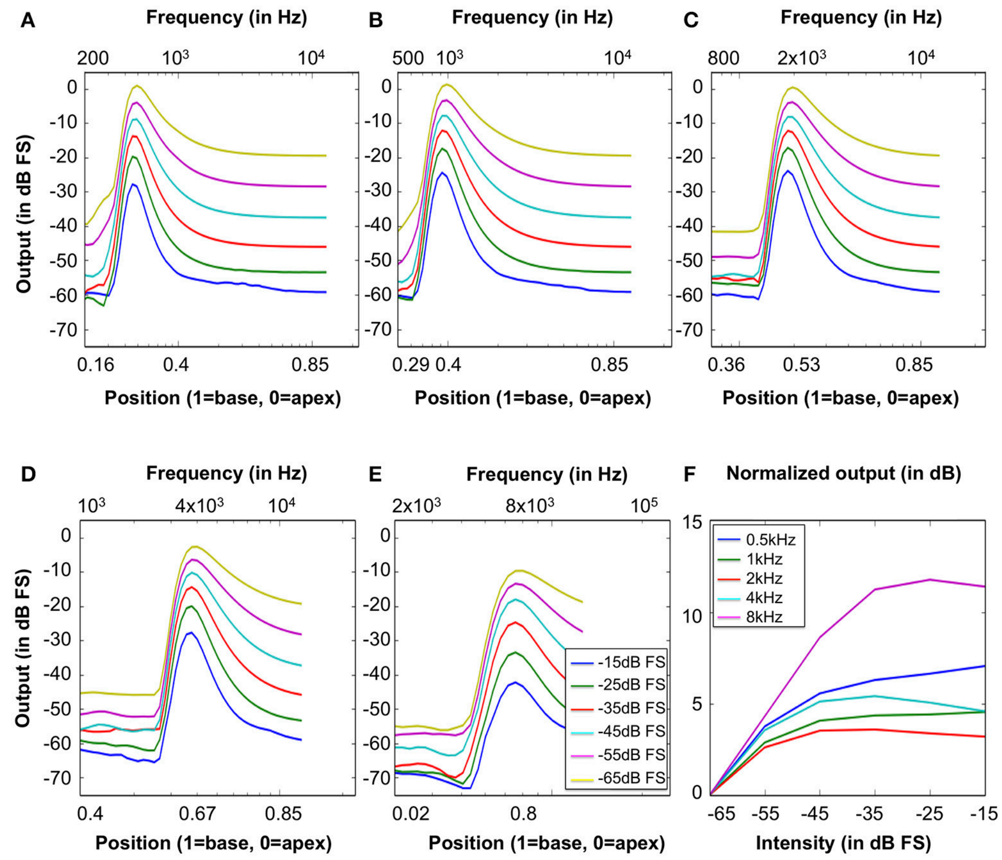


FIGURE 11 | Excitation patterns calculated as the RMS output signal of the 70 CAR-FAC channels in response to tones at **(A)** 0.5 kHz, **(B)** 1 kHz, **(C)** 2 kHz, **(D)** 4 kHz, and **(E)** 8 kHz with intensities ranging from -65 dB FS to -15 dB FS in steps of 10 dB FS. The x-axis shows both the frequency and the position-frequency location calculated from Equation (9). **(F)** The normalized nonlinear response growth of the system to the tones of 0.5, 1, 2, 4, and 8 kHz (squared-cosine rise and decay time of 10 ms) with intensities between -65 dB FS and -15 dB FS in steps of 10 dB FS.

DIHC Model Output

To investigate the DIHC characteristics, we measured the DIHC response to tones. In order to present stimuli with same amplitude to the DIHC, we made use of the linearity of the CAR: we switched off the FAC function, leaving the CAR amplifying the input tones linearly. Firstly, we presented 0.5, 1, and 4 kHz tones to the system, and measured the CAR output at channels with CFs corresponding to each of those tones. We adjusted each tone's amplitude to make sure the CAR output at the corresponding channel had the same amplitude of 2.28 dB FS. Next, we used the adjusted tones as the input to the system and measured the DIHC output in response to those tones with the same CAR output amplitude at the corresponding CFs (Gmel et al., 2011).

Figure 14 shows the DIHC output in response to 100 ms tones of 0.5, 1, and 4 kHz (squared-cosine rise and decay time of 10 ms). The DIHC detects and amplifies input signal onset well. For lower frequencies, e.g., 0.5 kHz, the DIHC output shows little DC offset and follows the sinusoidal curve of the input. As the input frequency is increased, the DIHC shows higher offset and reduced gain.

DISCUSSIONS

This paper presents a fully digital implementation of the CAR-FAC cochlear model. We use time-multiplexing and pipeline parallelizing techniques to implement a 70-channel real time CAR-FAC system at 44.1 kHz on a Cyclone V FPGA board. We measured the system responses to a set of stimuli such as pure tones and condensation clicks and analyzed the CAR-FAC nonlinear growth characteristics, excitation patterns, frequency selectivity and impulse response. We investigated the CAR-FAC Q tuning effects thought the damping factor tuning in Equation (10). Additionally, we measured the DIHC model responses to tones.

Here, we compare the system with prior silicon cochleae with respect to architecture, channel number, frequency range, input range, Q tuning, and power consumption, as shown in **Table 2** (Fragniere, 2005; Sarpeshkar et al., 2005; Wen and Boahen, 2006; Yang et al., 2015, 2016). We use a power analysis tool, PowerPlay, provided by Altera to estimate the power consumption of the system on FPGA, since a direct measurement of the power consumption on the FPGA board

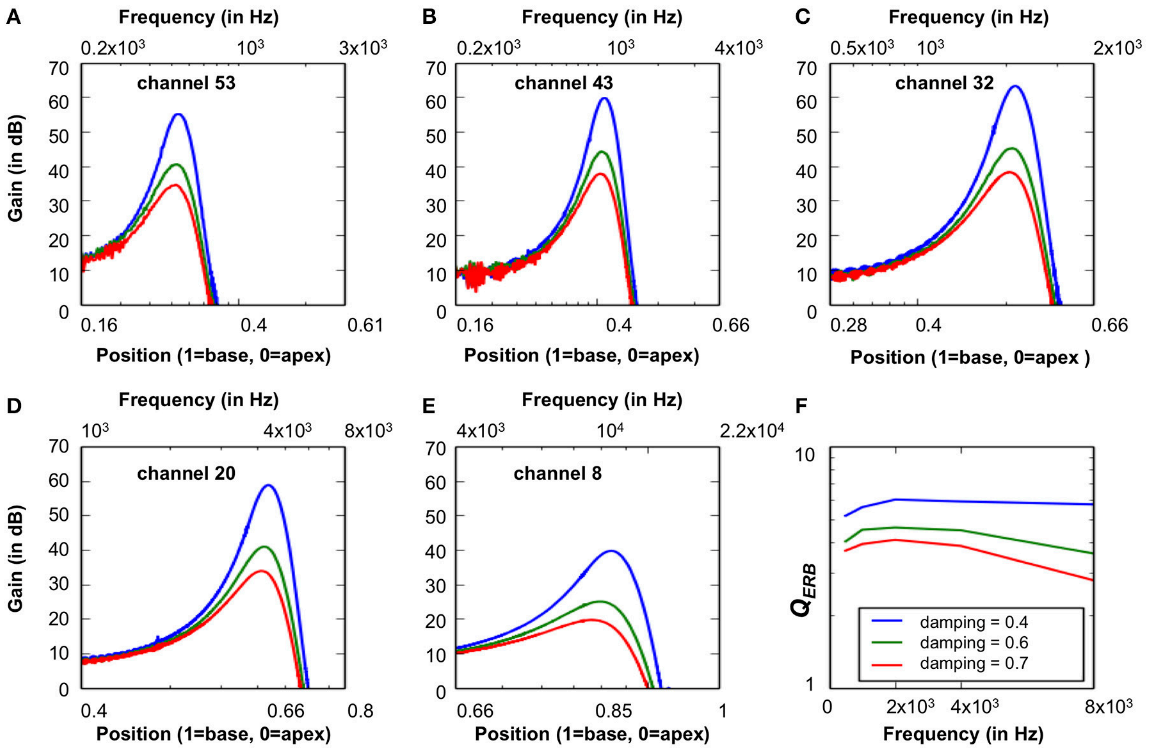


FIGURE 12 | (A–E) The CAR-FAC system response calculated at the CFs corresponding to 0.5, 1, 2, 4, and 8 kHz with three damping factors (0.4, 0.5, and 0.7) in Equation (11). The x-axis shows both the frequency and the BM location calculated from Equation (9). **(F)** The corresponding Q_{ERB} at CFs corresponding to 1, 0.5, 2, 4, and 8 kHz estimated from the BM impulse response PSD at CFs.

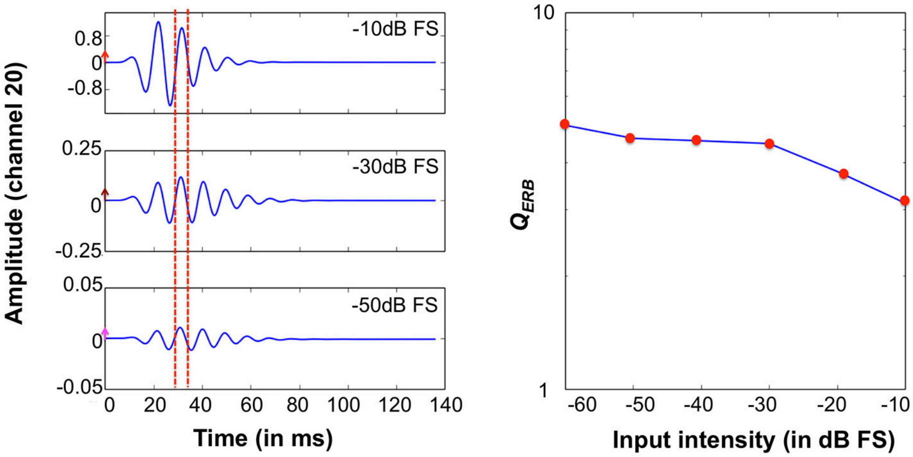


FIGURE 13 | System impulse responses at the 1 kHz CF channel to -50 dB FS, -30 dB FS, -10 dB FS clicks. The arrows mark the amplitude of clicks. The red dashed lines mark two consecutive impulse response zero-crossings (**Left**). 1 kHz Q_{ERB} factors derived from impulse responses at relative intensities from -60 dB FS and -10 dB FS in steps of 10 dB FS (**Right**).

is not possible for this development kit. **Table 2** reports the estimated FPGA chip power consumption by PowerPlay based on its default settings. The CAR-FAC system shows a wide input frequency range and dynamic range, and a small Q tuning range. The power consumption of the whole FPGA board is high compared to other analog silicon cochleae. However, this fully digital system is stable, scalable, and easy to use. Additionally, it shows an outstanding agreement with the biological data recordings and an improved signal to noise ratio (SNR) (Saremi et al., 2016). It is thus able to provide an excellent input

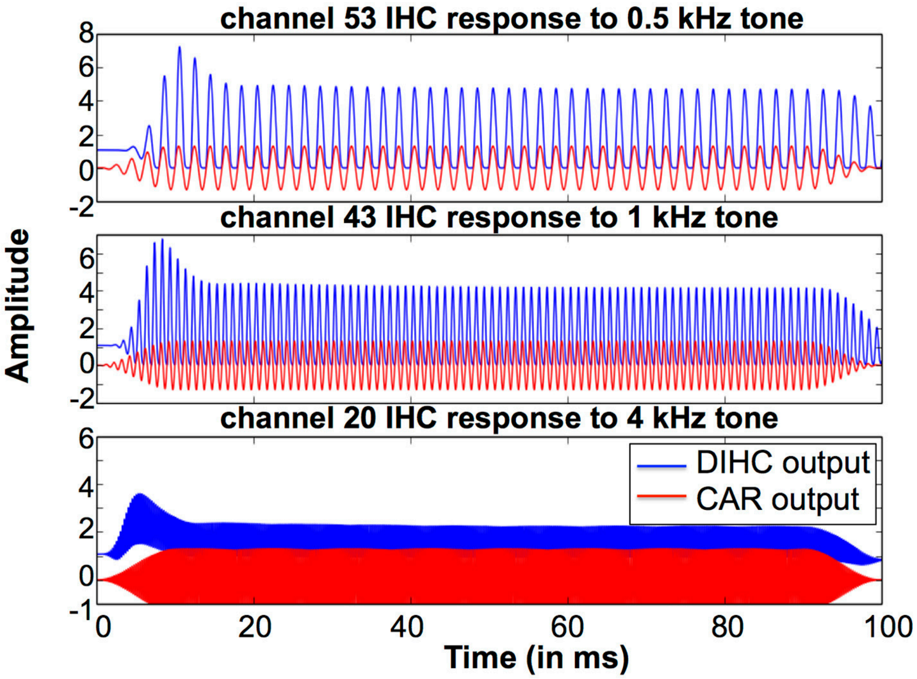


FIGURE 14 | DIHC output and CAR output in response to 100 ms tones of 0.5, 1, and 4 kHz at the channels of CFs corresponding to those tones.

TABLE 2 | Comparison with prior silicon cochleae.

	This work	**Yang et al. (2016)**	**Yang et al. (2015)**	**Wen and Boahen (2006)**	**Sarpeshkar et al. (2005)**	**Fragniere (2005)**
Architecture	Cascade	Parallel	Parallel	Active coupling	Parallel	Passive coupling
Channel number	70 × 3[a]	64 × 2	16	360	16	100
Frequency range	up to 22.05 k Hz	8–20 k Hz	N/A	210–14 k Hz	100–5 k Hz	200–20 k Hz
Input range (dB)	70	73(including 18dB of the attenuator)	92	52	75(with AGC) 55(without AGC)	50
Power supply (V)	1.1	0.5	1.8	2.5	2.8	3.3
Power (*mW*)	1,260[b]	0.055	0.028	35.9	0.06	1.7
Q tuning	<10 (through *damping* tuning)	1.3-39 from channel 18	0.83-7	1.16 ± 0.92	<10	0.25–12

[a] *The FPGA ALM utilization is only 18% for one CAR-FAC module, so the system can be rescaled up to a maximum of 210 cochlear channels by implementing three CAR-FAC modules on the FPGA board.*

[b] *The power consumption of the CAR-FAC system is given as the whole FPGA chip power consumption including the PLLs, the DSPs, the RAMs, the IOs and the Logics, and the static power consumption of the whole chip is 240 mW.*

hardware stage to more complex machine hearing tasks such as sound localization, sound segregation, speech recognition, and so on.

AUTHOR CONTRIBUTIONS

YX, RW, and AvS: proposed the idea and designed the FPGA system; YX: recorded the data; YX, TH, RW, and AvS: evaluated and discussed the results; YX: wrote the manuscript. All authors discussed the results, commented on the manuscript and approved it for publication.

ACKNOWLEDGMENTS

This work was supported by the Australian Research Council Grant DP140103001. It was inspired by a project at 2016 Telluride Neuromorphic workshop. The support by the Altera university program is gratefully acknowledged.

REFERENCES

Altoè, A., and Pulkki, V. (2014). Transmission line cochlear models: improved accuracy and efficiency. *J. Acoust. Soc. Am.* 136, 302–308. doi: 10.1121/1.4896416

Chan, V., Liu, S. C., and van Schaik, A. (2007). AER EAR: a matched silicon cochlea pair with address event representation interface. *IEEE Trans. Circ. Syst.* 54, 48–59. doi: 10.1109/TCSI.2006.887979

de Boer, E., and Nuttall, A. L. (2000). The mechanical waveform of the basilar membrane. III. Intensity effects. *J. Acoust. Soc. Am.* 107, 1497–1507. doi: 10.1121/1.428436

Duifhuis, H. (2004). Comment on "An approximate transfer function for the dual-resonance nonlinear filter model of auditory frequency selectivity" [J. Acoust. Soc. Am. 114, 2112-21171 (L)]. *J. Acoust. Soc. Am.* 115, 1889–1890. doi: 10.1121/1.1694999

Fettiplace, R., and Hackney, C. M. (2006). The sensory and motor roles of auditory hair cells. *Nat. Rev. Neurosci.* 7, 19–29. doi: 10.1038/nrn1828

Fragniere, E. (2005). "A 100-channel analog CMOS auditory filter bank for speech recognition," in *Solid-State Circuits Conference, (2005). Digest of Technical Papers*, 29, 297–299.

Glasberg, B. R., and Moore, B. C. (1990). Derivation of auditory filter shapes from notched-noise data. *Hear. Res.* 47, 103–138. doi: 10.1016/0378-5955(90)90170-T

Glasberg, B. R., Moore, B. C., and Nimmo-smith, I. (1984). Comparison of auditory filter shapes derived with three different maskers maskers. *J. Acoust. Soc. Am.* 75, 536. doi: 10.1121/1.390487

Gmel, G., Hamilton, T. G., Leblebici, Y., van Schaik, A. (2011). "A silicon model of the inner hair cell," in *International Conference on Intelligent Sensors, Sensor Networks and Information Processing* (Adelaide, SA), 91–96.

Greenwood, D. D. (1990). A cochlear frequency-position function for several species - 29 years later. *J. Acoust. Soc. Am.* 87, 2592–2605. doi: 10.1121/1.399052

Hamilton, T. J., Jin, C., van Schaik, A., and Tapson, J. C. (2008). An Active 2-D Silicon Cochlea. *IEEE Trans. Biomed. Circuits Syst.* 2, 30–43. doi: 10.1109/TBCAS.2008.921602

Irino, T., and Patterson, R. D. (2001). A compressive gammachirp auditory filter for both physiological and psychophysical data. *J. Acoust. Soc. Am.* 109, 2008–2022. doi: 10.1121/1.1367253

Jimenez-Fernandez, A., Cerezuela-Escudero, E., Miro-Amarante, L., Dominguez-Moralse, M. J., de Asis Gomez-Rodriguez, F., and Linares-Barranco, A. (2016). A Binaural neuromorphic auditory sensor for FPGA: a spike signal processing approach. *IEEE Trans. Neural Netw. Learn. Syst.* 28, 804–818. doi: 10.1109/TNNLS.2016.2583223

Katsiamis, A. G., Drakakis, E. M., and Lyon, R. F. (2007). Practical gammatone-like filters for auditory processing. *EURASIP J. Audio Speech Music Proc.* 2007:063685. doi: 10.1155/2007/63685

Kim, D. O. (1986). Active and nonlinear cochlear biomechanics and the role of outer-hair-cell subsystem in the mammalian auditory system. *Hear. Res.* 22, 105–114. doi: 10.1016/0378-5955(86)90088-2

Liu, S. C., van Schaik, A., Minch, B., and Delbruck, T. (2014). Asynchronous binaural spatial audition sensor with 2x64x4 channel output. *IEEE Trans. Biomed. Circuits Syst.* 8, 453–464. doi: 10.1109/TBCAS.2013.2281834

Liu, Y.-W., and Neely, S. T. (2010). Distortion product emissions from a cochlear model with nonlinear mechanoelectrical transduction in outer hair cells. *J. Acoust. Soc. Am.* 127, 2420–2432. doi: 10.1121/1.3337233

Lyon, R. F. (1997). "All-pole auditory filter models," in *Diversity in Auditory Mechanics*, eds E. R. Lewis, G. R. Long, R. F. Lyon, P. M. Narins, C. R. Steele, and E. Hecht-Poinar (Singapore: World Scientific), 205–211.

Lyon, R. F. (1998). "Filter cascades as analogs of the cochlea," in *Neuromorphic Systems Engineering: Neural Networks in Silicon* (Springer, Boston, MA), 3–18. doi: 10.1007/978-0-585-28001-1_1

Lyon, R. F. (2011). Cascades of two-pole–two-zero asymmetric resonators are good models of peripheral auditory function. *J. Acoust. Soc. Am*. 130, 3893–3904. doi: 10.1121/1.3658470

Lyon, R. F. (2017). *Human and Machine Hearing -Extracting Meaning from Sound.* Mountain View, CA: Cambridge University Press.

Lyon, R. F., Katsiamis, A. G., and Drakakis, E. M. (2010). "History and future of auditory filter models," in *IEEE International Symposium on Circuits and Systems (ISCAS)* (Paris), 3809–3812. doi: 10.1109/ISCAS.2010.5537724

Lyon, R. F., and Mead, C. (1988). An analog electronic cochlea. *IEEE Trans. Acoust.* 36, 1119–1134. doi: 10.1109/29.1639

Moore, B. C. J. (1995). "Frequency analysis and masking," in *Hearing*, ed B. C. J. Moore (New York, NY: Academic), 161–205.

Nouri, M., Ahmadi, A., Alirezaee, S.,and Abbott, D. (2015). A hopf resonator for 2-D artificial cochlea: piecewise linear model and digital implementation. *IEEE Trans. Circ. Syst.* 62, 1117–1125. doi: 10.1109/TCSI.2015.2390555

Patterson, R. D., Unoki, M., and Irino, T. (2003). Extending the domain of center frequencies for the compressive gammachirp auditory filter. *J. Acoust. Soc. Am.* 114, 1529–1542. doi: 10.1121/1.1600720

Ren, T. (2002). Longitudinal pattern of basilar membrane vibration in the sensitive cochlea. *Proc. Natl. Acad. Sci. U.S.A.* 99, 17101–17106. doi: 10.1073/pnas.262663699

Ruggero, M. A. (1992). Responses to sound of the basilar membrane of the mammalian cochlea. *Curr. Opin. Neurobiol.* 2, 449–456. doi: 10.1016/0959-4388(92)90179-O

Saremi, A., Beutelmann, R., Dietz, M., Ashida, G., Kretzberg, J., and Verhulst, S. (2016). A comparative study of seven human cochlear filter models. *J. Acoust. Soc. Am.* 140, 1618–1634. doi: 10.1121/1.4960486

Saremi, A., and Stenfelt, S. (2013). Effect of metabolic presbyacusis on cochlear responses: a simulation approach using a physiologically-based model. *J. Acoust. Soc. Am.* 134, 2833–2851. doi: 10.1121/1.4820788

Sarpeshkar, R., Baker, M. W., Salthouse, C. D., Sit, J.-J., Turicchia, L., and Zhak, S. P. (2005). "An analog bionic ear processor with zero-crossing detection," in *IEEE International Solid-State Circuits Conference* (San Francisco, CA), 78–79.

Sarpeshkar, R. (2006). Brain power - borrowing from biology makes for low power computing [bionic ear]. *IEEE Spectrum* 43, 24–29. doi: 10.1109/MSPEC.2006.1628504

Thakur, C. S., Hamilton, T. J., Tapson, J., and Lyon, R. F. (2014). "FPGA Implementation of the CAR Model of the Cochlea," in *IEEE International Symposium on Circuits and Systems* (Melbourne VIC), 1853–1856.

Unoki, M., Irino, T., Glasberg, B., Moore, B. C., and Patterson R. D. (2006). Comparison of the roex and gammachirp filters as representations of the auditory filter. *J. Acoust. Soc. Am.* 120, 1474–1492. doi: 10.1121/1.2228539

van Schaik, A., and Fragniere, E. (2001). Pseudo-voltage domain implementation of a 2-dimensional silicon cochlea. *IEEE Int. Symp. Circ. Syst.* 3, 185–188. doi: 10.1109/ISCAS.2001.921277

Wang, S., Koickal, T. J., Hamilton, A., Cheung, R., and Smith, L. S. (2015). A bio-realistic analog CMOS cochlea filter with high tunability and ultra-steep roll-off. *IEEE Trans. Biomed. Circ. Syst.* 9, 297–311. doi: 10.1109/TBCAS.2014.2328321

Wen, B., and Boahen, K. (2006). "A 360-channel speech preprocessor that emulates the cochlear amplifier," in *IEEE International Solid-State Circuits Conference* (San Francisco, CA), 556–557.

Xu, Y., Thakur, C. S., Singh, R. K., Wang, R., Tapson, J. C., and van Schaik, A. (2016). "Electronic Cochlea: CAR-FAC Model on FPGA," in *IEEE Biomedical Circuits and Systems Conference* (Shanghai), 1–4.

Yang, G., Lyon, R. F., and Drakakis, E. M. (2015). Psychophysical evaluation of an ultra-low power, analog biomimetic cochlear implant processor filterbank architecture with across channels AGC. *IEEE Trans. Audio Speech Lang. Process.* 23, 2465–2473. doi: 10.1109/TASLP.2015.2488290

Yang, M., Chien, C.-H., Delbrück, T., and Liu, S.-C. (2016). A 0.5 V 55 μW 64 × 2 channel binaural silicon cochlea for event-driven stereo-audio sensing. *IEEE J. Solid State Circ.* 51, 2554–2569. doi: 10.1109/JSSC.2016.2604285

Zweig, G., Lipes, R., and Pierce, J. R. (1976). The cochlear compromise. *J. Acoust. Soc. Am.* 59, 975–982. doi: 10.1121/1.380956

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Neuroscience

ORIGINAL RESEARCH
published: 21 March 2018
doi: 10.3389/fnins.2018.00177

Managing Clutter in a High Pulse Rate Echolocation System

Jacob Isbell[1]* *and Timothy K. Horiuchi*[1,2,3]

[1] Electrical and Computer Engineering Department, University of Maryland, College Park, College Park, MD, United States, [2] Institute for Systems Research, University of Maryland, College Park, College Park, MD, United States, [3] Neuroscience and Cognitive Science Department, University of Maryland, College Park, College Park, MD, United States

The use of echolocation for navigating in dense, cluttered environments is a challenge due to the need for rapid sampling of *nearby* objects in the face of delayed echoes from *distant* objects. In the wild, echolocating bats frequently encounter this situation when leaving the roost or while hunting. If long-delay echoes from a distant object are received after the next pulse is sent out, these "aliased" echoes appear as close-range phantom objects. Little is known about how bats cope with these situations. In this work, we demonstrate a novel strategy to manage aliasing in cases where a single target is actively being tracked at close range. This paper presents three reactive strategies for a high pulse-rate sonar system to combat aliased echoes: (1) changing the interpulse interval to move the aliased echoes away in time from the tracked target, (2) changing positions to create a geometry without aliasing, and (3) a phase-based, transmission beam-shaping strategy to illuminate the target and not the aliasing object.

Keywords: echolocation, sonar, bats, clutter, interpulse interval, pulse-echo ambiguity

OPEN ACCESS

Edited by:
John Gregory Harris,
University of Florida, United States

Reviewed by:
Soumyajit Mandal,
Case Western Reserve University,
United States
Yi-Wen Liu,
National Tsing Hua University, Taiwan

Correspondence:
Jacob Isbell
jisbell@umd.edu

Specialty section:
This article was submitted to
Neuromorphic Engineering,
a section of the journal
Frontiers in Neuroscience

Received: *01 November 2017*
Accepted: *05 March 2018*
Published: *21 March 2018*

Citation:
Isbell J and Horiuchi TK (2018)
Managing Clutter in a High Pulse Rate
Echolocation System.
Front. Neurosci. 12:177.
doi: 10.3389/fnins.2018.00177

INTRODUCTION

Bat echolocation is the unusual ability by bats to emit an ultrasonic sound pulse and measure the time until echoes begin to arrive (for estimating range) combined with the more general ability of mammals to determine the direction of sound. The ultrasonic frequencies used by bats are difficult to detect by most animals and have short wavelengths (∼ 3–17 mm) that produce detectable echoes from small insects. To localize the direction of echoes, bats (e.g., the big brown bat) have been shown to rely primarily on the use of interaural level differences produced by the head and pinnae, a common strategy for small mammals (Grothe et al., 2010). The use of ultrasonic frequencies and a small head size, strongly limit the use of phase-locking, and interaural-timing cues for localization. To estimate range, the bat measures the time-of-flight of the echo from an emitted sound. From an auditory processing point of view, echolocation is unique in that the sound being analyzed is *generated* by the bat and is therefore both known and under the control of the bat. It is well known that bats change both the properties of the echolocation pulse and the timing of pulses in response to their environment (Petrites et al., 2009; Hiryu et al., 2010; Bates et al., 2011), but seldom has this dynamic behavior been adopted in artificial sonar systems.

A typical operational assumption in echolocation is that all of the sounds following an emitted pulse are echoes from the *most recent* outgoing pulse. The duration of perceptible echoes resulting from a given pulse depends on the properties of the outgoing pulse (such as the amplitude, spectrum, and duration) as well as the properties of the environment (such as the distance, size, shape, orientation, and overall configuration of objects). A common-sense rule is that the next pulse should not be emitted until *all* perceptible echoes from the previous pulse have died out. In the majority of situations, bats appear to avoid this pulse-echo ambiguity, or "aliasing." Studies of big brown bats navigating in extremely cluttered environments, however, show cases where bats appear to tolerate such aliasing to sample the environment at a high-rate (Petrites et al., 2009; Schmidt et al., 2011).

In close-quarters maneuvering, a high sampling rate is desirable when the angle to nearby objects is changing rapidly. Little is known about what bats do when a high pulse rate is needed to maneuver near objects in an environment that produces long-delay echoes, a situation that produces echo aliasing. Big brown bats have been shown to alternate between pulsing rapidly and pulsing slowly. Pulsing rapidly gives a clearer picture for close ranges while pulsing slowly gives a clearer picture for long ranges (Petrites et al., 2009). Another possible strategy might be to reduce the intensity of the call or reduce the low-frequency components of the chirp to reduce the distance over which the perceptible acoustic pulse travels. Bats have also been observed to change the spectral content of consecutive pulses, largely by shifting the entire pulse up or down in frequency. The spectral signature of the returned echoes can then be used to assign them to a specific pulse (Hiryu et al., 2010). This technique has also been used in radar (Gokturk et al., 2004; Skolnik, 2008) to increase the effective sampling rate. Another technique utilized by radar systems is to transmit multiple pulses in a short temporal pattern (or "code"). Different codes can then be used to identify different pulses (Skolnik, 2008; Matsuta et al., 2013). When the task is to track a specific target object (e.g., an obstacle the bat is maneuvering around), an attentional mechanism can be used to ignore the background and any aliasing that may be occurring. This approach works well until an "aliased" echo arrives at or near the time of the tracked echo. Three strategies for avoiding aliased echoes are presented: (1) a dynamic pulse-timing strategy that would allow a bat to "push" aliased echoes away from the attended window in time (adaptive delay), (2) changing the sonar "viewing angle" to the target to change the background (movement), and (3) using temporal phasing of two transducers during transmission to create an interference pattern in the sonar beam (with peaks and valleys) that can be used to isolate the target object (beam shaping).

MATERIALS AND METHODS

Hardware

The sonar system used in the work presented here consists of two custom modified MaxBotix® sonar transducers (shown in **Figure 1**), a custom PIC® 18F2620[1] (Microchip Technologies Inc.) (MaxBotix Inc., 2016) microcontroller-based sonar controller board, a Futaba S148 hobby servo, and a computer interface to both record and display echo signals and control the servo to orient the sonar. The transducers act as both a speaker and a microphone. They resonate specifically at 40 kHz and will only detect signals near this frequency. The custom sonar boards report a logarithmically-compressed envelope signal as an analog voltage. This allows the output to report the very wide dynamic range of amplitudes that occurs with sonar without saturating. The transducers are placed in a 3-D printed mount on the servo motor. In this demonstration system, the transducers transmit and receive over a cone of about ±30°, so the transducers are held facing 30° apart to ensure sufficient overlap and coverage of the area in front of the transducers for binaural localization. The ultrasonic pulse trigger-timing and analog-to-digital (A/D) conversion is done by the microcontroller. The majority of the data processing is performed on the microcontroller to ensure a quick response. Echo data is transferred via serial interface to a PC and the PC controls the servo motor via a USB-interfaced servo control board (Pololu, 2017).

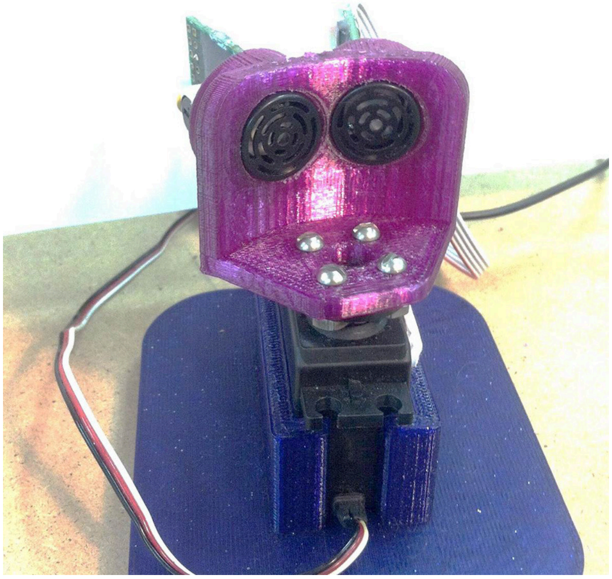


FIGURE 1 | The A two transducer sonar head mounted on a hobby servomechanism that was used for the experiments in this paper. The sonar modules are a custom 40 kHz system modified from a high-power Maxbotix sonar.

The Tracking Cycle

The sonar system executes four repeated steps: pulsing, sampling, processing, and communicating. As part of the cycle, there is an added delay interval that is used to reject aliased echoes (discussed in section Adaptive Delay). A few of these steps are shown in **Figure 2** for two cycles. In these examples, a short duration ultrasonic command pulse (∼0.25 ms) is used, however, due to the resonant quality of the transducer, the duration of the acoustic pulse is extended. Following the pulse, the transducer continues to ring for several milliseconds. Although echoes can be detected during this ringing period, their amplitudes are difficult to estimate, so a short 2 ms delay (i.e., dead-zone) is incorporated before sampling begins. The log-compressed envelope voltage is sampled every eighth of a millisecond. Object detection begins when the temporal derivative of the envelope exceeds a threshold of approximately 3.4 dB over an eighth of

[1] PIC18F2620. (n.d.). Retrieved March 21, 2016, Available online http://www.microchip.com/wwwproducts/en/PIC18F2620.

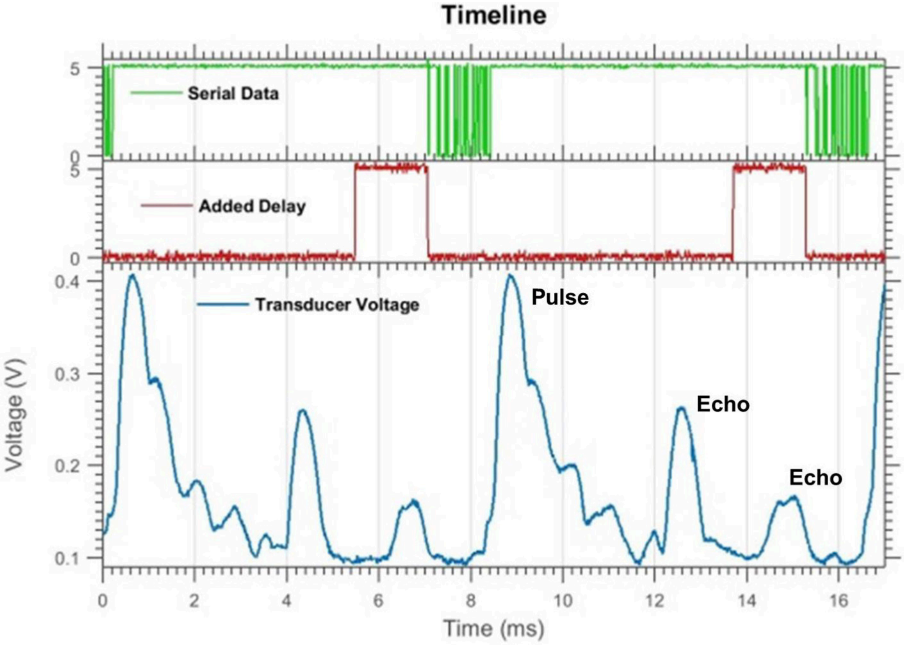


FIGURE 2 | An oscilloscope readout of two pulse-echo cycles (without aliasing) showing the transducer envelope voltage (**bottom**), serial data transfer (**top**), and added delay (**middle**). The added delay flag is set high when the delay is occurring. Objects can be seen as distinct peaks in the transducer voltage trace. Pulsing and sampling the transducer takes 5 ms, then there is a 1.5 ms delay and 1.5 ms of serial data transfer. The whole cycle takes about 9 ms.

a millisecond. Once the peak of the envelope has been reached, the object range is determined by the time since emission and the direction is estimated using the amplitudes on the two transducers.

At low pulse rates, the echoes are monitored for a period of time associated with the maximum range of the sonar and an extra delay would be added after transmitting the recorded data. In the case of fast pulsing where a target is being tracked, once the target echo is received, a short data burst is transmitted and the next cycle is initiated. After detecting the target echo, the tracking window (in time) is updated and the intensities of both transducers are compared to rotate the servo motor to center the target echo. At this time, temporal windows before and after the target echo are monitored to detect if other echoes are about to overlap with the target echo. This information is used to initiate the various reactive strategies to avoid interference with target tracking (described in sections Adaptive Delay, Movement, and Beam Shaping).

Target Tracking

There are occasions when the echo from the target disappears completely due to interference or occlusion by an object in the foreground. The tracker continues to search for the target at the same range for up to three cycles after the object disappears. If the object does not reappear, it will begin looking for a new target at a pre-specified acquisition range.

For the purposes of this study, the tracker is programmed to initially find the target at a single, pre-specified range (about 33 cm) and then follow it in range and in the horizontal plane by turning the sensor head to center the object. Centering is accomplished by rotating the sensor head until the detected amplitudes of the target in the two transducers are approximately equal. Only horizontal angles are considered. Since the echo amplitude is logarithmically compressed, the difference between left and right outputs corresponds to a ratio of the two received amplitudes. This ratio (invariant to echo amplitude) can be mapped to a specific angle. This mapping is defined by the spatial sensitivity and placement angles of the receivers and is found empirically. The ratio is monotonic and allows for reasonable angle measurements over a range of ±30°. Outside of this region, only one transducer will produce a significant response, allowing only a coarse approximation of direction. The response of our system at various angles is discussed in the Beam Shaping section.

The range of objects is determined by the time when the echo is received (i.e., time-of-flight). In practice, this is a very stable measurement that is minimally affected by noise. The echo amplitude, however, is very sensitive to factors such as the shape and orientation of the target, interfering reflections and echoes, and positioning of the transducers. At high repetition rates, a reverberant room can become filled with sound, introducing significant background interference. To avoid wild oscillations in the servo motor pointing, the system is restricted to moving a maximum step of 5° between echoes.

Once an object is found at the pre-specified acquisition range, it is labeled as the target and tracked. In the next pulse cycle, the sonar will expect to receive an echo within 6.3 cm of the previous target range. By restricting the temporal size of the tracking box, all echoes other than the target are ignored allowing the system to track a single object in the midst of other objects. Analog-to-digital sampling is performed with a period of an eighth of a millisecond and thus the range resolution is 2.1 cm/sample.

Aliasing and Clutter

In the rapid pulse mode, the maximum detection range for the sonar system is limited by the interpulse interval. If an object has an echo time that is greater than one pulse period, it is detected by the system in the next pulse cycle. It is then perceived as having an echo time that is one pulse period less than it actually is. Since this distortion is caused by sampling related to each pulse, we call it aliasing. This is demonstrated in **Figures 3**, **4**. While the perceived direction of this "phantom" object is unchanged, the range is wildly incorrect and may even overlap the echo from the tracked target. The techniques presented in this paper aim to keep the range and angle measurements of the target clean. This can be done by keeping other echoes far enough away (in time) to not overlap the target echo (∼ 0.5 ms). If that is not possible, the goal is to reduce the amplitude of the obstructing echo as much as possible.

Two strategies specific to problem of aliased echoes overlapping the target echo are presented: First, by using an **adaptive delay,** the interpulse interval can be manipulated to change the relative time of the aliased echo. This changes the perceived range of the alias to prevent it from overlapping with the target. Second, the sonar system can use **movement** to prevent objects in the background from falling in the main path of the sonar beam. This reduces the magnitude of clutter echoes.

These strategies may not always work, particularly if the aliased object is close in range to the target and the sonar beam is too wide for the movement strategy to avoid illuminating the aliasing object. In this case, **beam forming** of the transmitted pulse by firing both transducers in a phased manner can be used to increase the amplitude of the target echo and decrease the amplitude of the aliased object echo. This can also be effective

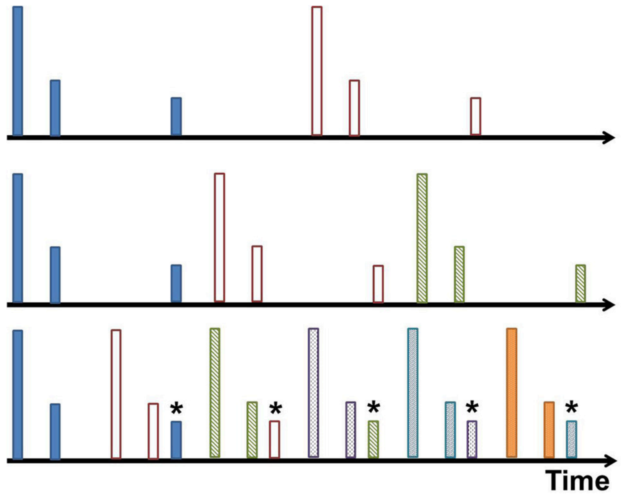


FIGURE 3 | Aliasing visualized. In this cartoon example, each timeline has pulses (represented by tall lines) and received echoes (represented by shorter lines). Each pulse and its echoes are given a unique color. From top to bottom, the interpulse interval decreases until a new pulse occurs before all echoes from the previous pulse are received, shown in the bottom timeline. The echo is misinterpreted as a closer object associated with the latest pulse. This is the aliased echo, and is labeled with an asterisk.

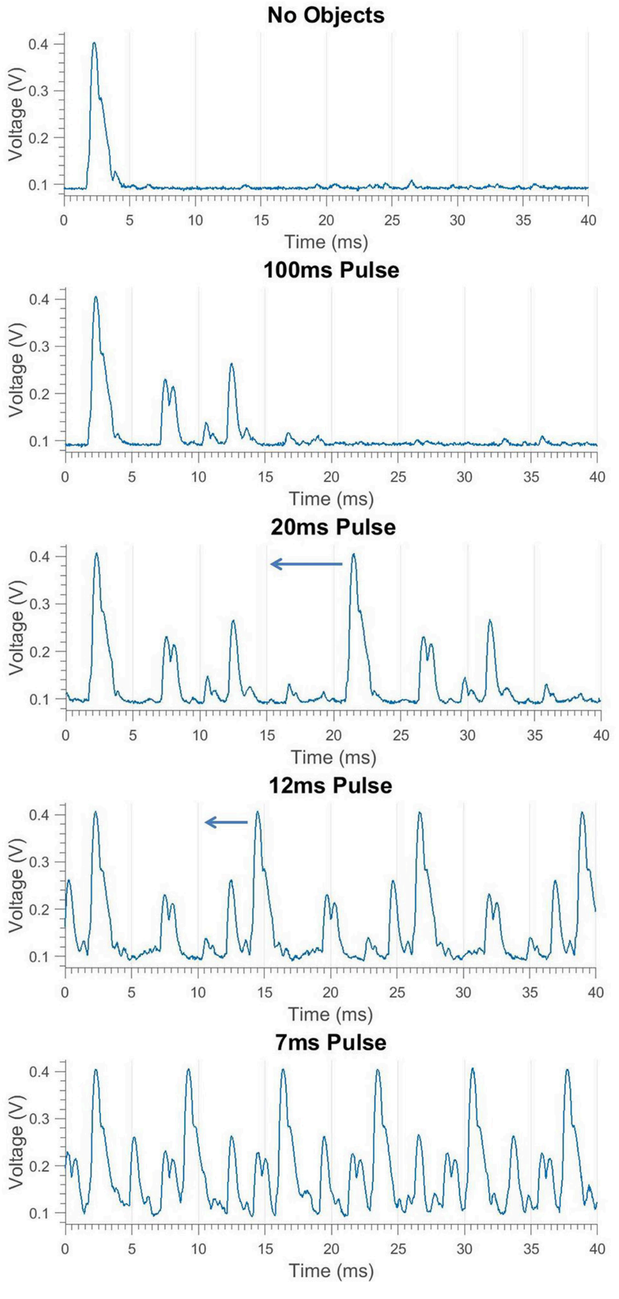


FIGURE 4 | Transducer envelope of pulses and echoes at different repetition rates demonstrating aliasing in the bottom graph. The outgoing pulse peaks at 0.4 V, overlapping echoes from two closely-spaced PVC pipes are seen peaking at 0.2 V, and a single loud echo made by a square poster board is seen peaking at 0.25 V. The interpulse interval is decreased in each graph until a new pulse occurs before all echoes from the previous pulse are received, causing an aliasing condition where the poster board incorrectly appears at short range.

in non-aliasing situations where a distractor object at the same range (but different angle) is causing interference.

Adaptive Delay

The range at which the aliased echo appears is dependent on the time between sending pulses. To control this, a variable delay period is inserted before sending the next pulse. Increasing this delay shortens the aliased echo time, making it appear to move closer to the sonar. Decreasing the delay increases the aliased echo time, making it appear to move away from the sonar (an example is shown in **Figure 5**).

The alias rejection system introduces a delay interval with a maximum of 3 ms into the timeline. The interval length is changed in eighth millisecond increments based on where the aliased echo appears relative to the tracked target. If an aliased echo is within 5 range samples, or 10.7 cm, of the target echo, the delay interval will be changed to repel the aliased echo. For an aliased echo that appears closer than the target echo (i.e., in between the target and the sonar system), we increase the delay to move the aliased echo away from the target echo; an aliased echo further away than the target echo decreases the delay. If the delay reaches its maximum amount or if it is decreased to zero, the delay value is reset to 1.5 ms (half of its maximum value). This will cause an aliased echo to jump to the other side of the target echo, being shifted by 12 range samples. If there is an aliased echo detected on both sides of the target, the delay is shifted by a large amount, equivalent to 11 range samples, in an attempt to clear both aliased echoes away from the target echo. This process is summarized below.

```
If alias in front
    Increase delay
If alias in back
    Decrease delay
If alias in front and alias in back
    Large delay shift
If delay is minimum or delay is maximum
    Reset delay
```

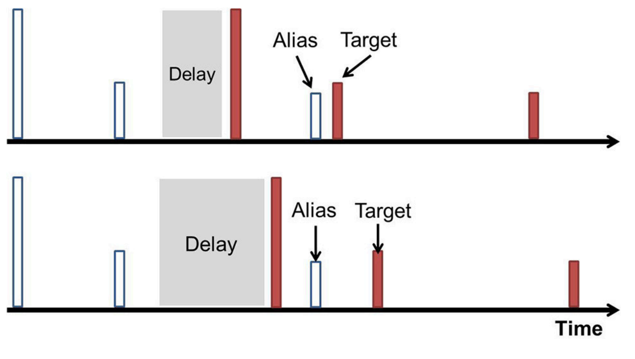


FIGURE 5 | Manipulating the received time of an aliased echo. The tall line represents the pulse and the short lines represent the echoes. The echoes associated with a given pulse are the same color. The top timeline shows an alias (white) that is close to interfering with the first dark echo, the target. The introduced delay is increased (in the bottom timeline) to shift this aliased echo away from the target in time. Similarly, an alias on the other side of the target can also be shifted away by decreasing the delay (not shown).

Checking for Real Objects

While we have assumed a relatively isolated target object to track, a real second object in close proximity to the target cannot be "rejected." In this case, the alias rejection system would continuously shift the delay, resulting in oscillations of the delay shifting and resetting when the delay interval reaches its limits. To prevent these oscillations, additional code is used to recognize authentic (i.e., non-aliased) echoes.

The most notable difference between an authentic echo and an aliased echo is their reaction to a large shift in the interpulse interval, a delay jump. An alias will be moved a significant amount, while an authentic echo will not be moved at all. Although a real object can still move noticeably, at low speeds (<3 m/s) it will not jump more the one range sample at a time.

The alias rejection system makes large delay shifts in three different scenarios: when the delay interval reaches its maximum, its minimum, and when two aliases sandwich the target (one on either side). The system uses these events as triggers to look for an authentic echo that remains in the same location. This is especially appropriate since an authentic echo triggers an oscillation that causes the delay to jump when the interval reaches a maximum or minimum. If an object doesn't move after a delay jump, it is recognized as an authentic echo and will not activate the alias rejection system. This is similar to a technique used in radar where a map of stationary clutter is memorized and removed (Skolnik, 2008).

Movement

An alternative method to avoid sonar aliasing is to reposition the sonar beam such that objects in the background do not generate echoes. The effectiveness of this technique will depend on using a relatively narrow transmission beam. Depending on the species of bat, transmission beam widths can range from 22 to 90° (Jakobsen and Surlykke, 2010; Nachtigall and Moore, 2012; Matsuta et al., 2013). The sonar beam width used in this study is approximately 30°.

When the sonar moves around, different sides of objects are exposed to the sonar. In general, this will complicate a decision to change the sensing angle, since the acoustic properties of an object can change greatly from different perspectives. To demonstrate this, two different objects were used as the aliasing object in two different trials: a large 46 cm (1.5 ft) diameter cardboard tube and a 30 cm wide, open cardboard box. The sonar was moved around a target object to continue aiming the beam at the target at the same range, but resulting in different backgrounds (**Figure 9**). As the sonar moves, the transmission beam is moved away from the aliasing object and the magnitude of its echo decreases. Theta is the angle of rotation the sonar system has made around the target relative to its starting location. For this study, only one transducer was used.

Beam Shaping

A third strategy for reducing the effect of aliasing and clutter objects is to shape the acoustic beam so that only the target object

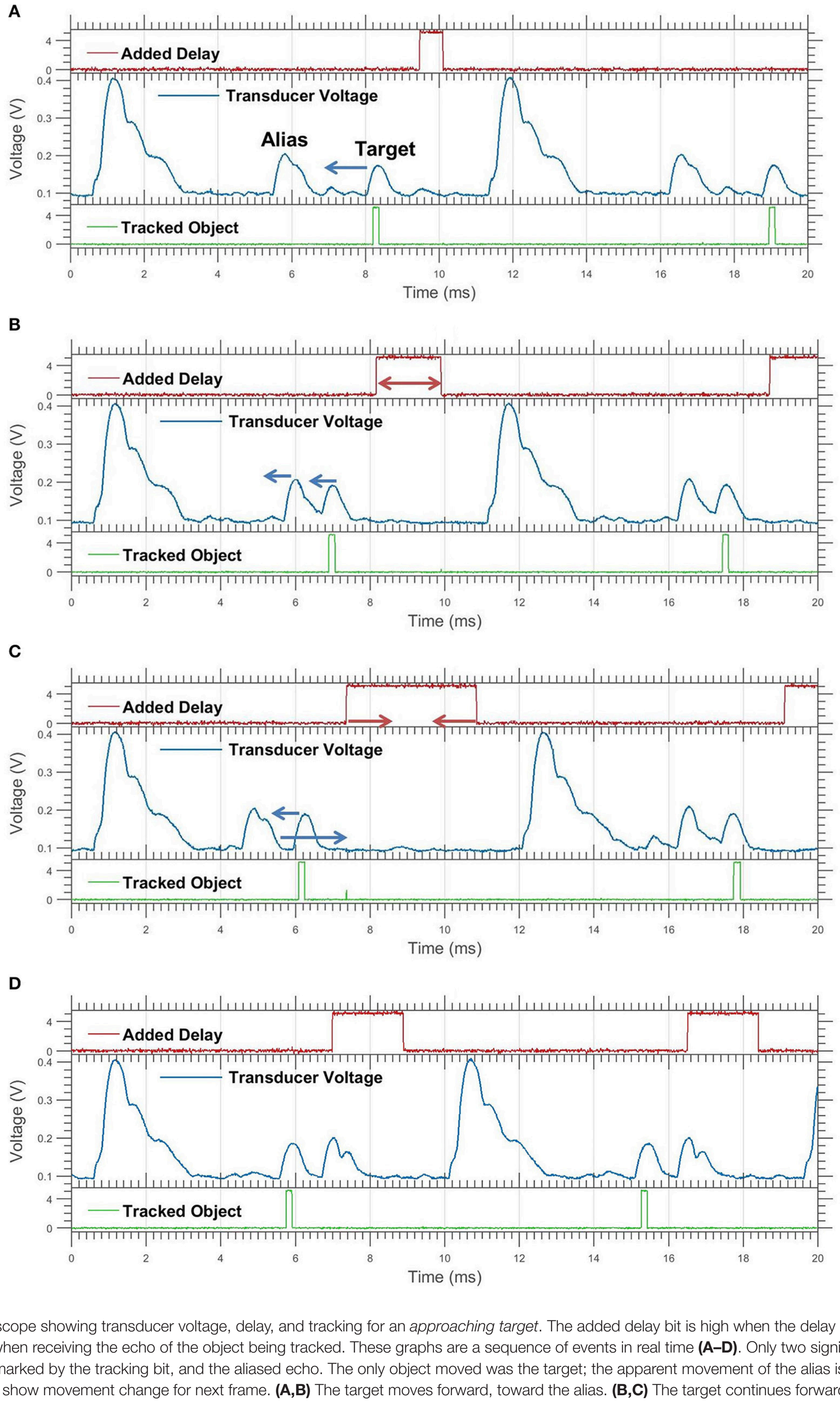


FIGURE 6 | Oscilloscope showing transducer voltage, delay, and tracking for an *approaching target*. The added delay bit is high when the delay is occurring. The tracking bit is high when receiving the echo of the object being tracked. These graphs are a sequence of events in real time **(A–D)**. Only two significant objects are present, the target marked by the tracking bit, and the aliased echo. The only object moved was the target; the apparent movement of the alias is due to the delay change. The arrows show movement change for next frame. **(A,B)** The target moves forward, toward the alias. **(B,C)** The target continues forward, the alias is pushed forward by the increasing delay. The delay buffer becomes maximized. **(C,D)** The delay buffer jumps down after reaching its maximum. This causes the alias to "jump" behind the target.

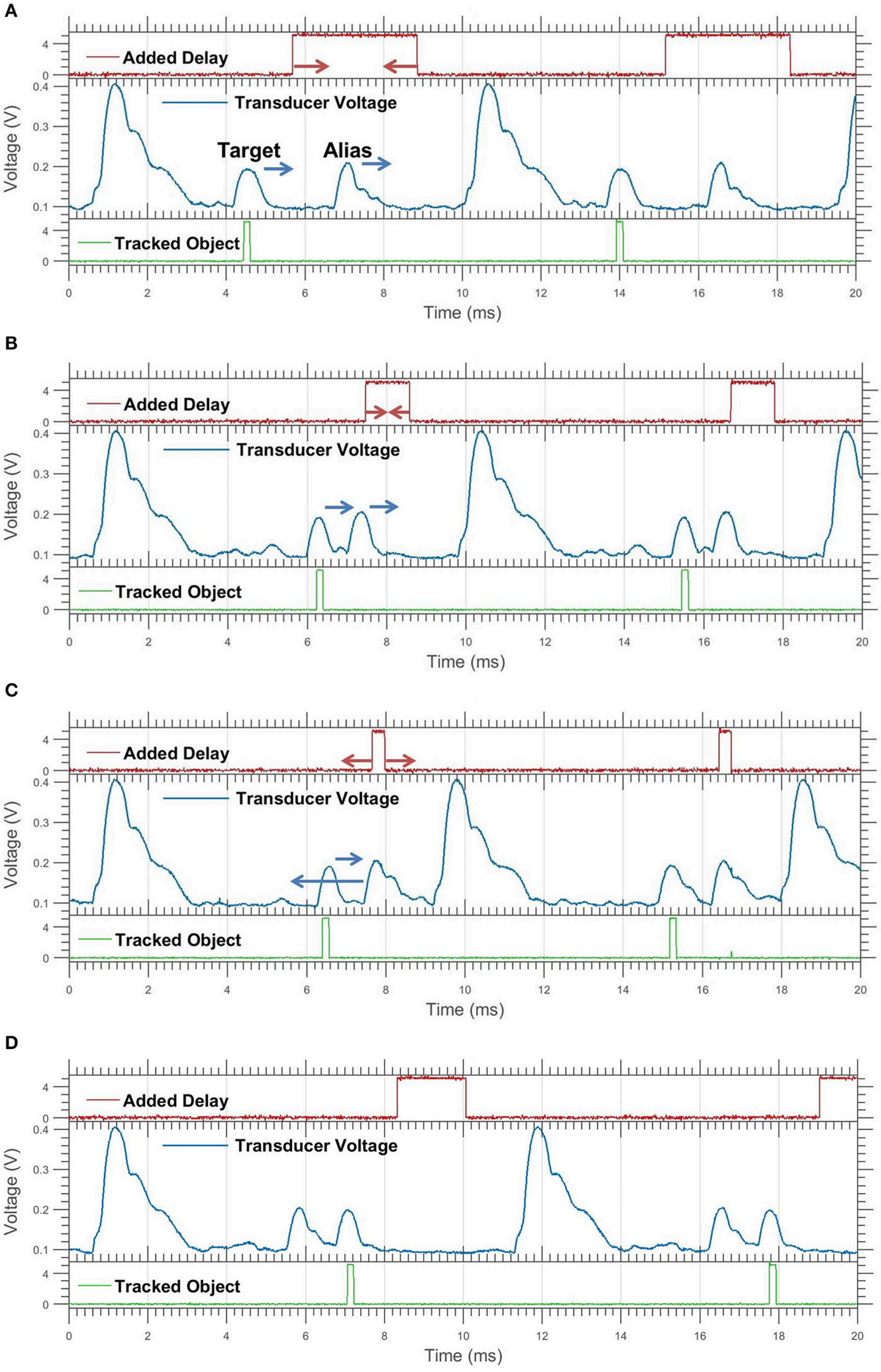


FIGURE 7 | Continuation of **Figure 6** for a *retreating target.* **(A,B)** The target moves back; the alias is pushed back by the decreasing delay. **(B,C)** Both echoes continue backwards, the delay buffer reaches its minimum value. **(C,D)** The delay buffer jumps upwards, causing the alias to jump forwards in front of the target.

is ensonified. With the two-transducer system used in the study, this is performed by transmitting with both transducers to create an interference pattern that has peaks and nulls that can be used to reduce interference. Plots of the beam shape are shown for a single transducer, the two transducers firing synchronously, and the two transducers firing out-of-phase (**Figure 11**). The synchronous in-phase firing pattern has a loud frontal lobe that is relatively narrow with weaker lobes on either side. The −6 db width of the front lobe is 19° (compared to 62° of a single transducer alone). The stronger, narrower central lobe would allow more precise ensonification of a target while reducing echoes from other directions. It is important to note that the patterns presented here represent the transmitted beam only. The sonar hardware presented here does not allow phased detection; although that is an additional capability in other systems that would further improve selectivity.

Figure 12 shows an example of how using this firing pattern can affect an echo trace. There are two objects in the field of view, both PVC pipes of equal diameter. Using only a single transducer without phasing, the clutter echo impinges on the target echo and disrupts the information conveyed. Using two transducers fired synchronously, the cluttered object has a significantly reduced magnitude and the target will not be as affected by the clutter. This example, however, represents a best-case scenario where the clutter object falls in the low trough of the firing pattern. Utilizing this system in arbitrary object configurations is not trivial. Since angle estimation is very noisy in this sonar system, predicting the effects of beam shaping can be error-prone and not guaranteed to be beneficial.

The experimental configuration used is shown in **Figure 13**. With the target centered in the sonar view, the clutter object is moved to different angles relative to the center. Both objects are 5 cm diameter PVC pipes at a range of 122 cm (4 ft).

RESULTS

Adaptive Delay

Figures 6–8 show the system in action, presenting consecutive graphs in time that demonstrate system functionality. The three figures represent the three different cases of aliases: aliases moving toward the target from the front, aliases moving toward the target from the back, and two aliases sandwiching the target. In each case it is assumed that the target starts clear and unobscured. The adaptive delay prevents the target from becoming obscured in all the cases.

Movement

Figure 10 shows the results of the movement study with the aliasing object at 4, 5, and 6 ft (labeled x) away from the target. At the same angle, larger x values push the aliasing object farther away from the center of the beam and cause a larger reduction in magnitude. For the column, at a 60° angle of rotation, the aliased echo amplitude had been reduced to below 19% of the target echo amplitude for all distances of x. For the box at the same angle, the amplitude was only reduced to 57% of the target echo amplitude in the worst case ($x = 4$ ft). This highlights the role of the object geometry. It should be noted that the measured amplitudes are logarithmically-compressed acoustic amplitudes and the actual percentage change seen will vary with signal level.

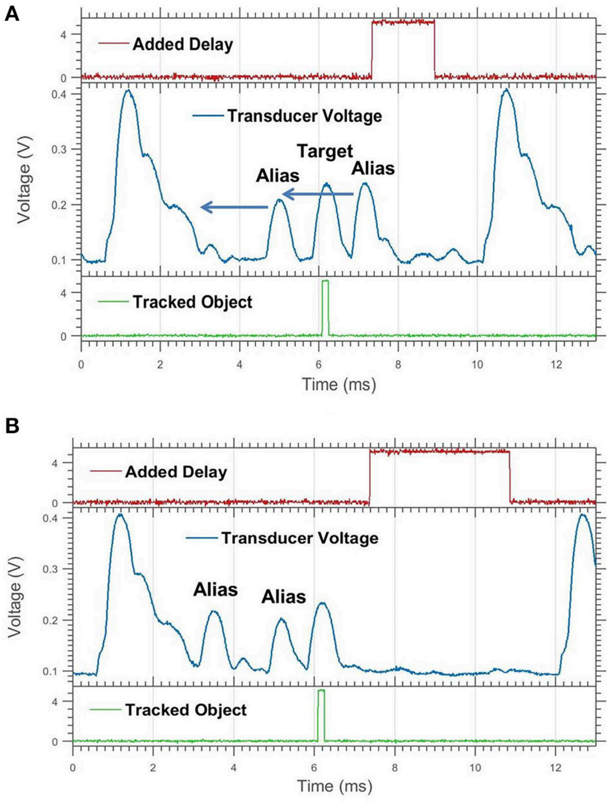


FIGURE 8 | A target being sandwiched by aliased echoes. **(A)** Here the alias in front of the stationary target approaches the target and triggers a delay jump. **(B)** This clears both aliases away from the target, drawing them both forward of the target.

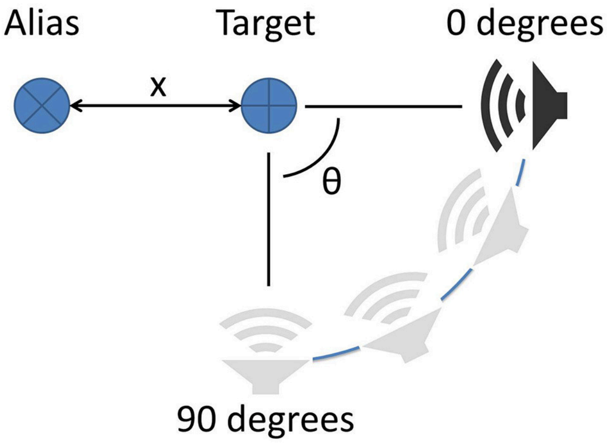


FIGURE 9 | Alias rejection via movement. The sonar system (speaker) is kept a constant distance from the target. The alias is located at a distance x from the target. The sonar is rotated around the target by angle θ to shift the view of the system.

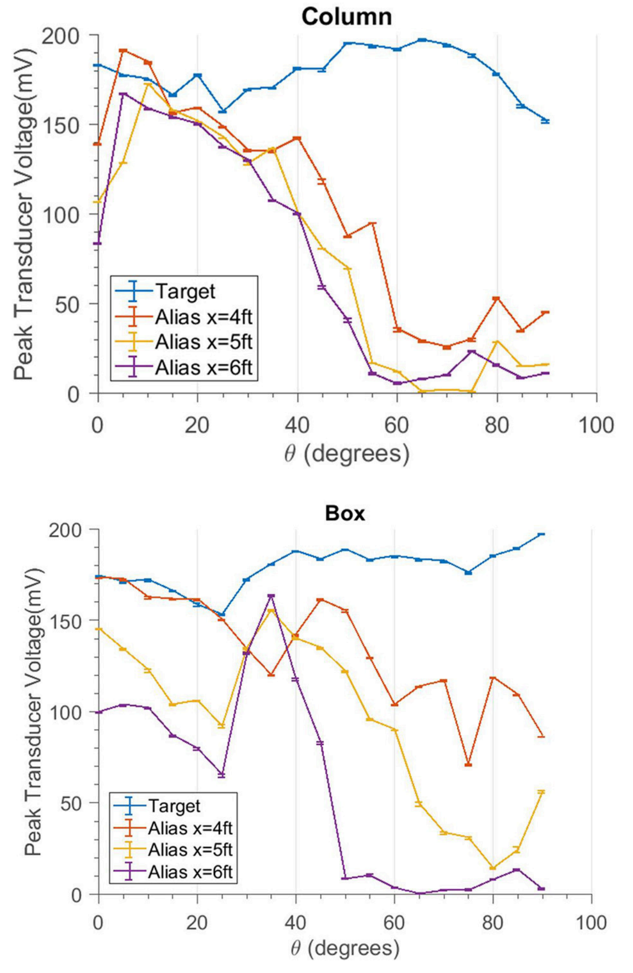


FIGURE 10 | Traces showing the echo response of the target and the alias at different angles. The target trace (blue) gives a baseline for comparison. The "Alias" traces reduce in amplitude as the angle increases. For larger distances *x* the amplitude decreases even more.

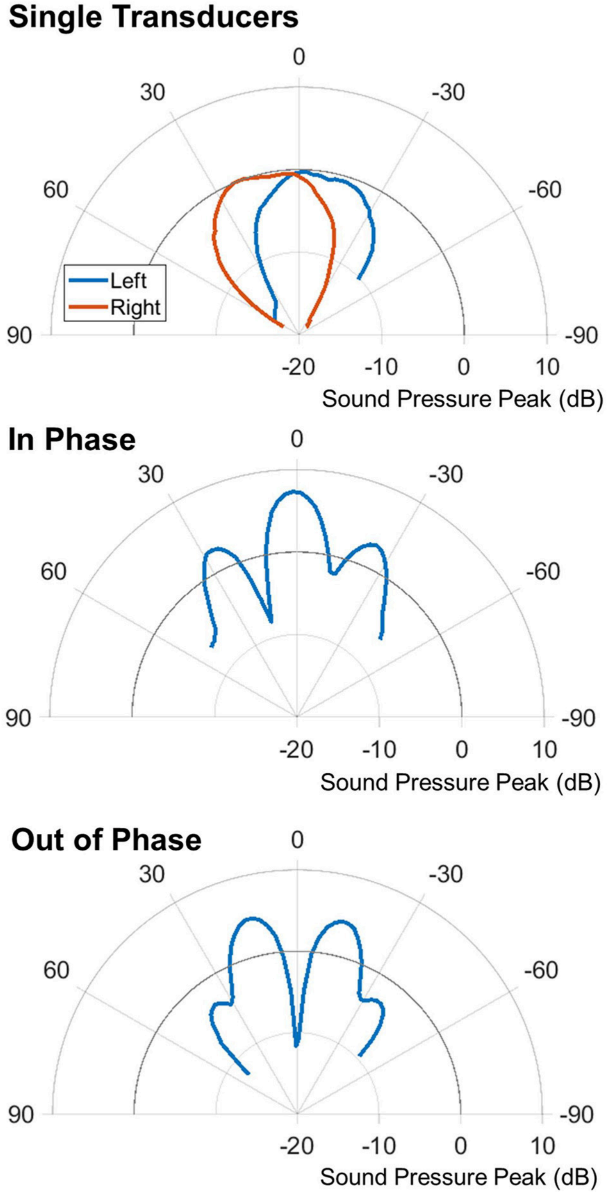


FIGURE 11 | Polar plots of the different firing patterns. Top shows single transducer pulses from the left and right transducers. Middle shows the synchronous in-phase firing pattern. Bottom shows the synchronous out-of-phase firing pattern.

Beam Shaping

For the beam shaping study, the results are shown in **Figure 14**. The target to clutter (amplitude) ratio is used to normalize the data, which accounts for the difference in magnitude of the different firing patterns. The simulated data was created using the echoes from one real PVC pole recorded across all of the angles. The center measurement is used as the target amplitude; all other angles are treated as clutter amplitudes. The target to clutter ratio is calculated between the center and all other angles.

The synchronous firing pattern has a higher target to clutter ratio than the left or right transducers alone. This only occurs for angles less than 18°. This is due to the side lobes of the interference pattern; once the clutter starts to enter these lobes it is no longer sufficiently rejected and a single transducer will yield a better target to clutter ratio. In between 6 and 18°, where the most benefit is seen, there is a 3.39 dB average increase in the signal to clutter ratio with the synchronous firing pattern compared to the next best single transducer.

DISCUSSION

Adaptive Delay

The adaptive delay system for alias rejection tackles a problem that most engineered sonar systems avoid at the cost of a lower sampling frequency. When overlapping echolocation cycles are unavoidable, some form of pulse labeling is most commonly used (Uppala and Sahr, 1996; Gokturk et al., 2004; Skolnik, 2008; Hiryu et al., 2010; Matsuta et al., 2013). These techniques remove the issue of pulse-echo ambiguity since every pulse has

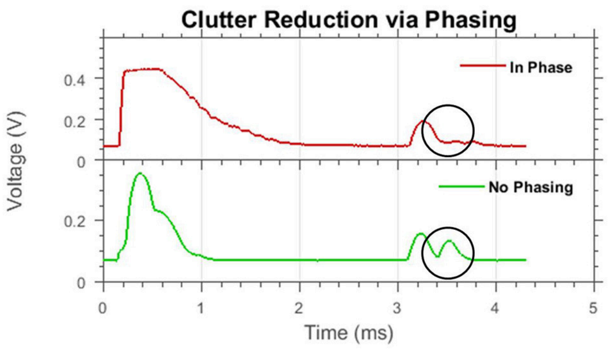


FIGURE 12 | A best-case example of clutter reduction using beam shaping. Shown are two echo traces from the same scene with different beam shapes. Two objects are present, the first echo is the target object (∼3.2 ms); the second echo is from the clutter object (∼3.5 ms) which is circled. When in-phase firing is used, the clutter echo is greatly reduced in amplitude.

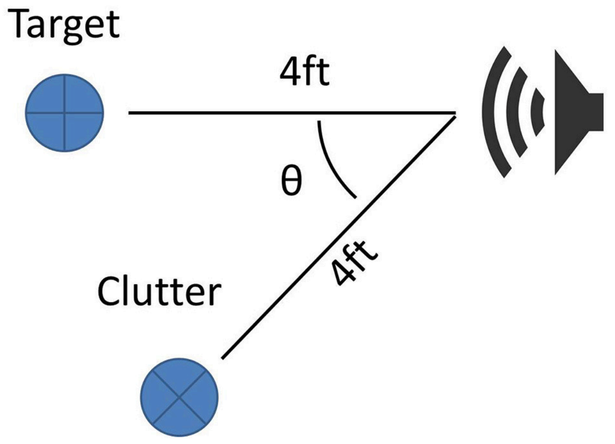


FIGURE 13 | Clutter rejection using beam shaping. The sonar system faces a target that is 4 ft away. The clutter object is also 4 ft from the sonar but is rotated around the sonar system, changing its angle in the view of the sonar.

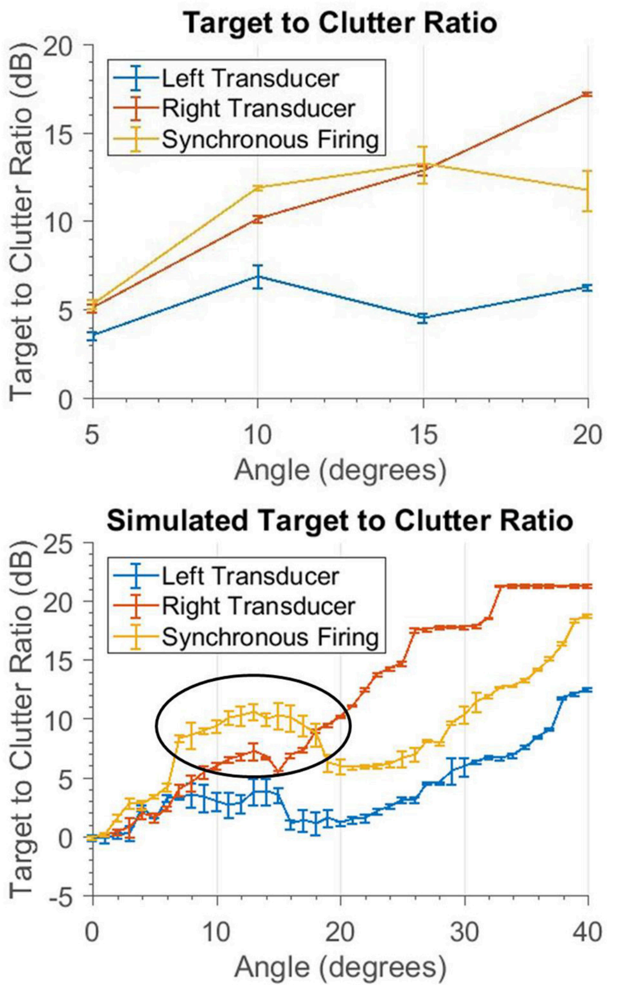


FIGURE 14 | These graphs show how a clutter object appears at different angles. The target and clutter objects are at the same range. Only the angle to the clutter object is changed. The top graph shows the ratio of the target and clutter amplitude. The bottom shows simulated data, where only one object was scanned across all of the angles. The ratio was computed using the echo at angle 0 (i.e., the target) and the other angles (i.e., the clutter object). The circled area shows that for angles less than 18° the synchronous firing has better clutter rejection.

its own unique characteristic. The approaches presented here are unique in that the pulse-echo ambiguity remains and tracking is maintained in spite of it. This allows a much simpler, single frequency system to be more useful.

The biggest limitation of the adaptive delay system is that it can only deal with a small number of aliased echoes. The case when two aliases sandwich the target is dealt with, but if three or more aliases occur in the right spots, there may be no delay time that prevents the target from being obscured.

Movement

The movement strategy is much different from the other strategies since it cannot be done on a pulse to pulse basis. Moving the sonar to improve sensing also impacts the decisions of navigation that the sensing is intended to facilitate. These results provide more information to consider by the navigation system that must balance sensing and overall task goals. The basic geometry and the angular response of the sonar system suggest that lateral movement with respect to orientation of the sonar is most effective. Another consideration is that any change in sensing angle may, in fact, generate new aliasing problems as it turns to include new background objects. Note that this approach (like the pulse timing method presented in section Adaptive Delay) will have little to no effect for clutter objects that appear at the same range as the target.

Beam Shaping

This technique is a useful way to reduce the effect of aliasing and is the only strategy presented here that is also potentially effective for objects at ranges similar to the target. It is most effective for small angles off-center. Synchronous firing creates a loud central lobe down the central axis of the sonar head. This allows for objects at longer ranges to be detected. This study did not utilize the out of phase firing primarily because the target is assumed to be held in the center of view. The out-of-phase transmission pattern has its minimum in the center of view. If a different tracking algorithm was used that kept the offending clutter in the center, this firing pattern could also be useful in rejecting clutter.

This kind of interference pattern has also been observed to be used by certain bats (Hartley and Suthers, 1987). *Carollia perspicillata* emits sound from two nostril holes. These two nostril holes appear to interact in the same way as depicted in the sonar system above.

Combining the Strategies

While these three strategies have been presented and considered separately, they can be combined into an integrated approach. Adaptive delay and beam shaping can be used simultaneously; the delay can be changed independently of the beam shape. Movements to specifically reduce aliasing can also be made, although other factors will likely affect what actions are taken.

If an alias is detected, the adaptive delay approach can be used to prevent the target from being obscured. At the same time, a movement direction can be suggested based on the apparent angle of the alias. If the obstructing echo is determined to be a real object and not an alias (part of the adaptive delay code), then different beam shapes can be used depending on the apparent angle of the obstructing echo. If the angle is less than 18°, synchronous firing will be used. If the angle is greater than 18°, only one transducer will be used. This approach is summarized in **Figure 15**.

These strategies complement each other well. Together, they present a multi-pronged approach for dealing with the interference produced while using high pulse repetition rates. Each strategy is suited to a different situation and need not be used simultaneously.

CONCLUSION

Three different active strategies for dealing with echo aliasing are described that can allow the use of sonar at high sampling rates in cluttered environments. Although a time-domain attentional system is assumed to be able to focus on a specific range to track objects, echoes from clutter objects can overlap in time, obscuring or confusing such an attentional system. At very short interpulse intervals, echoes from the background arriving after the next pulse appear to be at a shorter range then they actually are. These “aliases” can overlap the target and interfere, causing a failure of the tracking system. A dynamic pulse-timing strategy is proposed that can effectively “push” or “pull” the aliased echoes away from the tracked target echo by decreasing or increasing the interpulse interval. This prevents aliases from interfering with tracking. We have also presented a method of avoiding or reducing aliases based on positioning, as well as a method of shaping the echolocation beam to reduce the effect of aliasing or clutter.

Bats have been shown to use several different strategies when encountering cluttered situations that require fast sampling. They have been observed to change the frequency content of consecutive pulses (Hiryu et al., 2010), alternating between short and long pulses (Petrites et al., 2009), and using the directionality of certain harmonics to focus in a given direction (Bates et al., 2011). The system presented here operates on a single carrier frequency, so frequency-based techniques for clutter rejection were not explored, however, we have shown that other techniques are possible (pulse timing, flight steering, and beam shaping) and are possibly also in use by echolocating bats.

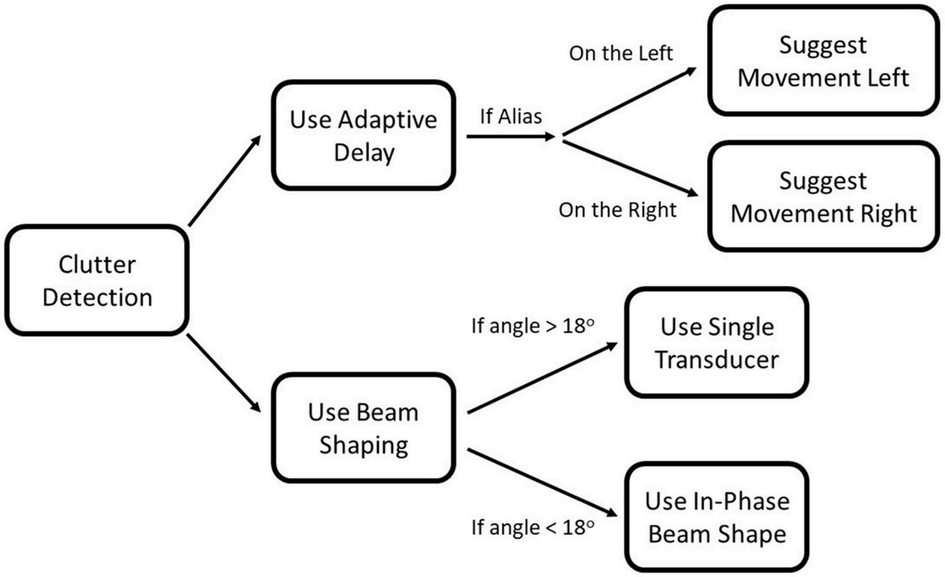


FIGURE 15 | Flowchart for integrating the three strategies. Once clutter is detected, beam shaping, and the adaptive delay can be used simultaneously. If the adaptive delay determines that the object is an alias, a movement direction will be suggested.

AUTHOR CONTRIBUTIONS

TH: created the hardware and software template for the sonar system; JI: created the sonar mount and modified the software for this project; JI: performed the data collection and analysis with advice and guidance from TH. All authors discussed the results and contributed to the final manuscript.

FUNDING

This work was support by grants from the U.S. Office of Naval Research (N000141210339), the U.S. Air Force Office of Scientific Research (FA9550-14-1-0398, Center of Excellence), and the U.S. National Science Foundation (SMA1540916, Neuromorphic Engineering Workshop).

REFERENCES

Bates, M., Simmons, J., and Zorikov, T. (2011). Bats use echo harmonic structure to distinguish their targets from background clutter. *Science* 333, 627–630. doi: 10.1126/science.1202065

Gokturk, B., Yalcin, H., and Bamji, C. (2004). "A time-of-flight depth sensor-system description, issues and solutions," in *Computer Vision and Pattern Recognition Workshop, CVPRW'04. Conference on IEEE, 2004* (Washington, DC).

Grothe, B., Pecka, M., and McAlpine, D. (2010). Mechanisms of sound localization in mammals. *Physiol. Rev.* 90, 983–1012. doi: 10.1152/physrev.00026.2009

Hartley, D., and Suthers, R. (1987). The sound emission pattern and the acoustical role of the noseleaf in the echolocating bat, *Carollia perspicillata*. *J. Acoust. Soc. Am.* 82, 1892–1900. doi: 10.1121/1.395684

Hiryu, S., Bates, M. E., Simmons, J. A., and Riquimaroux, H. (2010). FM echolocating bats shift frequencies to avoid broadcast–echo ambiguity in clutter. *Proc. Natl. Acad. Sci. U.S.A.* 107, 7048–7053. doi: 10.1073/pnas.1000429107

Jakobsen, L., and Surlykke, A. (2010). Vespertilionid bats control the width of their biosonar sound beam dynamically during prey pursuit. *Proc. Natl. Acad. Sci. U.S.A.* 107, 13930–13935. doi: 10.1073/pnas.1006630107

Matsuta, N., Hiryu, S., Fujioka, E., Yamada, Y., Riquimaroux, H., and Watanabe, Y. (2013). Adaptive beam-width control of echolocation sounds by CF–FM bats, *Rhinolophus ferrumequinum* nippon, during prey-capture flight. *J. Exp. Biol.* 216, 1210–1218. doi: 10.1242/jeb.081398

MaxBotix Inc. (2016). *High Performance Ultrasonic Rangefinders.* Available online at: http://www.maxbotix.com/ (Accessed March 21, 2016).

Nachtigall, P., and Moore, P. (eds.). (2012). *Animal Sonar: Processes and Performance.* Helsingor: Springer Science & Business Media.

Petrites, A., Eng, O., Mowlds, D., Simmons, J., and DeLong, C. (2009). Interpulse interval modulation by echolocating big brown bats (*Eptesicus fuscus*) in different densities of obstacle clutter. *J. Comp. Physiol.* 195, 603–617. doi: 10.1007/s00359-009-0435-6

Pololu (2017). *Mini Maestro. (n.d.).* Available online at https://www.pololu.com/product/1352 (Accessed October 31, 2017).

Schmidt, S., Yapa, W., and Grunwald, J. (2011). Echolocation behaviour of Megaderma lyra during typical orientation situations and while hunting aerial prey: a field study. *J. Comp. Physiol.* 197, 403–412. doi: 10.1007/s00359-010-0552-2

Skolnik, M. (2008). *Radar Handbook 3rd Edn*. McGraw-Hill Education.

Uppala, S., and Sahr, J. (1996). Aperiodic transmitter waveforms for spectrum estimation of moderately overspread targets: new codes and a design rule. *IEEE Trans. Geosci. Remote Sens.* 34, 1285–1287. doi: 10.1109/36.536545

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers
in Neuroscience

ORIGINAL RESEARCH
published: 06 March 2018
doi: 10.3389/fnins.2018.00140

Extraction of Inter-Aural Time Differences Using a Spiking Neuron Network Model of the Medial Superior Olive

Jörg Encke and Werner Hemmert*

Bioanaloge-Informationsverarbeitung, Department of Electrical and Computer Engineering, Technical University Munich, Munich, Germany

The mammalian auditory system is able to extract temporal and spectral features from sound signals at the two ears. One important cue for localization of low-frequency sound sources in the horizontal plane are inter-aural time differences (ITDs) which are first analyzed in the medial superior olive (MSO) in the brainstem. Neural recordings of ITD tuning curves at various stages along the auditory pathway suggest that ITDs in the mammalian brainstem are not represented in form of a Jeffress-type place code. An alternative is the hemispheric opponent-channel code, according to which ITDs are encoded as the difference in the responses of the MSO nuclei in the two hemispheres. In this study, we present a physiologically-plausible, spiking neuron network model of the mammalian MSO circuit and apply two different methods of extracting ITDs from arbitrary sound signals. The network model is driven by a functional model of the auditory periphery and physiological models of the cochlear nucleus and the MSO. Using a linear opponent-channel decoder, we show that the network is able to detect changes in ITD with a precision down to 10 μs and that the sensitivity of the decoder depends on the slope of the ITD-rate functions. A second approach uses an artificial neuronal network to predict ITDs directly from the spiking output of the MSO and ANF model. Using this predictor, we show that the MSO-network is able to reliably encode static and time-dependent ITDs over a large frequency range, also for complex signals like speech.

Keywords: spatial hearing, medial superior olive, computational model, artificial neural network, binaural model

OPEN ACCESS

Edited by:
Mounya Elhilali,
Johns Hopkins University,
United States

Reviewed by:
Edward Lee Bartlett,
Purdue University, United States
Leslie Samuel Smith,
University of Stirling, United Kingdom

****Correspondence:***
Jörg Encke
joerg.encke@tum.de

Specialty section:
This article was submitted to
Auditory Cognitive Neuroscience,
a section of the journal
Frontiers in Neuroscience

Received: *26 October 2017*
Accepted: *21 February 2018*
Published: *06 March 2018*

Citation:
Encke J and Hemmert W (2018)
Extraction of Inter-Aural Time
Differences Using a Spiking Neuron
Network Model of the Medial Superior
Olive. Front. Neurosci. 12:140.
doi: 10.3389/fnins.2018.00140

1. INTRODUCTION

Our remarkable sound localization acuity relies on the ability of the auditory system to decode the arrival time and intensity difference between the ear canal signals into information about the direction of sound sources. In mammals, the primary nucleus to extract fine structure interaural time differences (ITDs) is the medial superior olive (MSO), while the interaural level differences (ILDs) are extracted primarily at the lateral superior olive (LSO) (Grothe et al., 2010). The MSO neurons detect fine-structure ITDs by acting as coincidence detectors receiving excitatory inputs from both hemispheres. The existence of such neurons was already hypothesized by Jeffress (1948), who proposed an array of coincident detectors to be arranged along a neural delay line. In this hypothesis, each neuron would respond maximally to a specific ITD (best-ITD)—generating a topographical mapping of time differences within the nucleus. Later, such a circuit was found in the nucleus laminaris of birds like the barn owl (Carr and Konishi, 1988). However, more recent measurements of mammalian inferior colliculus (IC) and MSO neurons in gerbils (Brand et al., 2002) or guinea pigs (McAlpine et al., 2001) revealed broadly-tuned neurons, of which the majority had their best-ITDs at the border or even outside of the animals physiological range. This observation is inconsistent with place-code theory, which would require a vast amount of narrowly-tuned neurons with their best-ITDs distributed within the physiological range. One alternative ITD-coding mechanism is based on the comparison of firing rates between the nuclei in the two hemispheres. This mechanism has consequently been called the opponent-channel (Magezi and Krumbholz, 2010), count-comparison (Colburn and Durlach, 1978), or hemifield (Stecker et al., 2005) model. The opponent-coding model is in agreement with both observations, the wide tuning curves and the large best-ITDs (McAlpine and Grothe, 2003). There is also evidence that overall sound localization (Stecker et al., 2005; Briley et al., 2012) as well as specifically ITD-coding in the human auditory cortex is based on an opponent coding mechanism (Salminen et al., 2010). Lesion studies in cats showed that unilateral lesions at the level of the central auditory system (Jenkins and Masterton, 1982) as well as in cortical regions (Malhotra et al., 2004) mainly resulted in deficits localizing sounds from locations contralateral of the lesion. These results lead Jenkins and Masterton (1982) to conclude that each auditory-hemifield is represented solely in the respective contralateral hemisphere, which would contradict the opponent coding mechanism. One problem with applying this interpretation to ITD processing is that both studies used broad-band stimuli so that ITDs and ILDs, as well as spectral and monaural cues were available to localize the sound source this makes it difficult to draw conclusions about the representation of the individual cue. An alternative to the opponent-channel code, which uses the summed response of the neurons within each of the two hemispheres, is the population decoder that instead uses the individual response of each neuron for decoding. Based neuronal recordings of neurons in the IC, Goodman et al. (2013) and Day and Delgutte (2013) both proposed population decoders and showed that these decoders could outperform a two-channel decoder. On the other hand, Harper et al. (2014) used an optimal coding approach to show that ITDs in low-frequency signals would be best represented by a two-channel code. Additionally, results from psychoacoustic lateralization experiments using pure-tone adapter stimuli with fixed ITDs showed, that adaptation influences lateralization at ITDs not only close to that of the adapter but within the whole hemisphere (Phillips et al., 2006), which is more in line with an opponent-channel code.

The aforementioned remarkable sound localization ability has inspired numerous researchers to create computational binaural models. Most of the existing binaural models are phenomenological implementations of the delay-line principle proposed by Jeffress (1948), which have been tuned to successfully predict data from human psychoacoustics (Lindemann, 1986). Some more recent models were implemented following the opponent-coding mechanism (Pulkki and Hirvonen, 2009; Dietz et al., 2011; Takanen et al., 2014). Even though these models closely follow the functionality of the neuronal sound localization pathway, they provide only a phenomenological description of the processing stages. On the other hand, several biophysical models of MSO neurons have been published as well (Brughera et al., 1996, 2013; Zhou et al., 2005; Lehnert et al., 2014), but there are only a few biophysical models covering the complete neuronal circuit. Wang et al. (2013) used a circuit containing a model of the auditory periphery as well as spiking models of the MSO and LSO and a simplified IC model to investigate the sensitivity of IC neurons to envelope ITDs in high-frequency sounds. Due to the focus on high-frequency sounds where ITDs are extracted from the envelope of the sound signal instead of its fine structure (Nuetzel and Hafter, 1976), Wang et al. (2013) did not include any source for a shift in best-ITD and also neglected inhibitory inputs to the MSO. Glackin et al. (2010) presented a spiking neural network (SNN) constructed from leaky integrate-and-fire models of the CN and MSO nuclei. In disagreement with newer physiological studies, the SNN was constructed as a Jeffress-type delay-line decoder. Glackin et al. (2010) trained the network to localize the sounds using spike-timing-dependent plasticity learning rules.

To our knowledge, none of the previous models combined an SNN approach with the concept of opponent-coding to investigate ITD sensitivity. Brughera et al. (2013) presented a single spiking neuron model of the MSO to investigated ITD sensitivity, but used a periodic Poisson-like process as an input to the MSO. This limits the model to simple pure-tone-like scenarios while also neglecting any non-linear processing of the auditory periphery. To that end, we present here a new binaural model based on biophysical spiking neuron models of the mammalian MSO circuit. We show that a simple linear hemisphere decoder applied to the output of the model is sufficient to encode ITDs in tones with a precision that matches human performance. Furthermore, we show how the model in conjunction with a simple artificial neural network can decode ITDs from broadband signals, including complex signals like speech.

2. RESULTS

2.1. Model Structure

The primary mammalian MSO neurons receive excitatory inputs from spherical bushy cells (SBCs) as well as inhibitory inputs from the globular bushy cells (GBCs) of the cochlear nuclei in both hemispheres. Inhibitory inputs are being relayed via the trapezoid body (TB) (see Grothe et al., 2010 for an overview). Both SBCs and GBCs are directly excited by auditory nerve fibers (ANFs). GBCs in particular, but also SBCs have been found to enhance phase locking of the neuronal inputs (Joris et al., 1994; Dehmel et al., 2010). Our model consists of three stages, a model of the auditory periphery, a population of globular bushy cells and a population of MSO neurons (see **Figure 1**). For simplicity, SBC as well as the TB nuclei, were reflected as direct relays of the ANF signals so that our MSO model receives direct excitatory

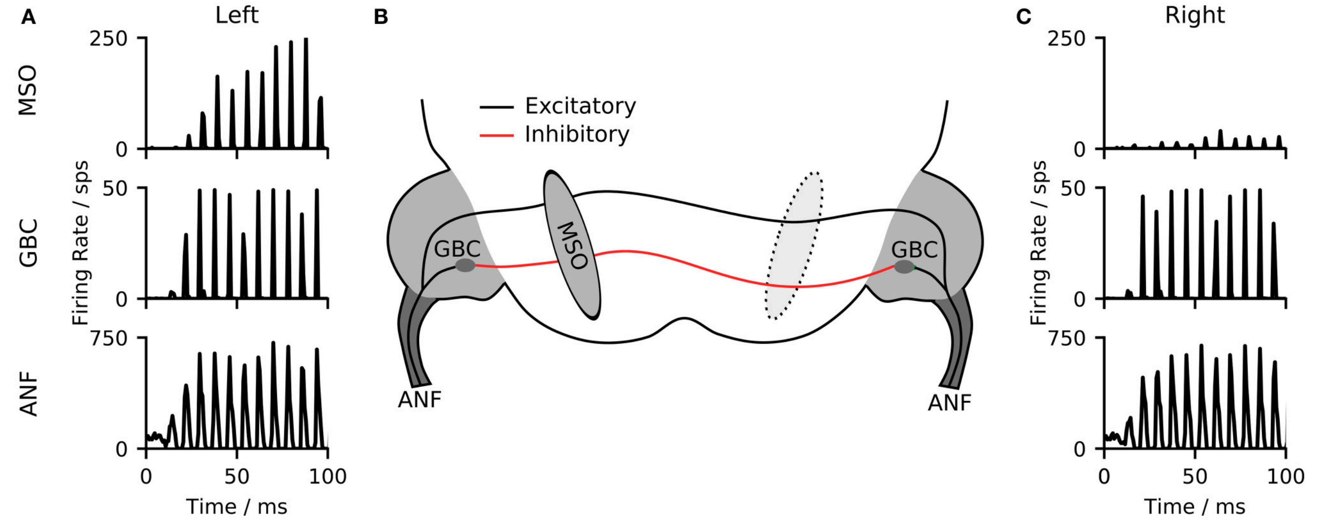


FIGURE 1 | (A,C) Poststimulus time histograms (750 µs bin size) of the responses of the three model stages to a 100 ms long pure tone. **(B)** The model network contains three stages. A model of the auditory periphery (ANF), A model of the globular bushy cells in the cochlear nucleus (GBC), and the model of the medial superior olive (MSO).

input from the ANF and inhibitory inputs from GBCs of both hemispheres (see section 4 for details on the implementation). In practice, our model takes digitized binaural signals as input and processes them first through the peripheral hearing models of the left and right ears. The peripheral model consists of a middle-ear compensation filter, a non-linear model of the basilar membrane and a functional model of the neural transduction of the inner hair cell and auditory nerve fibers (Zilany et al., 2014). All ANFs were modeled as high spontaneous rate units. The spike timings of the peripheral hearing models were then used as input to the biophysical neuron models. As a consequence of the direct excitation by ANF fibers, the frequency responses of both MSO and GBCs resemble that of the ANFs from the peripheral hearing model (see Figure S1).

As an example of the output from the different model stages, **Figures 1A,C** illustrate the outputs of ANFs, GBCs, and the MSO of the two hemispheres for a left-leading (150 µs ITD) 125 Hz pure-tone input. The ANFs of both hemispheres show a phase-locked response to the input stimulus. This phase-locked response is sharpened by the population of GBC neurons. The MSO neurons of the two hemispheres respond with different firing rates depending on the delay between the signals delivered to the left and right ear.

Most MSO neurons of gerbils show bell-shaped ITD-rate functions with their maximum (best-ITD) located outside of the animals physiological range (Brand et al., 2002). There has been much debate about the origin of this shift ranging from intra-cochlear delays (Joris et al., 2006) over asymmetric synaptic currents (Jercog et al., 2010) to effects of the recent stimulus history (Franken et al., 2015). Our model is based on the effect described by Brand et al. (2002) and Pecka et al. (2008), who showed that blocking of the inhibitory inputs results in a shift of the best-ITD toward zero. Measurements in gerbil brain slices have also shown that inhibitory inputs to the MSO precede the excitatory inputs in time (Roberts et al., 2013). Using conduction clamp measurements, Myoga et al. (2014) showed that the relative timing of inhibitory to excitatory inputs to the MSO can delay or advance the peak of the excitatory post-synaptic potential (EPSP) and consequently, affect the best ITD of the neurons. Our model is consistent with these findings. In agreement with Brand et al. (2002) and Pecka et al. (2008), the best-ITD shifted toward zero when simulating the effect of blocked inhibition by reducing the inhibitory synaptic strength (see **Figures 2A,B**). Similarly, and in accordance with Myoga et al. (2014), we could shift the best-ITD of the MSO model by adjusting the delay of contra- and ipsilateral inhibitory inputs. For the model used in later evaluations, we optimized both arrival times to obtain a maximal shift of the best-ITD toward contra-leading ITDs. This optimization resulted in a delay of 0.6 ms for the contralateral inhibitory input and 0 ms for the ipsilateral input (both values relative to the timing of the excitatory input from the corresponding side). These values are in agreement with the timescales observed by both Myoga et al. (2014) and Roberts et al. (2013). The study by Pecka et al. (2008) showed a residual shift of the best-ITD even when the inhibitory inputs were blocked. This could be explained by fundamental physics as the axons connecting inputs from the contralateral hemisphere to the MSO have to span over a larger distance than the ones for ipsilateral inputs. We considered this observation by adding a constant delay of 100 µs to the contralateral excitatory and inhibitory inputs, which resulted in an additional shift of the best-ITD toward negative values (see **Figure 2B**).

2.2. Decoding ITD Information From the Neuronal Responses

The opponent-coding theory is based on two populations of neurons, both firing maximally when the sound source is on the opposite side of the midline (Stecker et al., 2005). **Figure 3A**

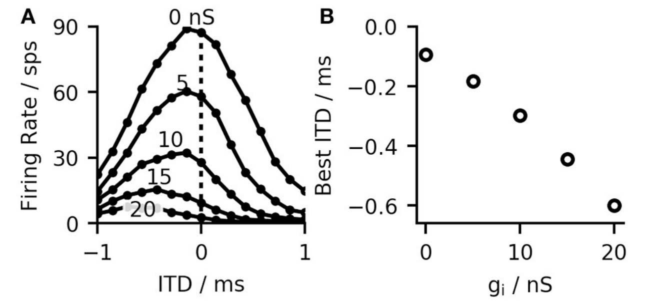


FIGURE 2 | (A) MSO ITD-rate functions (calculated for 15 ITDs in the range ±1 ms) for the right hemisphere of the model at different inhibitory conductivities $\hat{g}_{syn,i}$ **(B)** Increased inhibition, reduces the overall firing rate and shifts the best-ITD toward more contra-lateral leading ITDs. Without inhibition the best-ITD equals the predefined shift of 100 μs.

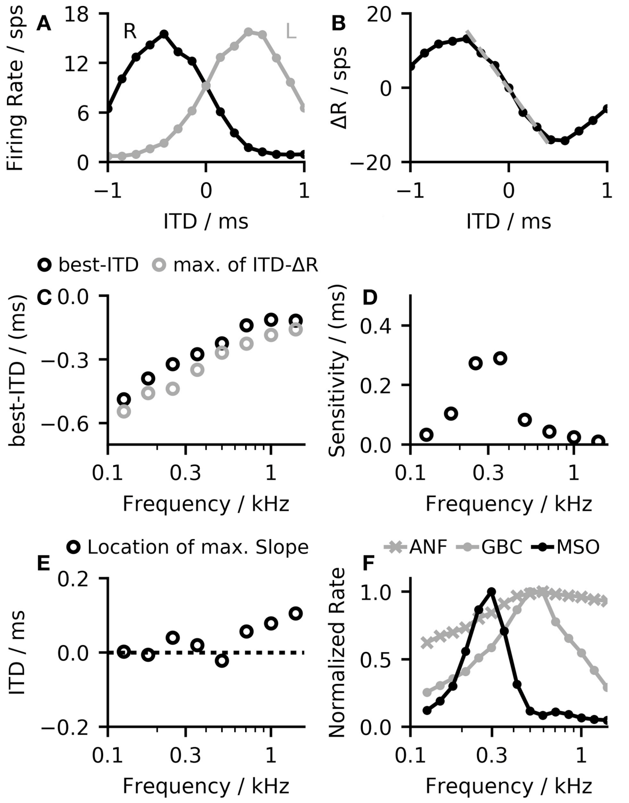


FIGURE 3 | (A) The model demonstrates ITD-rate functions for the left and right MSO that are effectively mirrored around zero so that a shift in ITD from the center line leads to an inverse response of the two channels forming the basis for the opponent-channel code (Stecker et al., 2005). **(B)** A simple difference computation between the ITD-rate functions of the two hemispheres resulted in a nearly linear relationship around zero ITD (dashed line). **(C)** The shift in the best-ITD decreases with increasing best-frequency of the MSO neuron, reaching the predefined shift of 100 μs at about 1.4 kHz. **(D)** The slope of the rate difference curve around midline can be seen as a sensitivity to changes in ITD. This value changes with the best frequency. The sensitivity peaks at 300 Hz from where it decreases again toward higher frequencies. **(E)** The single hemisphere responses showed the largest change at or close to the midline and therefore maximizes the sensitivity of the linear-decoder. **(F)** Frequency-dependent normalized firing rates of the three neuron populations in the model in response to a 100 ms pure tone at 50 dB.

shows firing rates of the MSO model in both hemispheres to a stimulation with varying ITDs. The left MSO responds strongest when the stimulus was right-leading (positive ITD), while the right hemisphere responds strongest to a left-leading ITD (negative value). Consequently, a change in ITD from zero results in an increased firing of one MSO and a reduced firing of the other. A very basic decoder for the opponent-channel code can be constructed by subtracting the firing rates of the left MSO (R_L) from the right MSO (R_R). Around zero ITD, the calculated firing rate difference $\Delta R = R_R - R_L$ shows an almost linear response to ITD changes (see **Figure 3B**). Due to the subtraction, this approach increases the slope around zero by a factor of two and consequently maximizes the sensitivity in this region. However, this approach is applicable only for ITDs for which the linear approximation is valid. The linear ITD region depends primarily on the location of the best-ITD (see **Figure 3C**) in the two hemispheres. When calculating ITD-rate functions for neurons with different best-frequency, the best ITD decreases with increasing sound frequency (see **Figure 3C**). The best-ITD is maximally 470 μs at 125 Hz and decreases to 110 μs at 1.4 kHz. The same trend of decreasing best-ITDs with increasing frequency has been found in *in-vivo* recordings of MSO neurons (Brand et al., 2002; Pecka et al., 2008) as well as in the IC (McAlpine et al., 2001). As aforementioned, this model mainly uses phase-locked inhibition to shift the best-ITD. This method relies on the slopes of the inhibitory post-synaptic potentials (IPSPs) of each phase (Myoga et al., 2014). At higher frequencies, the summation of individual IPSPs reduces the effectiveness in shifting the best-ITD (Roberts et al., 2013; Myoga et al., 2014), which is also seen in the model results. Experimental studies have shown that MSO and IC neurons exhibit a variety of different best-ITDs (McAlpine et al., 2001; Bremen and Joris, 2013), while in this model, all neurons with the same best frequency also show the same best-ITD. As this study does not use a population decoder but relies on the mean activity within each hemisphere, the single ITD-rate function can also be interpreted as the mean ITD-rate function of a single hemisphere.

The sensitivity of the linear-decoder to ITD changes is proportional to the slope of the ΔR function around zero ITD—a steeper slope results in larger changes. As the slope of the ΔR function around zero is twice the slope of a single hemisphere response, maximizing the slope of the single hemisphere will also result in a maximal slope of ΔR . Pecka et al. (2008) and McAlpine et al. (2001) both reported the maximal slope of single neuron responses to be located at or close to mid-line. In this model, responses to frequencies up to 700 Hz followed these findings (see **Figure 3E**). At higher frequencies, the location of the largest slope started to shifted away from midline as the best-ITD decreased faster than the width of the ITD-tuning function, which shifted the location of the largest slope toward positive ITDs. A second influencing factor on the sensitivity is the maximum firing rate of the MSO response—a higher rate of the single hemisphere responses will also result in a larger slope at midline. **Figure 3F** shows the frequency dependent normalized firing rates of all three neuron populations in the

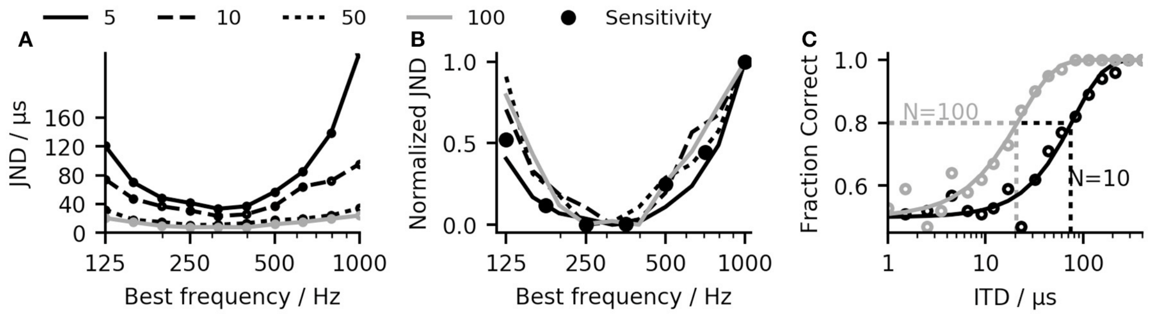


FIGURE 4 | (A) Frequency-dependent just-noticeable differences (JNDs) calculated for subsets of 5, 10, 50, and 100 randomly chosen neurons among a population of 500 neurons. **(B)** When normalized to a region between 0 and 1, all JND curves overlap. **(C)** The JND values were calculated by fitting weibull-functions to the fraction correct values (see section 4).

model. The firing rate of the MSO model is of course strongly influenced by the balance between the excitatory inputs from the ANFs and the inhibitory inputs from the GBCs but it is additionally modulated by changes of the spiking thresholds. MSO neurons have been found to exhibit subthreshold resonance (Remme et al., 2014; Mikiel-Hunter et al., 2016) which introduces frequency dependent thresholds. The MSO model used in this study exhibited a resonance frequency at about 260 Hz (see Figure S2) which is in agreement with the resonance frequencies found in electrophysiological studies (Remme et al., 2014; Mikiel-Hunter et al., 2016). The reduced spiking threshold around 260 Hz in combination with the dynamics of the synaptic inputs results in a peak in MSO response seen in **Figure 3F**, which also corresponds to the peak in sensitivity shown in **Figure 3D**.

While applying the linear-decoder does not directly result in an ITD estimate, it can be used to predict ITDs. The link between ITD and ΔR also allows for a direct comparison of the laterality of two signals with different ITDs without the necessity to map the MSO model response to the absolute ITD estimates. This highlights the difference between an absolute localization task, which requires the mapping of the auditory perception to a spatial measure and a relative comparison task where the relative location of one perception in comparison to a second perception is reported. In psychoacoustical experiments, the sensitivity to ITDs is often assessed by determining the just noticeable differences (JNDs) which describe the smallest change in ITD a subject can use to detect a change in lateralization between the two otherwise identical stimuli (Klumpp, 1956). Using the same method, we calculated JNDs for our network model using the linear-decoder (see section 4). In our model, the performance depends critically on the number of neurons composing the population, as the intrinsic stochasticity of the neuronal system loses its impact on the average firing rate when the population increases. To determine the influence of the population size on the performance of our model, JNDs were calculated separately for subsets of 5, 10, 50, and 100 randomly chosen neurons among a population of 500 neurons. **Figure 4C** shows exemplary psychometric curves derived for a population of 10 and 100 neurons. **Figure 4A** shows the result of the JND experiment for different pure tone stimuli. As expected, the predicted JND decreases when increasing the size of the population. The decrease in JND can be described by a $1/\sqrt{N}$ dependency, where N is the population size. The dependence is in line with the reduced effect of noise due to a larger population of neurons. If the JND thresholds are determined mainly by the noise of the system, they should also be reflected in the sensitivity described by the slope of the ΔR function. **Figure 4B** shows the JND curve as well as the inverse of the slope of ΔR with all values normalized to lie between 0 and 1. As expected, there is a good agreement between the normalized JND curves and the inverse of the slope, which confirms the aforementioned assumption that the detection threshold of the linear-decoder depends mainly on the slope of the rate-difference function around zero ITD.

One problem of such a linear-decoder is that the firing rates of the two MSO models depends not solely on the ITD, but also on other characteristics of the inputs to the MSO model. As the firing rate of the peripheral hearing model depend strongly on the sound pressure level, so will the output of the MSO model. To demonstrate such dependency, **Figure 5C** shows how the predicted sensitivity of our model varies with both frequency and level of the pure tone input. The ANFs also exhibit strong spike-rate adaptation (Smith, 1977) which, consequently affects the MSO response (**Figure 5A**). These variations could be compensated by normalizing the ITD-ΔR functions (overlay in **Figure 5B**) but this is not possible in practice as it would require *a priori* knowledge about the maximum firing rate of the ITD-ΔR function at each point in time. A much more practical approach is to compensate such non-linear dependencies using the information that is already encoded in the ANF firing rates.

The MSO exhibits a distinct tonotopic organization along its dorsoventral axis. As the neuronal populations along the axis differ in their characteristic frequency (Guinan et al., 1972), consequently, a given ITD decoder can specialize on decoding of ITDs within a specific frequency range. In addition, the non-linear and time-dependent output of the peripheral hearing process can be compensated by using direct knowledge about the firing rates of the ANF. However, implementing such corrections would require designing a complex multi-dimensional correction function. Artificial neuronal networks (ANN) have been proven to be quite successful in learning the behavior of highly nonlinear

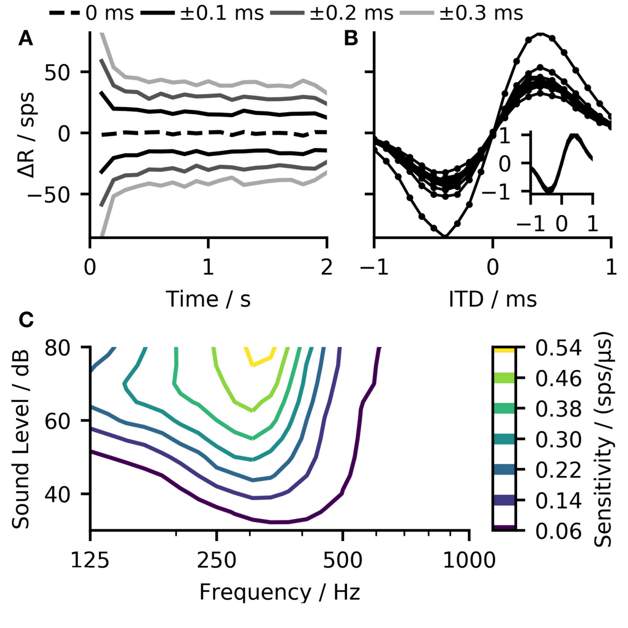


FIGURE 5 | (A) The ΔR function (calculated for 100 ms bins) of 500 MSO neurons in response to a two second long 250 Hz pure tone with ITDs ranging from 0 to ±0.3 ms show a strong influence of the peripheral hearing model's adaptation on the MSO model output. **(B)** The same data as in **(A)** but shown in form of ITD-ΔR functions. Every function corresponds to one point in time evaluated for 20 ITDs in the range from −1 to 1 ms. The difference in the slope of these functions illustrates that adaptation influences the sensitivity of the linear-decoder . Normalization with respect to the maximal rate of each function could compensate for this influence (overlay) **(C)** The model displays strong variations in sensitivity with sound frequency and sound level.

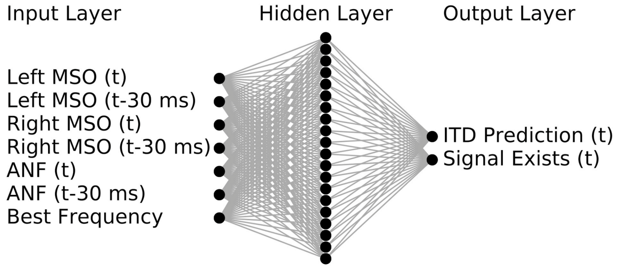


FIGURE 6 | The ANN predictor was implemented using seven input nodes, one hidden layer with twenty nodes and two output nodes. Inputs denoted with (t) are firing rates within a given time period (typically 30 ms) for which the ITD should be predicted, while the ones denoted with (t-30 ms) are firing rates of the preceding time frame.

systems (Almeida, 2002), hence, they provide an appealing alternative to tedious manual construction of a correction function.

2.3. Artificial Neuronal Network Predictor

We used a small multi-layer perceptron (MLP) to predict ITD values from the output of the SNN model by means of non-linear regression analysis. The regression is based on the average firing rates across the neuronal populations and predictions are calculated separately for each frequency band and time frame. The MLP was implemented using seven input nodes, one hidden layer with twenty nodes and two output nodes (for details see section 4). One of the MLP output nodes was used for the prediction task, while the second output was used to classify the reliability of the prediction based on the firing rates. This was deemed necessary to omit predictions for parts of the input signal, which did not contain enough energy in the given frequency band to enable robust predictions based on sufficient spiking activity.

The inputs to the MLP were designed to consist of the firing rates from the MSO of the left and right hemisphere and the characteristic frequency of the neuron population (see **Figure 6** for a schematic of the networks in- and outputs). As one of the main tasks of the predictor was to compensate for the influence of variations in the peripheral hearing model output, the MLP was also provided with a monolateral input of the ANF firing rate. All firing rates were provided as an average value computed over a predefined time period of 30 ms. This duration was chosen as it offered reasonably high temporal resolution and ensured that several periods of the phase locked input were included. In addition to the rates within the given time frame, we also provided firing rates of the previous time frame which reduced the noise in the predictions by effectively doubling of the time span that the network can employ in its predictions. The MLP was trained on 300 ms long pure tones (see section 4) covering the frequency range from 125 to 1,000 Hz so that predictions can be obtained for any stimuli within that range. For the following experiments we calculated predictions for 13 logarithmically spaced frequencies between 125 and 1,000 Hz.

Figures 7A–C compares the results of the ANN-predictor with those of the linear-decoder for an amplitude-modulated tone with 400 Hz carrier frequency and a modulation rate of 2 Hz. Since amplitude modulation is encoded in the firing rate of the ANF, it is also exhibited in the output of the linear-decoder (**Figures 7A,C**). On the other hand, the predictions from the ANN (**Figure 7B**) showed only minor deviations at the on- and offsets of each modulation cycle while largely compensating the strong onset response introduced by ANF's adaptation. **Figure 7B** shows only such predictions that the ANN classified to be reliable. In case of the amplitude-modulated signal, the frequency bands for which the ANN could predict ITDs are dependent on the phase of the modulation.

Omitting unreliable predictions enables the calculation of a general prediction across frequency bands. In case of the linear-decoder, zero output can correspond to two conditions—zero ITD and no signal. The employed method of omitting unreliable estimates is especially important for applying the ANN predictor to more complex signals that have several frequency components because the omission enables the ANN to predict ITDs without prior knowledge about the signal's frequency content. **Figures 7D–F** show examples of the ANN-predictor applied to a linear chirp. To demonstrate the ability of the predictor to follow changes both in frequency as well as in ITD, an additional phase shift was applied to the left ear

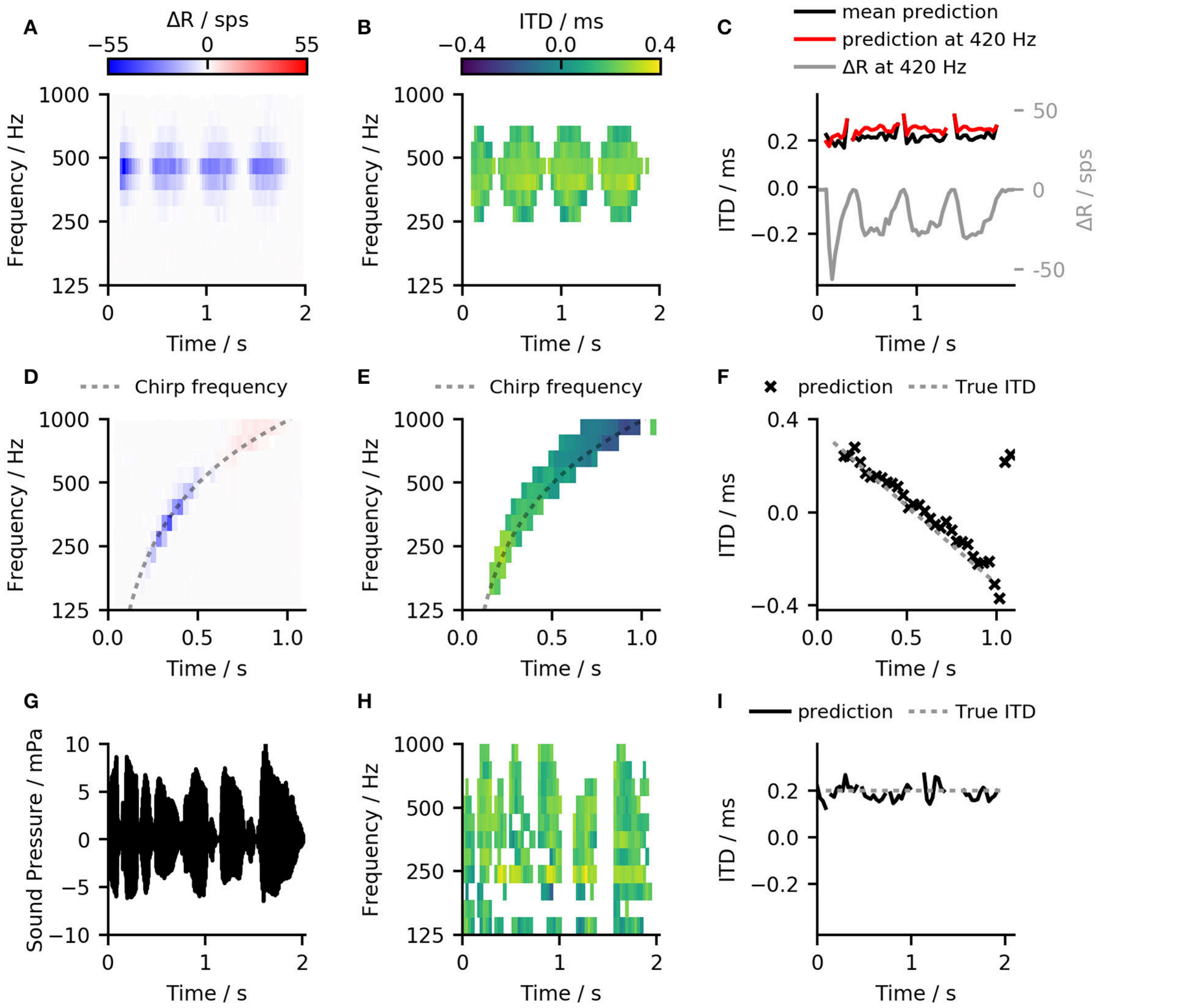


FIGURE 7 | (A) Results of the linear-decoder for an amplitude-modulated tone with 400 Hz carrier frequency and a modulation rate of 2 Hz presented with an ITD of 200 ms. ΔR showed strong modulation with the modulation frequency of the sound as well as an influence of ANF adaptation **(B)** Results of the ANN-predictor predictor for the same signal as in **(A)**. The ANN was able to correct for the variations conveyed by the ANF inputs and to provide a stable prediction within the frequency bands from 250 Hz to 595 Hz. **(C)** The output of both, the ANN-predictor and the linear decoder for the amplitude modulated signal over time. Red: ANN predictions for the 420 Hz channel which was the closest to the stimulation frequency. Black: Mean over all predictions that were classified to contain a useful signal. Gray: Result of the linear-decoder in the 420 Hz frequency band. **(D,E)** Same plots as in **(A,B)** but for a linear, one-second long chirp ranging from 125 to 1 kHz where the ITD changed from −0.4 to 0.4 ms. **(F)** The ANN-predictor was able to follow the change in frequency as well as in ITD, deviating from the true value only at the end of the signal. **(G–I)** The ANN-predictor applied to a speech signal (German sentence "Britta gewann drei schwere Steine") taken from the OLSA sentence test (Wagener et al., 1999).

signal. This phase shift was chosen to be proportional to an ITD-value that varied linearly from +300 to −300 μs. By calculating a cross-frequency prediction for every time frame, the ANN-predictor was able to follow the change in frequency as well in ITD (**Figure 7C**) deviating from the true value only in the last two time frames. As a final example, we show the ANN-predictor applied to a speech signal with a static ITD of 200 µs (**Figures 7G–I**). Again, the across-frequency estimation, combined with the omission of unreliable predictions allows the ANN-predictor to offer an accurate estimate of the ITD for the whole signal.

3. DISCUSSION AND CONCLUSION

In this study, we presented a novel binaural model and used it to detect ITDs in arbitrary sound signals. In contrast to previous binaural models that used a phenomenological modeling approach (Pulkki and Hirvonen, 2009; Dietz et al., 2011), this study used biophysical neuron models based on the current knowledge about the function of the mammalian MSO. Some previous studies implemented similar SNN but either used a simplified auditory periphery and thus limited the application of the model to pure tones (Brughera et al., 2013), based their model on topologies that disagree with newer physiological studies (Glackin et al., 2010) or focused on ITDs in the stimulus envelope (Wang et al., 2013). Using two different extraction methods, we found that applying the opponent-coding mechanism to the output of the model enabled a robust extraction of ITDs even in complex signals.

3.1. Sensitivity of the Linear-Decoder

We have shown that a simple linear-decoder can detect ITDs from the outputs of the left and right MSO models with a sensitivity that reflects human performance in a discrimination task and depended on sound frequency. The sensitivity was mainly determined by the maximum firing rate of the MSO at a given frequency. Our MSO neurons showed a peak at approximately 300 Hz. To our knowledge, no such systematic variation of firing rate with frequency has been described, but Yin and Chan (1990) noted a similar characteristic in the response of high-frequency MSO neurons. They recorded the response of neurons that phase-locked to the envelopes of amplitude-modulated tones and also showed a peak in the response at a modulation rate of 300 Hz. In the model, the responses were influenced by the subthreshold resonance of MSO neurons, which is due to the dynamics of the low threshold potassium current (Mikiel-Hunter et al., 2016). This resonance would also explain the results by Yin and Chan (1990). An explanation why no similar result in the response of low-frequency MSO neurons has been described is that these measurements are limited to responses derived at the neurons' best-frequency so that any systematic variation between neurons with different best-frequencies could be masked by variations in the overall response rate between neurons.

It should be noted that the sensitivity of the linear-decoder cannot be directly compared to results from psychoacoustical experiments, as the model only accounts for the lowest stages of the neuronal ITD-detection circuit in gerbils. In other words, it was not the goal of this study to replicate any psychophysical data *per se*, but rather to investigate the performance of the model on its own. Nevertheless, the model could be easily tuned to replicate psychoacoustic threshold data by adjusting the size of the neuronal population to fit human or animal data.

3.2. Influence of Missing SBCs on the Output of the Model

In the presented model network, MSO neurons received direct excitatory input from ANFs, while in the physiological case, they receive excitatory inputs from SBCs. SBCs have been found to increase the precision of phase-locking in comparison to ANFs (Dehmel et al., 2010; Künzel et al., 2011). The improvement shown in this study is rather small when compared to the large improvement that has been shown for GBCs (Joris et al., 1994). In spite of this Improvement, the precision is not much higher than that of the ANF model used in this study, and thus no further improvement in phase locking seemed necessary. A second function of SBCs could arise from non-monotonic rate-level functions due to an inhibitory sideband (Künzel et al., 2011; Keine and Rübsamen, 2015). Including a model that would reproduce the non-monotonic rate-level functions may also change the output of the MSO model, specifically, the behavior shown in **Figure 5C**. This change in the MSOs rate-level function may also be compensated by the ANN, so that the additional feature would not change the message of this paper, leading to the decision to neglect the influence of SBCs. It was also suggested that the slow GABA-ergic inhibition on the level of the SBC may support sound localization of complex sounds by acting as a gain control mechanism (Keine et al., 2016, 2017), this would be interesting to investigate in the context of the presented model but is outside of the scope of this paper.

3.3. Performance of the ANN-predictor

The model output showed a strong dependence on both frequency and level of the input signals. Previous models that employed the opponent-coding principle constructed the output of their models to be self normalizing (Pulkki and Hirvonen, 2009; Takanen et al., 2014) or directly extracted the phase from the left and the right input signals using gammatone filters (Dietz et al., 2011). While both methods are valid in view of a phenomenological modeling approach, they can not be easily applied to a neuronal network as presented in this study. We instead showed that a multilayer perceptron could be trained to compensate for frequency and level dependencies and to predict ITD values from the firing rate outputs of the spiking neuron network. By using an ANN to compensate for variability of the MSO output, this study neither makes any assumption about the exact location in the ascending auditory pathway, at which this compensation takes place, nor speculates about the exact mechanism underlying this compensation. We rather show that a very basic ANN containing only twenty hidden nodes in one layer is able to perform the compensation. The ANN-predictor was also shown to provide accurate ITD predictions for complex signals and for time-variant ITDs, even though it was trained on pure tones only. This suggests that the necessary compensation is independent of context. Psychoacoustic studies have shown that sound localization performance depends on the duration (Tobias, 1959) and bandwidth (Trahiotis and Stern, 1989) of the stimulus indicating an integration of information across frequency and time. In this study, the ANN predicted ITDs independently for each frequency and time frame. While integration over the frequency bands was implemented by calculating the mean prediction across all frequencies, no integration over time apart from the calculation of 30 ms averages was performed. Hence, the prediction capability is expected to further improve if the output of the model would also be integrated over time.

While the goal of this study was to evaluate the models' performance on the detection of ITDs, the prime interest of our binaural hearing lies in estimating the direction of a sound source instead of the ITD value. Since low-frequency ITDs between the ear canal signals provide a salient cue about sound source direction, reliable prediction of the ITDs indicates that the azimuthal sound direction may also be accurately predicted. To that end, the ANN could also be trained to directly predict azimuthal angles instead of ITDs.

4. METHODS

4.1. Topology of the Model

Both MSO and GBC neurons were modeled using single-compartment, Hodgkin-Huxley-type models simulated in python using the package Brian (Goodman, 2009). MSO as well as GBCs received direct excitatory input from ANF fibers, which were modeled using the model of Zilany et al. (2014), implemented in the python library *cochlea* (Rudnicki et al., 2015). Each population of neurons (ANF, GBC, MSO) always consisted of 500 independent neurons in each hemisphere. The frequency channel of the neuron population was set by selecting the appropriate critical frequency of the peripheral hearing model.

4.2. Spiking Models

While this study does not discuss the effect of single ionic currents, it makes use of Hodgkin-Huxley-type models, as simpler neuron models like the leaky integrate-and-fire neurons neglect the influence of ion channel dynamics. Especially the shift of best-ITD toward contralateral-leading ITDs has been shown to be influenced by both low-threshold potassium (Myoga et al., 2014) and hyperpolarizing ionic currents (Baumann et al., 2013), both of which are included in this model.

MSO neurons were simulated using single-compartment, Hodgkin-Huxley-type models. The dynamic of their membrane potential V_m is given by the following equation:

$$\begin{aligned}\frac{dV_m}{dt} = -\frac{1}{C_m}(I_{leak} + I_{Na} + I_K \\ + I_h + I_{syn,e} + I_{syn,i}),\end{aligned} \tag{1}$$

where C_m is the membrane capacitance, I_{leak} is the leakage current, I_{Na}, I_K, I_h are the sodium, potassium and hyperpolarizing ionic currents and $I_{syn,e}$, $I_{syn,i}$ are the excitatory and inhibitory synaptic currents respectively. All ionic currents were defined as follows:

$$I_x = \hat{g}_x a^m b^n (V_m - E_x), \tag{2}$$

where $\hat{g}_x$ and E_x are the maximal conductivity and Nernst potential for the respective ion species x. The gating variables a^m and b^n determine the channel kinetics. Equations for these variables can be found in the original publication: The sodium dynamics were implemented according to Rothman and Manis (2003) and were corrected for a body temperature of 37°C ($k = 3^{(T-22)/10}$). To gain realistic spike shapes as well as a spiking threshold, the activation kinetics had to be sped up by a factor of four. Potassium currents were modeled with the equations for the low threshold channels given by Khurana et al. (2011) with the steady-state inactivation z_∞ set to 0.4. The hyperpolarizing currents were modeled using the equations for dorsal MSO neurons from Baumann et al. (2013). We used a membrane capacity of 70 pF (Couchman et al., 2010) and the ionic conductivities were adjusted to fit the steady state and peak membrane resistances to values measured by Scott et al. (2005). Use of these values resulted in spiking thresholds close to the data published by Couchman et al. (2010). All parameters are summarized in **Table 1**. GBCs and their synaptic inputs were modeled using the neuron model with 40 non-depressing ANF inputs as proposed by Rudniki and Hemmert (2017).

TABLE 1 | Parameters for the MSO model.

Symbol	Value	Symbol	Value
C_m	70 pF	E_i	−70 mV
E_{rest}	−55.8 mV	$\hat{g}_{Na}$	3.9 μS
E_{Na}	56.2 mV	$\hat{g}_K$	650 nS
E_K	−90 mV	$\hat{g}_h$	520 nS
E_h	−35 mV	$\hat{g}_{leak}$	13 nS
E_e	0 mV		

4.3. Synaptic MSO Inputs

Each MSO neuron received six excitatory inputs from ANFs of each hemisphere. The excitatory post-synaptic currents (EPSCs) were modeled as an alpha function:

$$I_{syn,e} = \frac{t \cdot e^{1-t/\tau}}{\tau_e}(V_m - E_e). \tag{3}$$

Inhibition was provided via three GBC inputs per hemisphere. The inhibitory post-synaptic currents (IPSCs) were modeled using a bi-exponential function:

$$g_i = \hat{g}_i \frac{\tau_2 \cdot (e^{-t/\tau_{i,1}} - e^{-t/\tau_{i,2}})}{\tau_{i,2} - \tau_{i,1}} \cdot (V_m - E_i). \tag{4}$$

Both, excitatory and inhibitory timeconstants were fitted to recordings by Couchman et al. (2010) yielding values of $\tau_e = 0.17$ ms and $\tau_{i,1} = 0.14$ ms, $\tau_{i,2} = 1.6$ ms.

4.4. Sound Signals and Data Analysis

All sound signals were generated in Python at a sampling rate of 100 kHz as this sampling rate is required by the peripheral hearing model (Zilany et al., 2014). In the case of the speech signal, the sound was up-sampled from 44.2 to 100 kHz. Each sound signal was gated using a 20 ms long raised-cosine function and 20 ms of silence was attached to the beginning and the end of the signal. The stimuli were presented at a sound pressure level of 50 dB_{SPL} if not stated otherwise. ITDs were defined as the difference in the arrival times between the left and the right ears, with positive values corresponding to right leading sounds. To archive sub-sample ITDs, we generated the corresponding delays between the two signals by applying a fast Fourier-transform (FFT), adding the equivalent phase angles, which resulted from the delays, and reverse FFT back to time domain signal.

ITD-rate functions were fitted using a modified Gaussian function as shown in (5) were τ is the ITD value, R_{max} the maximum firing rate, W defines the width of the curve and B the location of the maximum (best-ITD).

$$R(\tau) = R_{max} \cdot e^{\frac{-(\tau-B)^2}{W^2}} + R_{offset} \tag{5}$$

Spiking data were analyzed using the Thorns toolbox for python. Firing rates were always given as the average response of the whole population. To compensate for the intracochlear delay of the inner ear model, we only considered action potentials arriving 25 ms after signal onset and up to 25 ms after the end of the signal.

4.5. Calculation of Just Noticeable Differences

JNDs for our model were calculated by presenting two stimuli with ITDs located symmetrically around zero—i.e., $-\tau/2$ and $\tau/2$. The difference between the two ITD was denoted ΔITD . We calculated independently, the difference in firing rate at both hemispheres (ΔR) for each of the presented signals. The two values were then compared to each other. If the ΔR value for the negative ITD signal was larger than the one for the positive ITD signal, the trial was considered as a correct prediction. Each ΔITD was presented 100 times and the fraction of correct trials was calculated. To calculate the JND, we presented 20 logarithmic arranged ΔITD in the range from 2 to 800 μs. The resulting fraction correct values were then fitted with a weibull function. The JND was defined as the ITD at which 75% correct predictions were achieved.

4.6. The Artificial Neural Network Predictor

The ANN network was implemented using the Theano package for Python. The ANN layout was that of a classic multilayer perceptron containing an input layer with seven nodes, one hidden layers with twenty nodes and an output layer with two nodes (see **Figure 6**). Both the hidden and the output layer consisted of non-linear nodes with a $\tanh(x)$ activation function.

The predictor was designed to make predictions for every 30 ms section of the signal. For this, average firing rates for both MSO hemispheres and for the ANF of one hemisphere were calculated in bins of 30 ms. The model firing rates of MSO and ANF as well as the best frequency of these neurons were given as the ANN inputs. To provide some history which can be used to compensate for on- and off-sets, the predictor was also provided with the firing rate in the previous 30 ms bin. Using this information, the ANN gave a prediction of the ITD value in the current bin and a classification whether the presented bin actually contained a signal (signal exists).

The network was trained on the MSO model output from 2,000 different 300 ms long sine tones which were padded by 60 ms of quiet. For each tone the level, frequency as well as ITD were randomly chosen to lie between 30 and 70 dB_{SPL}, 125 and 1,000 Hz, and ±500 µs, respectively. The target data for the training set consisted of the ITD value of the corresponding input signal, as well as the classification whether the current time frame contained a signal or not. The target for the classification was set to -1 for the two time bins at the start and at the end of each signal as those contained silence. It was set to 1 for all other bins. The set of training signals was then split into three subsets, a training set containing 80% of the data, a validation and test set both containing 10% of the signals. The ANN was trained on the training set, until the improvement on the mean squared error function for the validation set stayed consistently below 0.01%.

5. DATA SHARING

The neuronal model presented in this paper will be made available on request as well as through the GitHub Repository https://github.com/timtammittee/mso_model_frontiers2017.

AUTHOR CONTRIBUTIONS

JE: designed the study and wrote the paper; WH: supervised the study and helped with design and writing.

ACKNOWLEDGMENTS

This work was supported by the German Research Foundation (DFG) and the Technical University of Munich (TUM) in the framework of the Open Access Publishing Program and a DFG grant within the PP1608 Ultrafast and temporally precise information processing: normal and dysfunctional hearing (HE6713/1-1 and 1-2).

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fnins.2018.00140/full#supplementary-material

REFERENCES

Almeida, J. S. (2002). Predictive non-linear modeling of complex data by artificial neural networks. *Curr. Opin. Biotechnol.* 13, 72–76. doi: 10.1016/s0958-1669(02)00288-4

Baumann, V. J., Lehnert, S., Leibold, C., and Koch, U. (2013). Tonotopic organization of the hyperpolarization-activated current (Ih) in the mammalian medial superior olive. *Front. Neural Circuits* 7:117. doi: 10.3389/fncir.2013.00117

Brand, A., Behrend, O., Marquardt, T., McAlpine, D., and Grothe, B. (2002). Precise inhibition is essential for microsecond interaural time difference coding. *Nature* 417, 543–547. doi: 10.1038/417543a

Bremen, P., and Joris, P. X. (2013). Axonal recordings from medial superior olive neurons obtained from the lateral lemniscus of the chinchilla (chinchilla laniger). *J. Neurosci.* 33, 17506–17518. dOi:10.1523/JNEUROSCI.1518-13.2013

Briley, P. M., Kitterick, P. T., and Summerfield, A. Q. (2012). Evidence for opponent process analysis of sound source location in humans. *J. Assoc. Res. Otolaryngol.* 14, 83–101. doi: 10.1007/s10162-012-0356-x

Brughera, A., Dunai, L., and Hartmann, W. M. (2013). Human interaural time difference thresholds for sine tones: the high-frequency limit. *J. Acoust. Soc. Am.* 133, 2839–2855. doi: 10.1121/1.4795778

Brughera, A. R., Stutman, E. R., Carney, L. H., and Colburn, H. S. (1996). A model with excitation and inhibition for cells in the medial superior olive. *Audit. Neurosci.* 2, 219–233.

Carr, C. E., and Konishi, M. (1988). Axonal delay lines for time measurement in the owl's brainstem. *Proc. Natl. Acad. Sci. U.S.A.* 85, 8311–8315.

Colburn, H. S., and Durlach, N. I. (1978). "Models of binaural interaction," in *Handbook of Perception, Vol. 4*, eds E. Carterette and M. Friedman (New York, NY: Academic Press), 467–518.

Couchman, K., Grothe, B., and Felmy, F. (2010). Medial superior olivary neurons receive surprisingly few excitatory and inhibitory inputs with balanced strength and short-term dynamics. *J. Neurosci.* 30, 17111–17121. doi: 10.1523/JNEUROSCI.1760-10.2010

Day, M. L., and Delgutte, B. (2013). Decoding sound source location and separation using neural population activity patterns. *J. Neurosci.* 33, 15837–15847. doi: 10.1523/JNEUROSCI.2034-13.2013

Dehmel, S., Kopp-Scheinpflug, C., Weick, M., Dörrscheidt, G. J., and Rübsamen, R. (2010). Transmission of phase-coupling accuracy from the auditory nerve to spherical bushy cells in the mongolian gerbil. *Hear. Res.* 268, 234–249. doi: 10.1016/j.heares.2010.06.005

Dietz, M., Ewert, S. D., and Hohmann, V. (2011). Auditory model based direction estimation of concurrent speakers from binaural signals. *Speech Commun.* 53, 592–605. doi: 10.1016/j.specom.2010.05.006

Franken, T. P., Roberts, M. T., Wei, L., Golding, N. L., and Joris, P. X. (2015). *In vivo* coincidence detection in mammalian sound localization generates phase delays. *Nat. Neurosci.* 18, 444–452. doi: 10.1038/nn.3948

Glackin, B., Wall, J., McGinnity, T., Maguire, L., and McDaid, L. (2010). A spiking neural network model of the medial superior olive using spike timing dependent plasticity for sound localization. *Front. Comput. Neurosci.* 4:18. doi: 10.3389/fncom.2010.00018

Goodman, D. F., Benichoux, V., and Brette, R. (2013). Decoding neural responses to temporal cues for sound localization. *eLife* 2:e01312. doi: 10.7554/eLife.01312

Goodman, D. F. M. (2009). The brian simulator. *Front. Neurosci.* 3, 192–197. doi: 10.3389/neuro.01.026.2009

Grothe, B., Pecka, M., and McAlpine, D. (2010). Mechanisms of sound localization in mammals. *Physiol. Rev.* 90, 983–1012. doi: 10.1152/physrev.00026.2009

Guinan, J. J., Norris, B. E., and Guinan, S. S. (1972). Single auditory units in the superior olivary complex: II: locations of unit categories and tonotopic organization. *Int. J. Neurosci.* 4, 147–166.

Harper, N. S., Scott, B. H., Semple, M. N., and McAlpine, D. (2014). The neural code for auditory space depends on sound frequency and head size in an optimal manner. *PLoS ONE* 9:e108154. doi: 10.1371/journal.pone.0108154

Jeffress, L. A. (1948). A place theory of sound localization. *J. Comp. Physiol. Psychol.* 41, 35–39.

Jenkins, W. M., and Masterton, R. B. (1982). Sound localization: effects of unilateral lesions in central auditory system. *J. Neurophysiol.* 47, 987–1016.

Jercog, P. E., Svirskis, G., Kotak, V. C., Sanes, D. H., and Rinzel, J. (2010). Asymmetric excitatory synaptic dynamics underlie interaural time difference processing in the auditory system. *PLoS Biol.* 8:e1000406. doi: 10.1371/journal.pbio.1000406

Joris, P. X., Carney, L. H., Smith, P. H., and Yin, T. C. (1994). Enhancement of neural synchronization in the anteroventral cochlear nucleus. I. Responses to tones at the characteristic frequency. *J. Neurophysiol.* 71, 1022–1036.

Joris, P. X., de Sande, B. V., Louage, D. H., and van der Heijden, M. (2006). Binaural and cochlear disparities. *Proc. Natl. Acad. Sci. U.S.A.* 103, 12917–12922. doi: 10.1073/pnas.0601396103

Keine, C., and Rübsamen, R. (2015). Inhibition shapes acoustic responsiveness in spherical bushy cells. *J. Neurosci.* 35, 8579–8592. doi: 10.1523/JNEUROSCI.0133-15.2015

Keine, C., Rübsamen, R., and Englitz, B. (2016). Inhibition in the auditory brainstem enhances signal representation and regulates gain in complex acoustic environments. *eLife* 5:e19295. doi: 10.7554/eLife.19295

Keine, C., Rübsamen, R., and Englitz, B. (2017). Signal integration at spherical bushy cells enhances representation of temporal structure but limits its range. *eLife* 6:e29639. doi: 10.7554/eLife.29639

Khurana, S., Remme, M. W. H., Rinzel, J., and Golding, N. L. (2011). Dynamic interaction of Ih and IK-LVA during trains of synaptic potentials in principal neurons of the medial superior olive. *J. Neurosci.* 31, 8936–8947. doi: 10.1523/JNEUROSCI.1079-11.2011

Klumpp, R. G. (1956). Some measurements of interaural time difference thresholds. *J. Acoust. Soc. Am.* 28:859.

Künzel, T., Borst, J. G. G., and van der Heijden, M. (2011). Factors controlling the input-output relationship of spherical bushy cells in the gerbil cochlear nucleus. *J. Neurosci.* 31, 4260–4273. doi: 10.1523/JNEUROSCI.5433-10.2011

Lehnert, S., Ford, M. C., Alexandrova, O., Hellmundt, F., Felmy, F., Grothe, B., et al. (2014). Action potential generation in an anatomically constrained model of medial superior olive axons. *J. Neurosci.* 34, 5370–5384. doi: 10.1523/JNEUROSCI.4038-13.2014

Lindemann, W. (1986). Extension of a binaural crosscorrelation model by contralateral inhibition. I. Simulation of lateralization for stationary signals. *J. Acoust. Soc. Am.* 80, 1608–1622.

Magezi, D. A., and Krumbholz, K. (2010). Evidence for opponent-channel coding of interaural time differences in human auditory cortex. *J. Neurophysiol.* 104, 1997–2007. doi: 10.1152/jn.00424.2009

Malhotra, S., Hall, A. J., and Lomber, S. G. (2004). Cortical control of sound localization in the cat: Unilateral cooling deactivation of 19 cerebral areas. *J. Neurophysiol.* 92, 1625–1643. doi: 10.1152/jn.01205.2003

McAlpine, D., and Grothe, B. (2003). Sound localization and delay lines–do mammals fit the model? *Trends Neurosci.* 26, 347–350. doi: 10.1016/S0166-2236(03)00140-1

McAlpine, D., Jiang, D., and Palmer, A. R. (2001). A neural code for low-frequency sound localization in mammals. *Nat. Neurosci.* 4, 396–401. doi: 10.1038/86049

Mikiel-Hunter, J., Kotak, V., and Rinzel, J. (2016). High-frequency resonance in the gerbil medial superior olive. *PLoS Comput. Biol.* 12:e1005166. doi: 10.1371/journal.pcbi.1005166

Myoga, M. H., Lehnert, S., Leibold, C., Felmy, F., and Grothe, B. (2014). Glycinergic inhibition tunes coincidence detection in the auditory brainstem. *Nat. Commun.* 5:3790. doi: 10.1038/ncomms4790

Nuetzel, J. M., and Hafter, E. R. (1976). Lateralization of complex waveforms: effects of fine structure, amplitude, and duration. *J. Acoust. Soc. Am.* 60, 1339–1346.

Pecka, M., Brand, A., Behrend, O., and Grothe, B. (2008). Interaural time difference processing in the mammalian medial superior olive: the role of glycinergic inhibition. *J. Neurosci.* 28, 6914–6925. doi: 10.1523/JNEUROSCI.1660-08.2008

Phillips, D. P., Carmichael, M. E., and Hall, S. E. (2006). Interaction in the perceptual processing of interaural time and level differences. *Hear. Res.* 211, 96–102. doi: 10.1016/j.heares.2005.10.005

Pulkki, V., and Hirvonen, T. (2009). Functional count-comparison model for binaural decoding. *Acta Acust. United Acust.* 95, 883–900. doi: 10.3813/AAA.918220

Remme, M. W. H., Donato, R., Mikiel-Hunter, J., Ballestero, J. A., Foster, S., Rinzel, J., et al. (2014). Subthreshold resonance properties contribute to the efficient coding of auditory spatial cues. *Proc. Natl. Acad. Sci. U.S.A.* 111, E2339–E2348. doi: 10.1073/pnas.1316216111

Roberts, M. T., Seeman, S. C., and Golding, N. L. (2013). A mechanistic understanding of the role of feedforward inhibition in the mammalian sound localization circuitry. *Neuron* 78, 923–935. doi: 10.1016/j.neuron.2013.04.022

Rothman, J. S., and Manis, P. B. (2003). The roles potassium currents play in regulating the electrical activity of ventral cochlear nucleus neurons. *J. Neurophysiol.* 89, 3097–3113. doi: 10.1152/jn.00127.2002

Rudnicki, M., Schoppe, O., Isik, M., Völk, F., and Hemmert, W. (2015). Modeling auditory coding: from sound to spikes. *Cell Tissue Res.* 361, 159–175. doi: 10.1007/s00441-015-2202-z

Rudniki, M., and Hemmert, W. (2017). High entrainment constrains synaptic depression levels of an *in vivo* globular bushy cell model. *Front. Comput. Neurosci.* 11:16. doi: 10.3389/fncom.2017.00016

Salminen, N. H., Tiitinen, H., Yrttiaho, S., and May, P. J. C. (2010). The neural code for interaural time difference in human auditory cortex. *J. Acoust. Soc. Am.* 127, EL60–EL65. doi: 10.1121/1.3290744

Scott, L. L., Mathews, P. J., and Golding, N. L. (2005). Posthearing developmental refinement of temporal processing in principal neurons of the medial superior olive. *J. Neurosci.* 25, 7887–7895. doi: 10.1523/JNEUROSCI.1016-05.2005

Smith, R. L. (1977). Short-term adaptation in single auditory nerve fibers: some poststimulatory effects. *J. Neurophysiol.* 40, 1098–1111.

Stecker, G. C., Harrington, I. A., and Middlebrooks, J. C. (2005). Location coding by opponent neural populations in the auditory cortex. *PLoS Biol.* 3:e78. doi: 10.1371/journal.pbio.0030078

Takanen, M., Santala, O., and Pulkki, V. (2014). Visualization of functional count-comparison-based binaural auditory model output. *Hear. Res.* 309, 147–163. doi: 10.1016/j.heares.2013.10.004

Tobias, J. V. (1959). Lateralization threshold as a function of stimulus duration. *J. Acoust. Soc. Am.* 31, 1591. doi: 10.1121/1.1907664

Trahiotis, C., and Stern, R. M. (1989). Lateralization of bands of noise: Effects of bandwidth and differences of interaural time and phase. *J. Acoust. Soc. Am.* 86, 1285–1293.

Wagener, K., Brand, T., and Kollmeier, B. (1999). Entwicklung und evaluation eines satztests fr die deutsche sprache III: Evaluation des oldenburger satztests. *Zeitschrift Audiologie/Audiological Acoustics* 38:8695.

Wang, L., Devore, S., Delgutte, B., and Colburn, H. S. (2013). Dual sensitivity of inferior colliculus neurons to ITD in the envelopes of high-frequency sounds: experimental and modeling study. *J. Neurophysiol.* 111, 164–181. doi: 10.1152/jn.00450.2013

Yin, T. C., and Chan, J. C. (1990). Interaural time sensitivity in medial superior olive of cat. *J. Neurophysiol.* 64, 465–488.

Zhou, Y., Carney, L. H., and Colburn, H. S. (2005). A model for interaural time difference sensitivity in the medial superior olive: interaction of excitatory and inhibitory synaptic inputs, channel dynamics, and cellular morphology. *J. Neurosci.* 25, 3046–3058. doi: 10.1523/JNEUROSCI.3064-04.2005

Zilany, M. S. A., Bruce, I. C., and Carney, L. H. (2014). Updated parameters and expanded simulation options for a model of the auditory periphery. *J. Acoust. Soc. Am.* 135, 283–286. doi: 10.1121/1.4837815

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Neuroscience

ORIGINAL RESEARCH
published: 09 February 2018
doi: 10.3389/fnins.2018.00023

Feature Representations for Neuromorphic Audio Spike Streams

*Jithendar Anumula**, *Daniel Neil*†, *Tobi Delbruck and Shih-Chii Liu*

Institute of Neuroinformatics, University of Zurich and ETH Zurich, Zurich, Switzerland

Event-driven neuromorphic spiking sensors such as the silicon retina and the silicon cochlea encode the external sensory stimuli as asynchronous streams of spikes across different channels or pixels. Combining state-of-art deep neural networks with the asynchronous outputs of these sensors has produced encouraging results on some datasets but remains challenging. While the lack of effective spiking networks to process the spike streams is one reason, the other reason is that the pre-processing methods required to convert the spike streams to frame-based features needed for the deep networks still require further investigation. This work investigates the effectiveness of synchronous and asynchronous frame-based features generated using spike count and constant event binning in combination with the use of a recurrent neural network for solving a classification task using N-TIDIGITS18 dataset. This spike-based dataset consists of recordings from the Dynamic Audio Sensor, a spiking silicon cochlea sensor, in response to the TIDIGITS audio dataset. We also propose a new pre-processing method which applies an exponential kernel on the output cochlea spikes so that the interspike timing information is better preserved. The results from the N-TIDIGITS18 dataset show that the exponential features perform better than the spike count features, with over 91% accuracy on the digit classification task. This accuracy corresponds to an improvement of at least 2.5% over the use of spike count features, establishing a new state of the art for this dataset.

Keywords: dynamic audio sensor, spike feature generation, exponential kernels, recurrent neural network, audio word classification

OPEN ACCESS

Edited by:
Jeffrey L. Krichmar,
University of California, Irvine,
United States

Reviewed by:
Thomas Nowotny,
University of Sussex, United Kingdom
Garrick Orchard,
National University of Singapore,
Singapore

****Correspondence:***
Jithendar Anumula
anumula@ini.uzh.ch

†Present Address:
Daniel Neil,
BenevolentAI, New York, NY,
United States

Specialty section:
This article was submitted to
Neuromorphic Engineering,
a section of the journal
Frontiers in Neuroscience

Received: *07 September 2017*
Accepted: *11 January 2018*
Published: *09 February 2018*

Citation:
Anumula J, Neil D, Delbruck T and
Liu S-C (2018) Feature
Representations for Neuromorphic
Audio Spike Streams.
Front. Neurosci. 12:23.
doi: 10.3389/fnins.2018.00023

1. INTRODUCTION

The event processing methods for the asynchronous spikes of event-based sensors such as the Dynamic Vision Sensor (DVS) (Lichtsteiner et al., 2008; Berner et al., 2013; Posch et al., 2014; Yang et al., 2015) and the Dynamic Audio Sensor (DAS) (Liu et al., 2014; Yang et al., 2016) fall roughly into two categories: either by the use of neural network methods or machine learning algorithms. These methods have been primarily developed for event-based vision sensors and with the availability of DVS datasets (Orchard et al., 2015; Serrano-Gotarredona and Linares-Barranco, 2015; Barranco et al., 2016), performances of these methods can be compared.

In recent years, the field of deep learning has seen major developments leading to networks that achieve state-of-art performance on complex tasks such as speech recognition and visual object recognition (Schmidhuber, 2014; LeCun et al., 2015). With event-based sensors finding increasing relevance in event-driven artificial sensory or cognitive systems, there has been a new effort in interfacing these sensors with these powerful machine learning networks. However, deep learning frameworks typically use frame-based data. To interface the output of the event-based sensors to the deep network, there are two alternative methods. The first method is to present the spikes to spiking deep networks as has been reported (Farabet et al., 2012; Pérez-Carrasco et al., 2013; Zhao et al., 2015; Esser et al., 2016; Amir et al., 2017). By using conversion methods that convert pre-trained standard deep networks into equivalent-accurate spiking networks (Diehl et al., 2015; Rueckauer et al., 2017) or by using the training methods from deep learning on networks that capture the underlying parameters of the spiking neuron (O'Connor et al., 2013; Stromatias et al., 2015), we are starting to see spiking deep networks that can be competitive with the standard deep networks.

Another method is to create either synchronous or asynchronous feature frames from the spikes before presentation to the time-stepped deep networks. This method has seen success in the field of neuromorphic vision primarily, as pre-processing methods produce frames from event-driven sensor data to use as inputs to deep networks for classification tasks (Moeys et al., 2016; Neil and Liu, 2016; Lungu et al., 2017). Although these pre-processing methods are outperformed on standard classification tasks by the methods using the traditional frame based sensors, they can help reduce computation by using the data driven nature of the sensors and processing the networks only when the sensor produces events.

This work aims to methodically examine existing and novel spike pre-processing methods for processing the output of the DAS for use with deep networks and machine learning algorithms, in particular for real-time applications. We consider two existing feature extraction methods that generate feature frames using spike counts within a fixed time bin and constant spike count (event) bins respectively. We also propose a new pre-processing method that generates feature frames by applying an exponential kernel to each event. We compare the performances of the different pre-processing methods by combining them with deep learning recurrent neural networks which include gated units (Chung et al., 2014; Neil et al., 2016) and testing the networks on two audio classification tasks (isolated recordings and connected streams) using a recorded audio spike dataset called N-TIDIGITS18. This dataset consists of spike recordings from a Dynamic Audio Sensor in response to the TIDIGITS (Leonard and Doddington, 1993) audio dataset.

2. METHODS

This section presents a description of the hardware cochlea sensors, details the feature generation methods, including the proposed exponential feature generation method and briefly describes the deep network architectures used in this study.

2.1. Dynamic Audio Sensor

The Dynamic Audio Sensor is a binaural silicon cochlea system, with each ear connected to a set of 64 bandpass filters whose center frequencies are logarithmically distributed from approximately 50 Hz to 20 kHz. The events are then asynchronously generated from each of the filters. A silicon cochlea sensor using half wave rectification for the generation of events is the CochleaAMS1b (Chan et al., 2007) and the CochleaAMS1c (Liu et al., 2014), while a cochlea sensor using asynchronous delta modulation for the generation of events is the CochLP (Yang et al., 2016). The CochleaAMS1c sensor is an improved design of the CochleaAMS1b. Each channel of the CochleaAMS1b and CochleaAMS1c has four neurons and each neuron implements a different threshold level for spike generation. In many of the experiments, only the events from a single neuron of one ear are used. An example output for the CochleaAMS1c is shown in **Figure 1**. The methods evaluated in this work were carried out on recordings from the CochleaAMS1b and CochleaAMS1c, while they will be evaluated on CochLP in the future.

2.2. Feature Extraction Methods

The event data from the cochlea sensors can be converted to frame-based features through multiple methods. One commonly used feature type is the Spike Count (SC) feature (Zai et al., 2015; Anumula et al., 2017), that is generated by the creation of a histogram across the frequency channels of the events within a time window. In the case of the DAS, the feature vector for each time frame is, at maximum, a 64-length vector where each element consists of the number of events in that frequency channel. The two main variants of SC features are time-binned and event-binned features. Their formulation is described below.

2.2.1. Raw Spikes

An audio event stream can be mathematically represented as

$$e_i = \left[t_i, f_i\right], i \in \mathbb{N} \qquad (1)$$

where e_i is the ith event from the frequency channel f_i in the event stream at time t_i. The f_i can range between 1 and N_c where N_c is the number of frequency channels in the sensor. Also note that the events are time ordered, i.e., for $i < j$, $t_i \leq t_j$. These raw spike information can be processed directly as a sequence by the

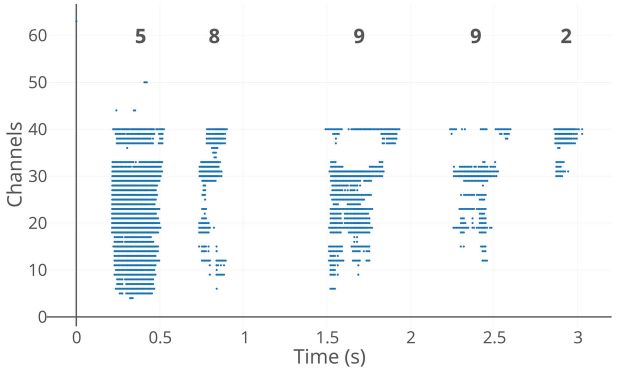


FIGURE 1 | CochleaAMS1c spike output example. The y-axis indicates the 64 frequency channels of the sensor with lower frequency channels at the top. The spikes are in response to the spoken digit sequence "5-8-9-9-2" from the speaker "IM" in the TIDIGITS dataset. The five digits in the sequence can be clearly seen to be apart with significant gaps between them in the encoded sample above. This example also demonstrates the data driven nature of the sensor where it outputs events only when there's a stimulus in the environment.

recurrent networks. Such a method is not usually feasible though because of the inability of the standard recurrent networks to process longer sequences, but they can be efficiently processed through the Phased LSTM, a recently introduced gated recurrent network architecture (Neil et al., 2016).

2.2.2. Time-Binned Spike Count Features

For the generation of time binned Spike Count features, the frame duration for generating the feature is of fixed time length. Time-binned SC features have been used for the speaker identification task using spike recordings generated from the TIMIT dataset (Liu et al., 2010; Li et al., 2012), the YOHO dataset (Chakrabartty and Liu, 2010), and real-world DAS recordings (Anumula et al., 2017).

The time-binned SC features F^{tb} for a time window length of T_l are defined as follows:

$$F_j^{tb}\left(f\right) = \mathbf{card}(\{e_i \mid T_l \cdot (j-1) \leq t_i < T_l \cdot j, f_i = f\}) \qquad (2)$$

where F_j^{tb} is the jth frame of the features, **card**() is the cardinality of a set, $\cdot$ is the standard multiplication operator, and f is the position of the frequency channel.

Figure 2 shows how the time-binned SC features are generated from the spikes.

2.2.3. Event-Binned Spike Count Features

Event-binned SC features consist of frames in which there are a fixed number of events. Unlike time-binned spike count features, event binning is a data driven approach and eliminates the need for input normalization. These features have been used for both the DVS and the DAS. In the robot predator-prey scenario in Moeys et al. (2016), the DVS retina data is integrated into 36 × 36 frames as 2D histograms obtained by integrating 5,000 events in 200 possible gray level values. Since the DVS frames are sparse, active DVS frame pixels accumulate about 50 events. Constant-event frames from the spiking TIMIT dataset have also been used together with a Support Vector Machine Classifier in a speaker identification task (Li et al., 2012).

The event-binned spike count features F^{eb} are defined as follows. The jth frame is given by

$$F_j^{eb}\left(f\right) = \mathbf{card}(\{e_i \mid E \cdot (j-1) \leq i < E \cdot j, f_i = f\}) \qquad (3)$$

where **card**() is the cardinality of a set, $\cdot$ is the standard multiplication operation, f is the position of the frequency channel and E is the number of events binned into a single frame.

Figure 3 shows how the event-binned spike count features are generated from the spikes.

2.2.4. Comparison of Time Binning and Event Binning

Although both methods capture the distribution of the events across the frequency channels, there is a difference between the features generated from these methods. The main difference is that the time window used for time binning is of constant length, while the time window of the event-binned features are of varying lengths. The lengths depend on the input event rate over time. This can be seen in the examples of time-binned and event-binned SC features for a single word as shown in **Figure 4** and for a sentence as shown in **Figure 5**. In **Figure 5**, it can be seen that the information about silences in the sentence is temporally smeared in the event-binned features. This property is not desirable as it could be a disadvantage when trying to extract information that depend on the silence periods within the

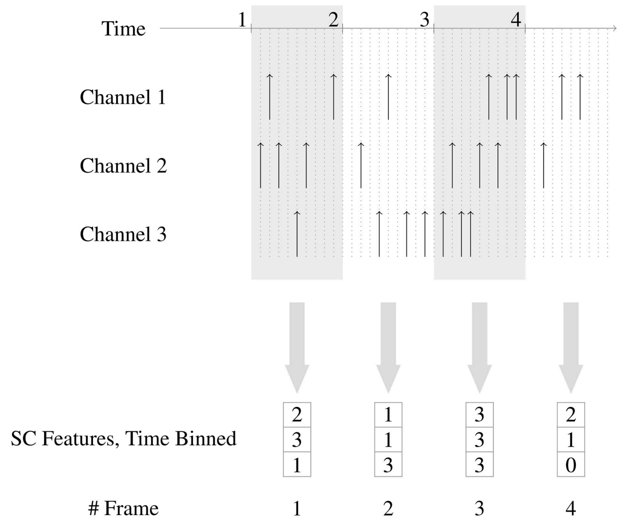


FIGURE 2 | Generation of time-binned Spike Count features. Three channels are shown in this example. The fixed length time windows used for binning the events are non overlapping and of unit time length. In frame 2 , there is 1 event in channel 1, 1 event in channel 2 and 3 events in channel 3, and hence the corresponding feature is (1, 1, 3).

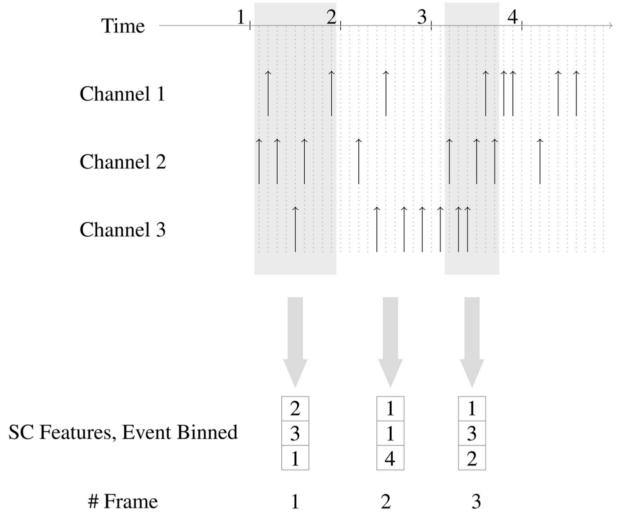


FIGURE 3 | Generation of event-binned Spike Count features. Three channels are shown in this example. Every time window frame used for binning the events has 6 events and there is no overlap of events between consecutive time window frames. In the second time frame, the 6 events are distributed as 1, 1, and 4 across channels 1, 2, and 3, respectively, hence the corresponding feature is (1, 1, 4).

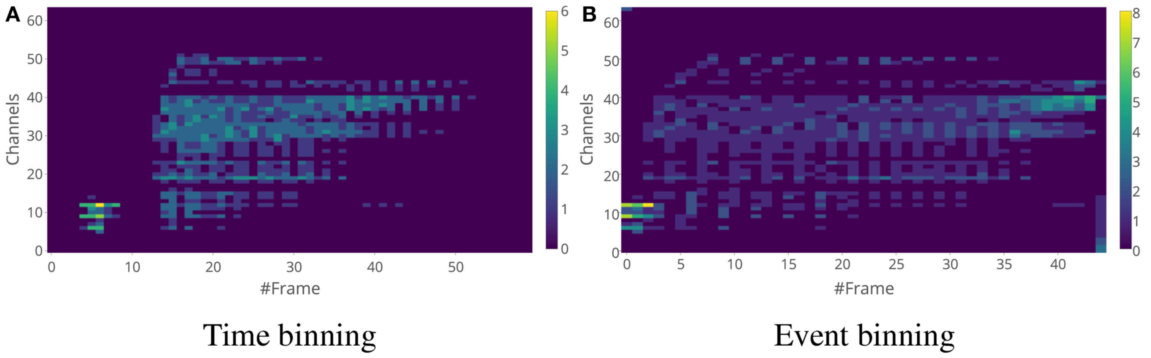


FIGURE 4 | Spike Count features for a digit sample “2”. The time window length for time binning in **(A)** is 5 ms and the number of events in a single frame for event binning in **(B)** is 25. There does not seem to be a clear advantage of choosing event binning over time binning when it comes to individual digits. Note that event binning for this example produces fewer frames compared to the time binning.

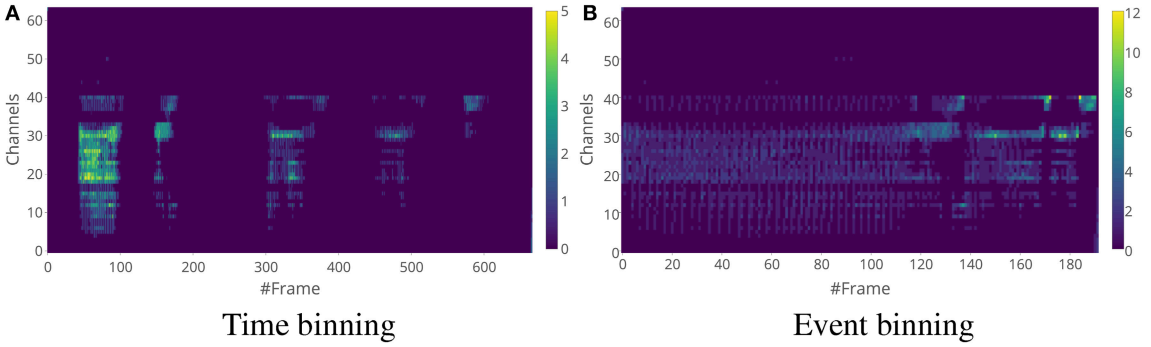


FIGURE 5 | Spike Count features for a digit sequence “5-8-9-9-2”. The time window length for time binning in **(A)** is 5 ms and the number of events in a single frame for event binning in **(B)** is 25. The event binning method does not completely encode the timing information in the sample. Also, the silence periods between the digits is absent in the event-binned features.

sentences, unless silence segmentation is done before generating the features.

2.2.5. Data-Driven Time-Binned Spike Count Features

Further, a data-driven time-binning method is introduced and employed in this work. In contrast to the previous time-binned SC features described in section 2.2.2, a feature frame is not processed if no spikes occurred within the corresponding time bin. In addition, this method specifically uses a brief time-bin length. This allows fewer inputs compared to time-binned spike counts (as a fixed-size vector is either presented or skipped), and far fewer inputs to be presented to the network compared to sequentially presenting raw events while maintaining much of the time resolution. Here, using a short time-bin length allows a high degree of spike time accuracy to be maintained, as individual spikes have correct timestamps discretized to the bin length. These data-driven time-binned SC features F^d can be defined as

$$\begin{aligned} F^d_j = F^{tb}_i, &\text{ where } i \text{ is such that } \mathbf{max}\left(F^{tb}_i\right) > 0 \text{ and} \\ &\mathbf{card}\left(\{k \mid k \le i, \mathbf{max}\left(F^{tb}_k\right) > 0\}\right) = j \end{aligned} \qquad (4)$$

2.2.6. Exponential Features

Finally, we introduce a real-valued feature representation that is more amenable to training deep neural networks. This feature is created by convolving each spike with an exponential kernel, that captures the timing information carried by the spikes and has been used in various models, for e.g., Abdollahi and Liu (2011) and Lagorce et al. (2015, 2016). Exponentials are frequently used in neuronal models such as the exponential integrate-and-fire model (Brette and Gerstner, 2005). Although other kernels such as the Gaussian kernels used in the analysis of neuronal firing patterns (Szűcs, 1998) can also be used, we restrict our study here to exponential kernels because they can be applied easier to create real-time features. The resulting output after the convolution is sometimes treated as a real-valued time surface as described in Lagorce et al. (2016). These exponential features have also been used in classification tasks such as image classification (Tapson et al., 2013; Cohen et al., 2016). We first describe the creation of the exponential features and then the binning methods used on these features.

For an audio event stream defined as in Equation (1), the exponential feature F^e_i for an event e_i is constructed by first defining a time context $\mathcal{T}_i$ for the event. The time context is an N_c dimensional vector where N_c is the number of frequency channels in the audio sensor and is defined as

$$\mathcal{T}_i\left(f\right) = \max_{j \leq i} \left\{t_j \mid f_j = f\right\} \tag{5}$$

where f is the position of a frequency channel. The exponential feature for an event is then defined as

$$F_i^e\left(f\right) = e^{-\left(t_i - \mathcal{T}_i(f)\right)/\tau} \tag{6}$$

An illustration describing the generation of the exponential features for the events is shown in **Figure 6**.

Once these exponential features are created, the events are binned into time window frames either through time binning or event binning like in the SC features, and the average of the exponential features for the events in the time window frame is used as the exponential feature for the frame. For the rest of the paper, we use the term "exponential features" to mean exponential features for a frame. Examples of time binning and event binning exponential features for a single word are shown in **Figure 7** and for a sentence are shown in **Figure 8**.

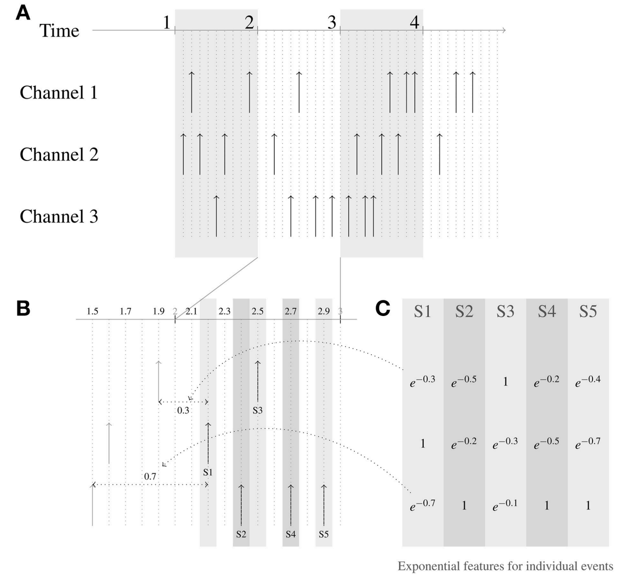


FIGURE 6 | Generation of exponential features for events. Three channels are shown in this example. The time constant parameter t used for generating the features is 1 time unit. The events streams are shown in **(A)**, the zoomed-in picture of the events in the second frame are shown in **(B)**, and the exponential features for this frame is shown in **(C)**. Consider the event at time $t = 2.2$, labeled S1. In channel 1, the closest event in time to the current event occurred 0.3 time units before, and thus the corresponding feature value for the channel 1 in the exponential feature vector for event S1 is $e^{-(0.3/1)}$. Similarly for channel 3, the closest event in time to the current event occurred 0.7 time units before, and thus the corresponding entry for channel 3 in the exponential feature for S1 is $e^{-(0.7/1)}$. For channel 2, since the current event is at channel 2, the exponential feature value at channel 2 is $e^{-(0/1)=1}$.

For a real-time implementation, the exponential features are computed recursively as follows.

$$F_i^e\left(f\right) = \begin{cases} e^{-(t_i - t_{i-1})/\tau} F_{i-1}^e\left(f\right), & \text{if} f \neq f_i \\ 1, & \text{if} f = f_i \end{cases} \tag{7}$$

With F_0^e initialized to a zero vector, it can easily be seen that the above implementation corresponds to the definition in Equation (6).

2.3. Recurrent Neural Networks

Convolutional Neural Networks are typically used in vision classification tasks and have been successfully used together with the Dynamic Vision Sensor (Moeys et al., 2016). These networks have a feedforward architecture where the neurons in one layer only drive the neurons in the upper layers. However, recurrent neural networks (RNNs) in which neurons in one layer recurrently receive input from neurons in the same layer, are more generally used when the inputs consist of temporal sequences.

Given a sequence $x = (x_1, x_2, \ldots, x_T)$, the RNN layer updates its hidden state h_t with $t \in \{0, 1, 2, \ldots, T\}$, with h_0 being the initial state and $h_t = \phi\left(h_{t-1}, x_t\right)$, where ϕ is a non-linear function. Generally, the update function for the hidden state is of the form $h_t = \varphi\left(Uh_{t-1} + Wx_t\right)$, where U and W are connection matrices of appropriate sizes and φ is an activation function such as a logistic sigmoid or the hyperbolic tangent (Chung et al., 2014).

Training RNNs using gradient descent to learn long term time dependencies in the input is difficult because of the vanishing/exploding gradient problem (Bengio et al., 1994). In order to counter this problem, the Long-Short Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) neuron model was proposed. This model has an activation function that is managed by different gates acting like a memory control for the neuron. The subsequently proposed Gated Recurrent Unit (GRU) (Cho et al., 2014) model performs well on similar tasks and has the advantage of using fewer parameters. In our experiments, we use both GRU and LSTM RNNs and the following sections introduce these models.

2.3.1. Long-Short Term Memory

The form of LSTM used in this work derives from Graves (2013):

$$i_t = \sigma_i(W_{xi}x_t + W_{hi}h_{t-1} + w_{ci} \odot c_{t-1} + b_i) \tag{8}$$

$$f_t = \sigma_f(W_{xf}x_t + W_{hf}h_{t-1} + w_{cf} \odot c_{t-1} + b_f) \tag{9}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \sigma_c(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \tag{10}$$

$$o_t = \sigma_o(W_{xo}x_t + W_{ho}h_{t-1} + w_{co} \odot c_t + b_o) \tag{11}$$

$$h_t = o_t \odot \sigma_h(c_t) \tag{12}$$

The introduction of gating functions in Hochreiter and Schmidhuber (1997) differed from traditional RNNs, and allowed substantially easier training for recurrent networks. The gate activation vectors, i_t, f_t, o_t, represent the input, forget, and output gates respectively. Each neuron stores an internal cell activation

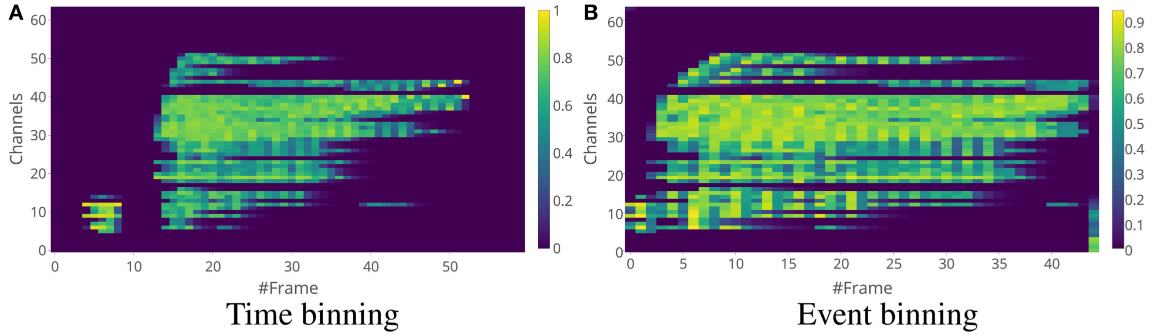


FIGURE 7 | Exponential feature examples for the same word as in **Figure 4**. The time window length for time binning in **(A)** is 5 ms and the number of events in a single frame for event binning in **(B)** is 25. One main difference between the spike count features and the exponential features is that the exponential feature values are in the range between 0 and 1, while the spike count feature values depend on the volume of the spikes in the time window.

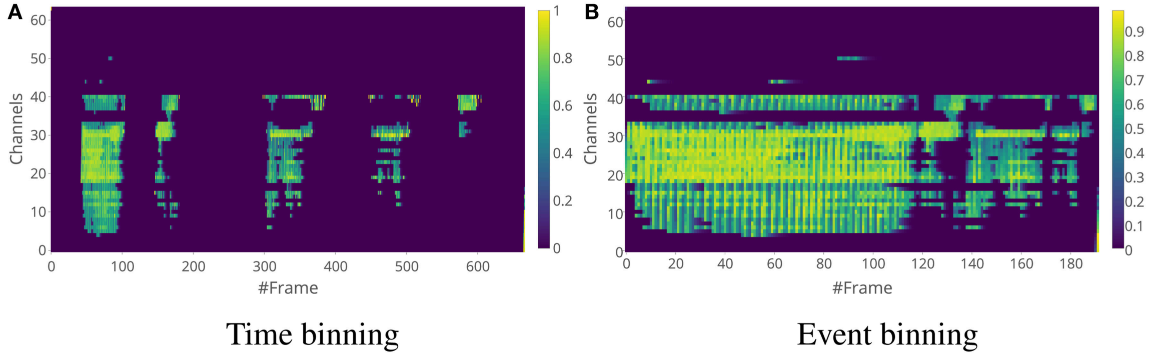


FIGURE 8 | Exponential feature examples for the same sequence as in **Figure 5**. The window length for time binning in **(A)** is 5 ms and the number of events in a single frame for event binning in **(B)** is 25.

vector c_t, while the input and hidden state vectors are x_t and h_t, respectively. A sigmoidal nonlinearity, $y = 1/(1+e^{-x})$, is applied to constrain the gates to lie between 0 and 1, and applied to the gates with σ_i, σ_f, and σ_o for the input, forget, and output gates. For these gates, each gate has a weight parameter for the input x and the hidden state h, resulting W_{xi} and W_{hi}, W_{xf} and W_{hf}, W_{xo} and W_{ho} for the input, forget, and hidden gates, respectively. Additionally, each gate has a bias b_i, b_f, and b_o for the input, forget, and output gates. The $\odot$ notation signifies an elementwise (Hadamard) product, implying that each cell state c_t is a linear interpolation between the previous cell state (controlled by f_t) and the new cell state (controlled by i_t). Finally, the cell state is transformed by the output gate o_t to produce a new hidden state h_t. Optionally, peephole connections, Gers and Schmidhuber (2000) w_{ci}, w_{cf}, and w_{co}, are commonly employed for the cell state c_t to further influence the input, forget, and output gates.

2.3.2. Gated Recurrent Units

Another commonly used gated architecture is the GRU architecture. The primary difference compared to LSTM is the removal of one gate, which results in faster training and execution time while achieving approximately the same accuracy in most tasks. The form employed in this work is the most common implementation from Chung et al. (2014):

$$r_t = \sigma_r(W_{xr}x_t + W_{hr}h_{t-1} + b_r) \tag{13}$$

$$u_t = \sigma_u(W_{xu}x_t + W_{hu}h_{t-1} + b_u) \tag{14}$$

$$c_t = \sigma_c(W_{xc}x_t + r_t \odot (W_{hc}h_{t-1}) + b_c) \tag{15}$$

$$h_t = (1 - u_t) \odot h_{t-1} + u_t \odot c_t \tag{16}$$

Similar to the above, there are the gate states r_t and u_t, referred to as the reset and update gates, as well as a combination gate and state c_t called the candidate state. As above, each consists of the application of a matrix multiplication of a weight vector (W_{xr}, W_{xu}, W_{xc}) to the input (x_t), as well as another matrix multiplication of a weight vector (W_{hr}, W_{hu}, W_{xc}) to the previous hidden state (h_{t-1}), for the reset, update, and candidate gates respectively. For the two pure gates, these two terms are summed with a bias (b_r and b_u) and the logistic sigmoid nonlinearity $y = 1/(1 + e^{-x})$ is applied to constrain each gate to lie between 0 and 1. For the candidate state, the reset gate is applied elementwise to the previous hidden state, and the bias b_c is added before the candidate state is transformed nonlinearly using the same logistic sigmoid. Finally, the new hidden state h_t is the result of a linear mixture of the update gate elementwise applied to the candidate state, and the previous hidden state controlled by the complement of the update gate.

2.3.3. Phased LSTM

The Phased LSTM model, which was introduced in Neil et al. (2016), equips the LSTM model with the ability to process irregularly-sampled continuous-time sequences through the application of a novel time gate k_t. This time gate, similar to other gates, produces a continuous value between 0 and 1 but is instead controlled by an external timing input. Each neuron has independent, learnable timing parameters that allow the neuron's time gate to execute a rhythmic wake-sleep cycle over time. When the time gate is open (close to 1), the neuron performs as a normal LSTM neuron does; when the time gate closes (close to 0), the neuron performs no updates until its next wake period. Other neurons, however, can still inspect a sleeping neuron's state. When continuous time sequences are applied, the timestamp of the event controls which subset of neurons are updated, and permits calculations based on the rhythmic wake-sleep cycle of the neurons in response timestamp itself.

Rigorously, the opening and closing of the gate is a periodic oscillation controlled by three parameters: a period τ that controls the duration, a shift s that applies a phase shift offset, and an on ratio r_{on} that determines the duration of the open period. The time (khronos) gate k_t can be calculated as:

$$\phi_{i,t} = \frac{(t - s_i) \bmod \tau_i}{\tau_i}, \quad k_{i,t} = \begin{cases} \dfrac{2\phi_{i,t}}{r_{on,i}}, & \text{if } \phi_{i,t} < \dfrac{1}{2} r_{on,i} \\ 2 - \dfrac{2\phi_{i,t}}{r_{on,i}}, & \text{if } \dfrac{1}{2} r_{on,i} < \phi_{i,t} < r_{on,i} \\ \alpha \phi_{i,t}, & \text{otherwise} \end{cases} \tag{17}$$

The neuron index i indicates which parameters are neuron-specific ($\phi_{i,t}$, $k_{i,t}$, s_i, τ_i, $r_{on,i}$) and which are global (t, α). Here, $\phi_{i,t}$ is introduced as an auxiliary variable to represent the percentage of the phase within the rhythmic cycle, ranging from 0 to 100%. There are three piecewise phases of the operation of the gate functionally represented in Equation (17): an open and rising phase (during the first half of r_{on}), an open and falling phase (during the second half of r_{on}) and an off phase. The linear slopes of the rising and falling phase have a constant gradient to preserve strong gradient information, in the same manner that allows ReLUs to train so well (LeCun et al., 2015). Further, note a leak is applied during the off phase with analogy to the leaky rectified linear unit (He et al., 2015) to permit the flow of gradient information even when the neuron is off. However, after training, the leak can be set to zero (i.e., $\alpha = 0$) and thus truly off, so no updates need be performed when the neuron is in the closed or sleep phase of the cycle. This continuous-time equation is defined at all time points t but requires no computation between sampled data points, allowing irregularly-spaced points in time to be effectively used within this framework as the neurons have an explicit model of time. The LSTM equations from above can then be rewritten to permit arbitrary time points j rather than timestep indices, using a proposed cell state $\widetilde{c_j}$ and proposed hidden state $\widetilde{h_j}$ controlled by the time gate k_j:

$$\begin{aligned}
i_j &= \sigma_i(W_{xi}x_j + W_{hi}h_{j-1} + w_{ci} \odot c_{j-1} + b_i) \qquad (18)\\
f_j &= \sigma_f(W_{xf}x_j + W_{hf}h_{j-1} + w_{cf} \odot c_{j-1} + b_f) \qquad (19)\\
\widetilde{c_j} &= f_j \odot c_{j-1} + i_j \odot \sigma_c(W_{xc}x_j + W_{hc}h_{j-1} + b_c),\\
&\quad c_j = k_j \odot \widetilde{c_j} + (1 - k_j) \odot c_{j-1} \qquad (20)\\
o_j &= \sigma_o(W_{xo}x_t + W_{ho} + w_{co} \odot \widetilde{c_j} + b_o) \qquad (21)\\
\widetilde{h_j} &= o_j \odot \sigma_h(\widetilde{c_j}),\\
&\quad h_j = k_j \odot \widetilde{h_j} + (1 - k_j) \odot h_{j-1} \qquad (22)
\end{aligned}$$

The sparseness in time of computation (typically, with r_{on} set to 5%) allows this implementation to be far sparser than traditional gated implementations in computation while maintaining high performance. Furthermore, as timesteps are no longer required and the neuron has an explicit model of time, even raw spike events can be directly used with Phased LSTM. For further information, refer to the formulation of Phased LSTM in Neil et al. (2016) or one of its publicly-available implementations[1,2].

2.4. Datasets

This paper introduces the N-TIDIGITS18 dataset by playing the audio files from the TIDIGITS dataset to the CochleaAMS1b sensor. The dataset is publicly accessible at http://sensors.ini.uzh.ch/databases.html. The dataset includes both single digits and connected digit sequences, with a vocabulary consisting of 11 digits ("oh," "zero" and the digits "1" to "9"). Each digit sequence is of length 1–7 spoken digits. There is a total of 55 male and 56 female speakers in the training set with a total of 8,623 training samples, while the testing set has a total of 56 male and 53 female speakers with a total of 8,700 testing samples.

The entire dataset is used or a reduced version of the dataset is used where only the single digit samples are used for training and testing. In the single digits dataset, there are two samples for each of the 11 single digits from every speaker, with a total of 2,464 training samples and 2,486 testing samples. The N-TIDIGITS18 dataset with all the samples was used to train a sequence classification task while the digit samples were used to train a digit recognition task. For most of our training, unless specified, we only use events from one ear and one neuron.

2.5. Network Architectures and Training Criterion

2.5.1. GRU/LSTM Architectures

Two network models were trained separately for the digit recognition task and the sequence classification task. For the digit recognition task, the network consists of 2 GRU layers with 100 units each, followed by a fully connected layer of 100 units with a ReLU activation followed by a Softmax classification layer. For the sequence classification task, each

[1] https://github.com/dannyneil/public_plstm

[2] https://www.tensorflow.org/api_docs/python/tf/contrib/rnn/PhasedLSTMCell

network consists of a fully connected layer of 100 units with SELU activation (Klambauer et al., 2017) followed by 2 LSTM layers of 100 units each followed by the final classification layer. The recently introduced SELU activation helps with regularization of the network by pushing the neuronal activations of the corresponding layer to zero mean and unit variance without the need for batch normalization. The SELU activation function was used over other activation functions mainly because the overall accuracy was significantly improved by using it.

The network for the digit recognition task was trained using a categorical cross entropy objective, while the network for the sequence classification task was trained using the Connectionist Temporal Classification (CTC) objective (Graves et al., 2006). For the CTC objective on sequence classification, the accuracy metrics used were the label error rate and the phrase error rate. For the label error rate, we first calculate the average edit distance (Levenshtein, 1966) between the correct label sequences and the corresponding predicted label sequences. The edit distance between two sequences is the minimum number of insertions, deletions and substitutions required to transform one into the other. The label error is then given by the ratio of the calculated average edit distance and the total number of labels in the correct label sequences. This metric is not a strict proper fraction for its value can go above 1. The phrase error rate is given by the ratio of the correctly predicted label sequences and the total number of label sequences.

All networks were trained on the Tensorflow framework (Abadi et al., 2015) using Adam optimizer with a learning rate of 0.001 over 200 epochs. All the presented accuracy numbers are based on at least three experimental runs. The network and simulation parameters are summarized in **Table 1**.

2.5.2. Phased LSTM Architecture

The single-event architecture was used on the raw input spikes. Because of the volume of input spikes and the difficulty in training extremely long sequences, only one neuron (out of four possible neurons) from one ear was used, resulting in sequences of approximately 4,000 spikes. Because a raw spike address and the corresponding spike time was used, an embedding layer of size 40 was used. As in Neil et al. (2016), a multi-resolution embedding layer downsamples the address by 1, 2, 4, and 16, and concatenates the 10-dimensional embedded feature from each result together. This allows learning features across multiple pitches (neuron addresses) as well as learning features particular to each pitch. After the embedding layer, two layers of 250 Phased LSTM neurons are included, with period $\tau \sim \exp(\mathcal{U}(0,3))$

TABLE 1 | Summary of the different training parameters used in this study.

Network	Model architecture	Batch size	No.of epochs
GRU RNN	2x 100 GRU - 100 Dense (ReLU) - 10 Softmax	128	200
LSTM RNN	100 Dense (SELU) - 2x 100 LSTM - 10 Dense	128	200
Phased LSTM	2x 250 Phased LSTM - 10 Dense	16	50

Tha Adam optimizer with a learning of 0.001 was used for all the networks.

TABLE 2 | Summary of investigated models on N-TIDIGITS18 dataset.

Feature type	Sensor	Task	Classifier	Accuracy (%)
MFCC		**Digit**	**GRU RNN**	**97.90**
Binned frames (fixed bins/sample)*	AMS1b	Digit	SVM	95.08
Constant time bins**	AMS1b	Digit	CNN	87.65
Constant time bins**	AMS1b	Digit	GRU RNN	82.82
Single events (raw data)	AMS1b	Digit	Phased LSTM	87.75
Data-driven time-binned features	AMS1b	Digit	Phased LSTM	91.25[a]
Constant time bins	AMS1b	Digit	GRU RNN	86.4
Exponential features	**AMS1b**	**Digit**	**GRU RNN**	**90.9**
Constant time bins	AMS1c	Digit	GRU RNN	88.6
Exponential features	**AMS1c**	**Digit**	**GRU RNN**	**91.1**
Constant time bins	AMS1b	Sequence	LSTM RNN	86.1[b]
Exponential features	**AMS1b**	**Sequence**	**LSTM RNN**	**87.3**[b]

The MFCC features are extracted from the original TIDIGITS dataset.
[a] *Events from all neurons and both ears used in training.*
[b] *Label accuracy on sequences.*
**Abdollahi and Liu (2011).*
***Neil and Liu (2016).*

milliseconds (with $x \sim \mathcal{U}(a, b)$ implying a random draw of x from the uniform distribution between limits a and b), shifts $s \sim \mathcal{U}(0, 100)$ milliseconds, and the on ratio $r_{on} = 0.05$ resulting in 5% activity. The output of the second Phased LSTM layer is fully connected via a dense layer to the ten output classes.

For the N-TIDIGITS18 dataset, only 40% of 0.5 ms time bins (also timesteps) have data (with an average of 3.6 spikes per time bin), running at a 2.5× increase in speed over calculating every timestep. Furthermore, the number of bins are far fewer in number than the number of input spikes as would be the case with processing the raw input data. Compared to processing every spike sequentially in the full dataset (all neurons, all ears), there are now 30 times fewer timesteps, resulting in a dramatic decrease in training time when training on data-driven bins.

All Phased LSTM networks were trained on the Lasagne framework (Dieleman et al., 2015) using the Adam optimizer and a learning rate of 0.001 over 50 epochs.

3. RESULTS

We present the network accuracy results of the different pre–processing methods on the audio classification tasks based on the N-TIDIGITS18 dataset when these features are presented to the different recurrent models.

3.1. Comparison of Feature Representations

The performance of the pre–processed features are tested through two classification tasks. The first is a word recognition task on the single digit samples in the dataset, and the second is a sentence prediction task on the connected digit samples. The classifiers used for different tasks and their performances on the different feature types are shown in **Table 2**. The results in the table show that the networks using the exponential features consistently perform better than the spike count features across both the tasks. The Phased LSTM networks which were used to process either the raw event data or the data-driven bins outperform the spike count features, and produce similar accuracies as GRU RNNs with exponential features.

Although the Phased LSTM network takes a longer time to train because the input sequences of single spikes are longer, one advantage of this method over the other pre-processing methods is that there are no hyper parameters that need fine tuning such as the time window length parameter T_l used for binning or the tau parameter τ used in the exponential features.

The performance of the method using the binned frames on a Support Vector Machine (SVM) classifier is better than all the other methods but this method relies on access to the whole sample which is then converted into a fixed number of bins per sample and unfortunately cannot be used on a real-time recognition system.

3.2. Optimizing Parameters

As discussed in section 3.1, both spike count features and exponential features have a few hyperparameters that need fine tuning for optimal performance. The network hyperparameters were optimized once using a small validation split on the training data from the N-TIDIGITS18 dataset. The small validation dataset was created by using 10% of the training samples while the model was trained on the other 90% of the samples.

The variation of the error rates on the τ parameter and the T_l parameter for the exponential features in the sequence classification task are shown in **Figures 9A,B**, respectively. In **Figure 9A**, we can see that the error rate is very high for τ less than 2 ms, and then remains fairly steady for values of τ till 5 ms and then rises slowly as τ increases. The optimal value of τ is related to the mean inter-spike interval in the frequency channels. While for low values of τ the features do not properly encode the history of the events and thus do not perform well, while for increasing τ the exponential feature values saturate toward 1 and thus do not provide enough contrast among the features for the classification tasks to decode properly.

In **Figure 9B**, we can see that the error rates increase with larger T_l, because with a bigger T_l, the exponential features get smeared because of averaging over more events. But it should be considered that with increasing T_l, the sequences to be processed become shorter which makes the recurrent network training easier and efficient.

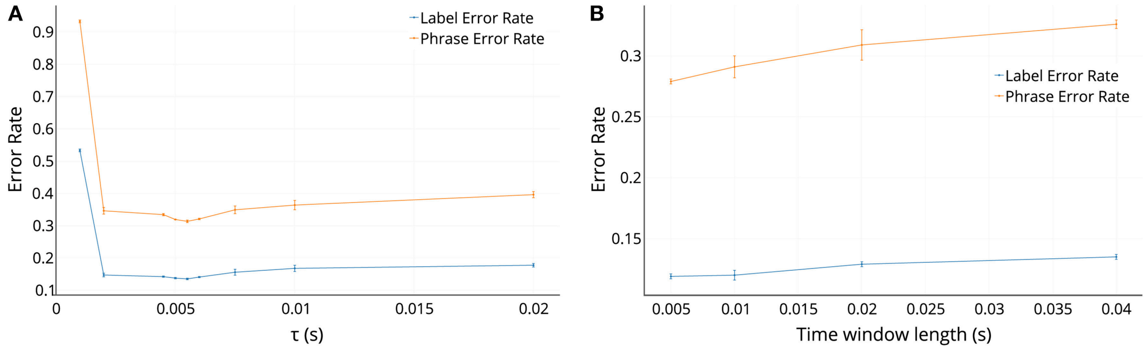


FIGURE 9 | The effect of τ and time window length parameters on the error rates in the sequence classification task in the case of exponential features. The window length for time binning in **(A)** is 5 ms and the value of τ used in **(B)** is 5.5 ms. Although using the optimal values gives lower error rates, using a larger time window length than the optimal value of 5 ms would help in having shorter sequences to process.

These plots suggest that the optimal τ is 5.5 ms and the optimal T_l is 5 ms, which were also the values used in the experiments for **Table 2**. Although the optimal value of T_l was 5 ms, using a parameter value of 40 ms would help save training time and number of computes required to process the network per unit time since there are fewer frames to process per unit time. This advantage comes though at the expense of a reduced accuracy of about 1.6% (88.1% for 5 ms and 86.5% for 40 ms) on the validation set.

3.3. Comparison of Processing Times for Feature Generation

To compare the processing times of conventional MFCC with the SC and exponential filters, we used a Raspberry Pi 3 Model B, with ARM Cortex A-53 processor. We used a feature frame rate of 100 Hz. We created random data for both the raw audio and the event data. For raw audio we used a sampling rate of 16 kHz with uniformly sampled data between $[-1, 1]$ at every sampling point. Since the observed average spike rate was about 3,400 Hz for the N-TIDIGITS18 dataset, for the computational test with event data, we generated Poisson spike trains with a total spike rate of 4,000 Hz. Across 100,000 runs, the average processing times per frame on the hardware were 5.79 ms for the MFCC features, 0.72 ms for the SC features and 2.2 ms for the exponential features. Thus the event-driven features are faster to generate by a factor of 2.2X for exponential features and 8X for SC features. This result is not surprising given the computational simplicity of the cochlea features afforded by the sensor preprocessing.

4. DISCUSSION

In this work, we performed a comparative study of the performance accuracy of a gated recurrent neural network that processes either the raw audio spikes or framed features extracted by different spike processing methods. We demonstrate the use of a recent LSTM model called the Phased LSTM which operates on raw audio spikes. We compared the performance accuracy of this model to that of the standard gated recurrent neural networks, the LSTM and the GRU networks, that processed framed features extracted by different spike processing methods.

The results show that it is possible to achieve a good performance through processing the raw events using the Phased LSTM model. This model, designed for use on long sequences, makes use of the inherent timing information present in the spikes. Although the training time is long because the model has to learn to process more timesteps, there are no meta-parameters to tune in the feature generation.

Alternatively, pre-processing the spikes to produce framed features is appealing because the input sequences to the recurrent networks will be shorter than the sequence of raw events. For both the single digit and digit sequence datasets, the network classification accuracy is higher by approximately 2.5% when using exponential features over spike count features. It should be noted that the results are obtained on the N-TIDIGITS18 dataset, a relatively small dataset. We will investigate in the future if the higher accuracy from using exponential features extend to larger datasets.

We hypothesize that the increased accuracy from exponential features is due to two reasons. First, interspike intervals in the spike streams carry information useful for the classification task and therefore exponential features are more desirable. Second, the encoded exponential feature values are real-valued and range between 0 and 1 while the spike count feature values are quantized in discrete quantities of 1. Having real-valued input features might help during training of the recurrent networks.

Even though the accuracy results from using the pre-processed audio spike frames were lower than the results obtained from using MFCC features, the focus of this work is to present improved methods for processing the outputs of event driven sensors in real-time applications. Our evaluation of the average processing time per frame on a Raspberry Pi shows that generation of the event-driven features is faster than that of MFCCs by a factor of 2–8 depending on the cochlea features used. We also aim towards an event-driven system where processing would be activated only if there are sufficient spikes from the sensor, e.g., the processing is inactivated during silent periods.

The results presented here serve as a baseline for future studies on algorithms that process spikes from spiking audio sensors. The pre-processing methods and the LSTM/GRU networks used in the work above are already implemented in real time (Anumula et al., 2017) using the jAER framework (Delbruck, 2008). The N-TIDIGITS18 dataset used in our experiments is publicly accessible at http://sensors.ini.uzh.ch/databases.html.

AUTHOR CONTRIBUTIONS

JA performed the RNN experiments and contributed to the writing, DN performed the Phased LSTM experiments and contributed to the writing, TD contributed to discussions on the feature extraction methods and assisted in the development of the hardware infrastructure of the cochlea boards, and S-CL contributed to the design of the experiments and the writing.

FUNDING

This work was partially supported by the European Union's Horizon 2020 research and innovation program under grant agreement No 644732.

ACKNOWLEDGMENTS

We thank members of the INI Sensors group, particularly, Xiaoya Li for help with the development of the recording setup. We are also thankful to the reviewers for their valuable comments toward improving the manuscript.

REFERENCES

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., et al. (2015). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Available online at: http://www.tensorflow.org/

Abdollahi, M., and Liu, S. C. (2011). "Speaker-independent isolated digit recognition using an aer silicon cochlea," in *2011 IEEE Biomedical Circuits and Systems Conference (BioCAS)* (San Diego, CA), 269–272. doi: 10.1109/BioCAS.2011.6107779

Amir, A., Taba, B., Berg, D., Melano, T., McKinstry, J., Di Nolfo, C., et al. (2017). "A low power, fully event-based gesture recognition system," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Honolulu, HI), 7243–7252. doi: 10.1109/CVPR.2017.781

Anumula, J., Neil, D., Li, X., Delbruck, T., and Liu, S.-C. (2017). "Live demonstration: event-driven real-time spoken digit recognition system," in *Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS)* (Baltimore, MD). doi: 10.1109/ISCAS.2017.8050394

Barranco, F., Fermuller, C., Aloimonos, Y., and Delbruck, T. (2016). A dataset for visual navigation with neuromorphic methods. *Front. Neurosci*. 10:49. doi: 10.3389/fnins.2016.00049

Bengio, Y., Simard, P., and Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *Trans. Neur. Netw*. 5, 157–166. doi: 10.1109/72.279181

Berner, R., Brandli, C., Yang, M., Liu, S. C., and Delbruck, T. (2013). "A 240 × 180 10mW 12μs latency sparse-output vision sensor for mobile applications," in *2013 Symposium on VLSI Circuits* (Kyoto), C186–C187.

Brette, R., and Gerstner, W. (2005). Adaptive exponential integrate-and-fire model as an effective description of neuronal activity. *J. Neurophysiol.* 94, 3637–3642. doi: 10.1152/jn.00686.2005

Chakrabartty, S., and Liu, S. C. (2010). "Exploiting spike-based dynamics in a silicon cochlea for speaker identification," in *Proceedings of 2010 IEEE International Symposium on Circuits and Systems* (Paris), 513–516. doi: 10.1109/ISCAS.2010.5537578

Chan, V., Liu, S. C., and van Schaik, A. (2007). AER EAR: a matched silicon cochlea pair with address event representation interface. *IEEE Trans. Circ. Syst. I Regul. Papers* 54, 48–59. doi: 10.1109/TCSI.2006.887979

Cho, K., van Merrienboer, B., Gülçehre, Ç., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR* abs/1406.1078.

Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555.

Cohen, G. K., Orchard, G., Ieng, S. H., Tapson, J., Benosman, R. B., and van Schaik, A. (2016). Skimming digits: neuromorphic classification of spike-encoded images. *Front. Neurosci.* 10:184. doi: 10.3389/fnins.2016.00184

Delbruck, T. (2008). "Frame-free dynamic digital vision," in *Proceedings of International Symposium on Secure-Life Electronics*, vol. 1 (Tokyo: University of Tokyo), 21–26.

Diehl, P. U., Neil, D., Binas, J., Cook, M., Liu, S. C., and Pfeiffer, M. (2015). "Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing," in *2015 International Joint Conference on Neural Networks (IJCNN)* (Killarney), 1–8. doi: 10.1109/IJCNN.2015.7280696

Dieleman, S., Schlüter, J., Raffel, C., Olson, E., Sønderby, S. K., Nouri, D., et al. (2015). *Lasagne: First Release*. doi: 10.5281/zenodo.27878

Esser, S. K., Merolla, P. A., Arthur, J. V., Cassidy, A. S., Appuswamy, R., Andreopoulos, A., et al. (2016). Convolutional networks for fast, energy-efficient neuromorphic computing. *Proc. Natl. Acad. Sci. U.S.A.* 113, 11441–11446. doi: 10.1073/pnas.1604850113

Farabet, C., Paz, R., Pèrez-Carrasco, J., Zamarreño-Ramos, C., Linares-Barranco, A., LeCun, Y., et al. (2012). Comparison between frame-constrained fix-pixel-value and frame-free spiking-dynamic-pixel convnets for visual processing. *Front. Neurosci.* 6:32. doi: 10.3389/fnins.2012.00032

Gers, F. A., and Schmidhuber, J. (2000). "Recurrent nets that time and count," in *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks (IJCNN)*, Vol. 3 (Como: IEEE), 189–194. doi: 10.1109/IJCNN.2000.861302

Graves, A. (2013). Generating sequences with recurrent neural networks. arXiv preprint arXiv:1308.0850.

Graves, A., Fernández, S., Gomez, F., and Schmidhuber, J. (2006). "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd International Conference on Machine Learning, ICML '06* (Pittsburg, CA; New York, NY: ACM), 369–376. doi: 10.1145/1143844.1143891

He, K., Zhang, X., Ren, S., and Sun, J. (2015). "Delving deep into rectifiers: surpassing human-level performance on imagenet classification," in *The IEEE International Conference on Computer Vision (ICCV)* (Santiago), 1026–1034. doi: 10.1109/ICCV.2015.123

Hochreiter, S., and Schmidhuber, J. (1997). Long short-term memory. *Neural Comput.* 9, 1735–1780. doi: 10.1162/neco.1997.9.8.1735

Klambauer, G., Unterthiner, T., Mayr, A., and Hochreiter, S. (2017). "Self-normalizing neural networks," in *Advances in Neural Information Processing Systems 30*, eds I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Long Beach, CA: Curran Associates, Inc.), 972–981.

Lagorce, X., Ieng, S.-H., Clady, X., Pfeiffer, M., and Benosman, R. B. (2015). Spatiotemporal features for asynchronous event-based data. *Front. Neurosci.* 9:46. doi: 10.3389/fnins.2015.00046

Lagorce, X., Orchard, G., Gallupi, F., Shi, B. E., and Benosman, R. (2016). Hots: a hierarchy of event-based time-surfaces for pattern recognition. *IEEE Trans. Patt. Anal. Mach. Intell.* 39, 1346–1359. doi: 10.1109/TPAMI.2016.2574707

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature* 521, 436–444. doi: 10.1038/nature14539

Leonard, R. G., and Doddington, G. (1993). *Tidigits ldc93s10.* Philadelphia, PA: Linguistic Data Consortium.

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions and reversals. *Sov. Phys. Doklady* 10:707.

Li, C.-H., Delbrück, T., and Liu, S.-C. (2012). "Real-time speaker identification using the AEREAR2 event-based silicon cochlea," in *Proceedings of IEEE International Symposium on Circuits and Systems* (Seoul), 1159–1162.

Lichtsteiner, P., Posch, C., and Delbruck, T. (2008). A 128 × 128 120 db 15 μs latency asynchronous temporal contrast vision sensor. *IEEE J. Solid State Circuits* 43, 566–576. doi: 10.1109/JSSC.2007.914337

Liu, S.-C., Mesgarani, N., Harris, J., and Hermansky, H. (2010). "The use of spike-based representations for hardware audition systems," in *Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS)* (Paris), 505–508.

Liu, S. C., van Schaik, A., Minch, B. A., and Delbruck, T. (2014). Asynchronous binaural spatial audition sensor with 2 × 64 × 4 channel output. *IEEE Trans. Biomed. Circ. Syst.* 8, 453–464. doi: 10.1109/TBCAS.2013.2281834

Lungu, I., Corradi, F., and Delbruck, T. (2017). "Live demonstration: convolutional neural network driven by dynamic vision sensor playing RoShamBo," in *Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS)* (Baltimore, MD). doi: 10.1109/ISCAS.2017.8050403

Moeys, D. P., Corradi, F., Kerr, E., Vance, P., Das, G., Neil, D., et al. (2016). "Steering a predator robot using a mixed frame/event-driven convolutional neural network," in *2016 Second International Conference on Event-based Control, Communication, and Signal Processing (EBCCSP)* (Krakow), 1–8. doi: 10.1109/EBCCSP.2016.7605233

Neil, D., and Liu, S. C. (2016). "Effective sensor fusion with event-based sensors and deep network architectures," in *2016 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2282–2285.

Neil, D., Pfeiffer, M., and Liu, S.-C. (2016). "Phased LSTM: accelerating recurrent network training for long or event-based sequences," in *Advances in Neural Information Processing Systems* (Barcelona), 3882–3890.

O'Connor, P., Neil, D., Liu, S.-C., Delbruck, T., and Pfeiffer, M. (2013). Real-time classification and sensor fusion with a spiking Deep Belief Network. *Front. Neurosci.* 7:178. doi: 10.3389/fnins.2013.00178

Orchard, G., Jayawant, A., Cohen, G., and Thakor, N. (2015). Converting static image datasets to spiking neuromorphic datasets using saccades. *Front. Neurosci.* 9:437. doi: 10.3389/fnins.2015.00437

Pérez-Carrasco, J. A., Zhao, B., Serrano, C., Acha, B., Serrano-Gotarredona, T., Chen, S., et al. (2013). Mapping from frame-driven to frame-free event-driven vision systems by low-rate rate coding and coincidence processing–application to feedforward convnets. *IEEE Trans. Patt. Anal. Mach. Intell.* 35, 2706–2719. doi: 10.1109/TPAMI.2013.71

Posch, C., Serrano-Gotarredona, T., Linares-Barranco, B., and Delbruck, T. (2014). Retinomorphic event-based vision sensors: bioinspired cameras with spiking output. *Proc. IEEE* 102, 1470–1484. doi: 10.1109/JPROC.2014.2346153

Rueckauer, B., Lungu, I.-A., Hu, Y., Pfeiffer, M., and Liu, S.-C. (2017). Conversion of continuous-valued deep networks to efficient event-driven networks for image classification. *Front. Neurosci.* 11:682. doi: 10.3389/fnins.2017.00682

Schmidhuber, J. (2014). Deep learning in neural networks: an overview. *CoRR* abs/1404.7828.

Serrano-Gotarredona, T., and Linares-Barranco, B. (2015). Poker-DVS and MNIST-DVS. Their history, how they were made, and other details. *Front. Neurosci.* 9:481. doi: 10.3389/fnins.2015.00481

Stromatias, E., Neil, D., Pfeiffer, M., Galluppi, F., Furber, S. B., and Liu, S.-C. (2015). Robustness of spiking Deep Belief Networks to noise and reduced bit precision of neuro-inspired hardware platforms. *Front. Neurosci.* 9:222. doi: 10.3389/fnins.2015.00222

Szűcs, A. (1998). Applications of the spike density function in analysis of neuronal firing patterns. *J. Neurosci. Methods* 81, 159–167. doi: 10.1016/S0165-0270(98)00033-8

Tapson, J., Cohen, G., Afshar, S., Stiefel, K., Buskila, Y., Wang, R., et al. (2013). Synthesis of neural networks for spatio-temporal spike pattern recognition and processing. arXiv:1304.7118.

Yang, M., Chien, C. H., Delbruck, T., and Liu, S. C. (2016). “A 0.5V 55 μW 64×2-channel binaural silicon cochlea for event-driven stereo-audio sensing,” in *2016 IEEE International Solid-State Circuits Conference (ISSCC)* (San Francisco, CA), 388–389. doi: 10.1109/ISSCC.2016.7418070

Yang, M., Liu, S. C., and Delbruck, T. (2015). A Dynamic Vision Sensor with 1% temporal contrast sensitivity and in-pixel asynchronous delta modulator for event encoding. *IEEE J. Solid State Circ.* 50, 2149–2160. doi: 10.1109/JSSC.2015.2425886

Zai, A., Bhargava, S., Mesgarani, N., and Liu, S.-C. (2015). Reconstruction of audio waveforms from spike trains of artificial cochlea models. *Front. Neurosci.* 9:347. doi: 10.3389/fnins.2015.00347

Zhao, B., Ding, R., Chen, S., Linares-Barranco, B., and Tang, H. (2015). Feedforward categorization on AER motion events using cortex-like features in a spiking neural network. *IEEE Trans. Neural Netw. Learn. Syst.* 26, 1963–1978. doi: 10.1109/TNNLS.2014.2362542

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

ORIGINAL RESEARCH
published: 28 March 2018
doi: 10.3389/fnins.2018.00160

A Comparison of Low-Complexity Real-Time Feature Extraction for Neuromorphic Speech Recognition

Jyotibdha Acharya[1], *Aakash Patil*[2], *Xiaoya Li*[3], *Yi Chen*[2], *Shih-Chii Liu*[3] and *Arindam Basu*[2]*

[1] HealthTech NTU, Interdisciplinary Graduate School, Nanyang Technological University, Singapore, Singapore, [2] School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, Singapore, [3] Institute of Neuroinformatics, University of Zurich and ETH Zurich, Zurich, Switzerland

This paper presents a real-time, low-complexity neuromorphic speech recognition system using a spiking silicon cochlea, a feature extraction module and a population encoding method based Neural Engineering Framework (NEF)/Extreme Learning Machine (ELM) classifier IC. Several feature extraction methods with varying memory and computational complexity are presented along with their corresponding classification accuracies. On the N-TIDIGITS18 dataset, we show that a fixed bin size based feature extraction method that votes across both time and spike count features can achieve an accuracy of 95% in software similar to previously report methods that use fixed number of bins per sample while using ∼3× less energy and ∼25× less memory for feature extraction (∼1.5× less overall). Hardware measurements for the same topology show a slightly reduced accuracy of 94% that can be attributed to the extra correlations in hardware random weights. The hardware accuracy can be increased by further increasing the number of hidden nodes in ELM at the cost of memory and energy.

Keywords: silicon cochlea, neural engineering framework, extreme learning machine, neuromorphic, real-time

OPEN ACCESS

Edited by:
*Huajin Tang,
Sichuan University, China*

Reviewed by:
*Timothée Masquelier,
Centre National de la Recherche Scientifique (CNRS), France
Sadique Sheik,
BioCircuits Institute (BCI), University of California, San Diego, United States*

***Correspondence:**
*Arindam Basu
arindam.basu@ntu.edu.sg*

Specialty section:
This article was submitted to Neuromorphic Engineering, a section of the journal Frontiers in Neuroscience

Received: *09 October 2017*
Accepted: *27 February 2018*
Published: *28 March 2018*

Citation:
Acharya J, Patil A, Li X, Chen Y, Liu S-C and Basu A (2018) A Comparison of Low-Complexity Real-Time Feature Extraction for Neuromorphic Speech Recognition. Front. Neurosci. 12:160. doi: 10.3389/fnins.2018.00160

1. INTRODUCTION

Considerable progress has been made recently in machine learning for speech recognition tasks with the developments in traditional Gaussian Mixture Models and Hidden Markov Models to the more recent deep neural networks (Hinton et al., 2012). However, these models require very complicated processing of the input speech and are not suited for simple sensor nodes with limited power; nor do they perform well in the presence of large background noise (cocktail party problem). In contrast, the human auditory system is able to perform sound stream segregation easily. This has led to an interest in studying the biological auditory system and developing silicon models of cochleas that operate in an event-driven asynchronous fashion (Liu et al., 2014) much like the neurons in the auditory pathway. These event-based asynchronous cochlea sensors implement a bio-mimetic filtering circuit that produces spikes at the output in response to input sounds (Chan et al., 2007; Liu and Delbruck, 2010; Liu et al., 2014). The AEREAR2 sensor has been used previously for typical speech recognition problems such as speaker identification (Chakrabartty and Liu, 2010; Li et al., 2012) and digit recognition (Abdollahi and Liu, 2011; Anumula et al., 2018). The inter-spike intervals and channel specific spike counts are used as features for these tasks. High classification accuracy (95%) was reported using these features for a speaker independent digit recognition task using a software implementation of support vector machine (SVM) based implementation (Abdollahi and Liu, 2011). However, this method required the storage of the entire spike response of the cochlea channels to one spoken digit so that the spikes can be pre-processed prior to classification resulting in huge memory requirements.

In parallel, there has been considerable progress in developing neural models of cognition and a particularly popular one based on population coding is the Neural Engineering Framework (NEF) (Eliasmith and Anderson, 2004; Eliasmith et al., 2012). proposes a framework for neural simulations where the input is non-linearly encoded using random projections and linearly decoded to model the required function. The typical NEF architecture consists of three layers, the input layer, a hidden layer consisting of a large number of non-linear neurons and an output layer consisting of linear neurons. In the encoding phase, the inputs are multiplied with random weights and passed to the non-linear neurons. The non-linear function can be any neural model from the spiking Leaky-Integrate-and-Fire model to more complex biological models (Stewart, 2012). With the use of recurrent connections, NEF can also be used for modeling even dynamic functions. NEF has been proved to be an efficient tool for implementing large scale brain models like SPAUN (Stewart et al., 2012) and therefore, is being widely used in neuromorphic research community.

A similar model has been separately developed in the machine learning community. Termed as the Extreme Learning Machine (ELM) (Huang et al., 2006), it also uses a three layered architecture with random projection of the input and linear decoding. It is essentially a feedforward network and does not have feedback connections allowed in NEF—hence, it may be considered as a sub-category of NEF architectures. It has been used in a variety of applications ranging from neural decoding (Chen et al., 2016) and epileptic seizure detection (Song et al., 2012) to speech recognition (Deng et al., 2017) and big data applications (Akusok et al., 2015) in the past. Low power hardware implementations of this algorithm have also been reported recently (Yao and Basu, 2017). Since we also use a feedforward network in this work, we will refer to our algorithm as ELM in the rest of the paper acknowledging that it can be referred to as NEF as well.

In this work, we bring together these two developments of neuromorphic spiking cochlea sensors and population encoding based ELM hardware to lay the groundwork for a low power bio-inspired real-time sound recognition system. Several different low-complexity feature extraction methods that do not require storage of entire spike trains are explored in this paper and tradeoffs between memory/computation requirements and recognition accuracy are presented. Measured accuracy results using the silicon cochlea in Liu et al. (2014) and ELM chip in Yao and Basu (2017) are presented for the TIDIGITS dataset with 11 spoken digit classes. Though the entire processing of the signal does not use spike times, our method still uses "physical" computation in the cochlea and NEF/ELM blocks which is the essence of neuromorphic engineering as described in Mead (1990).

The remainder of this paper is organized as follows: section 2 details the hardware and the proposed methods. section 3 computes the hardware complexity for the proposed methods. section 4 reports the results for both software simulation and hardware measurements and finally, section 5 presents a discussion on the obtained results.

2. MATERIALS AND METHODS

The basic architecture of our proposed speech recognition system is shown in **Figure 1**. The speech input is acquired by the Dynamic Audio Sensor and the spikes produced are then passed to the feature extraction block. The extracted features are then sent to an Extreme Learning Machine for classification. For the experiments in this paper, we have simulated the feature extraction block in software only, but the feature extraction techniques described here can easily be implemented in hardware using standard microcontrollers. Measured results from hardware are presented for the cochlea and the ELM chip.

2.1. Silicon Cochlea and Recordings

The N-TIDIGITS18 dataset (Anumula et al., 2018) used in this work, consists of recorded spike responses of a binaural 64-channel silicon cochlea (Chan et al., 2007) in response to audio waveforms from the original TIDIGITS dataset (Leonard, 1984). The silicon cochlea and later generations of this design, model the basilar membrane, inner hair cells and spiral ganglion cells of the biological cochlea. The basilar membrane is implemented by a cascaded set of 64 second-order band-pass filters, each with its own characteristic frequency. The output of each filter goes to an inner hair cell block which performs a half-wave rectification of its input. The output of the inner hair cell goes to a ganglion cell block implemented by a spiking neuron circuit. The spike output is transmitted off-chip using the asynchronous address-event representation (AER). The binaural chip is connected to microphones emulating left and right ears. The circuit architecture of one ear is shown in **Figure 2A**. Circuit details are described in Chan et al. (2007) and Liu et al. (2014).

In the recordings, impulses are added at the beginning and end of the audio digit files so that the start and end points of the spike recordings are visible. The impulses lead to spike responses from all channels. **Figures 2B,C** show two sample spikes of digit "2". Dots correspond to spike outputs from the 64 channels of one ear of the cochlea.

2.2. Preprocessing Methods

To obtain the feature vectors from the spike recordings of the silicon cochlea, we used the spike count per window or bin for two modes of binning with two binning strategies which resulted in four preprocessing techniques as shown in **Table 1**. In the methods described, we used bins of width W and used counters to count the number of spikes across different channels within that bin. The output of the ith bin can be represented as $X_W(i)$ where X_W is a $[1 \times C]$ vector containing spike counts across C channels. Next, we cascaded the bin outputs to produce the feature vectors. The 4 modes differ in the choice of W and the number of vectors to be cascaded.

2.2.1. Binning Modes

We used two modes for binning the cochlea images to extract features. The first one is time based binning (1A, 1B) where the

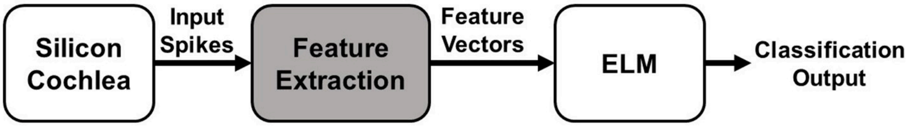


FIGURE 1 | Block diagram of the proposed speech recognition system. The shaded block for feature extraction is implemented in software in this work while the other two blocks are implemented in hardware.

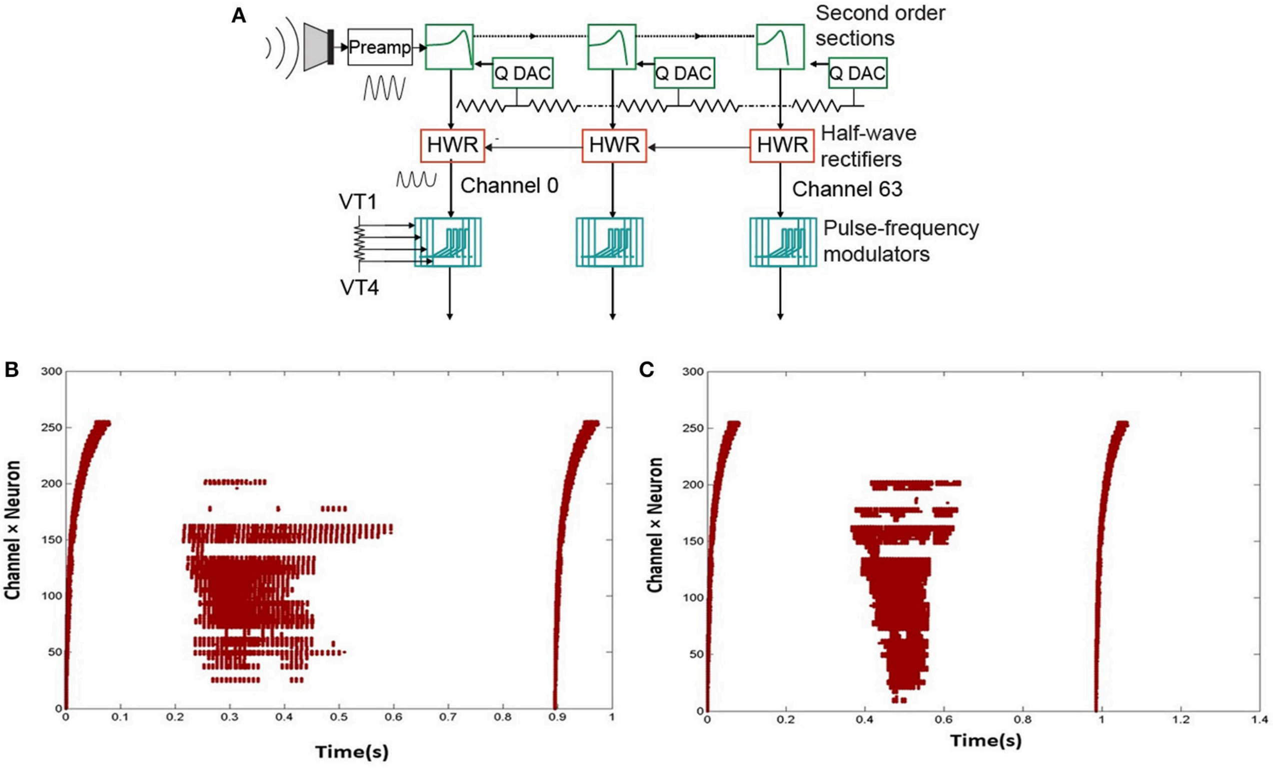


FIGURE 2 | (A) Circuit architecture of one ear of the Dynamic Audio Sensor (adapted from Liu et al., 2014). The input goes through a cascaded set of 64 bandpass filters. The output of each of the filters is rectified. This rectified signal then drives an integrate-and-fire neuron model. **(B,C)** Two sample spikes of digit "2." Dots correspond to spike outputs from the 64 channels of one ear of the cochlea.

TABLE 1 | Preprocessing methods.

Mode / Binning	Time	Spike count
Fixed Bin Size	1A	2A
Fixed No. of Bins	1B	2B

whole spike sample is divided into several bins based on time or duration of the sample (T_{sample}). The second one is spike count based binning (2A, 2B) where we binned the spike trains based on the total number of spikes in the sample (N_{sample}). While the time based strategy captures the spike density variation in cochlear images quite well, it completely ignores the temporal variation (longer vs. shorter samples). On the other hand, the spike count based strategy captures the temporal variation but ignores the spike density variation (dense vs. sparse samples).

2.2.2. Binning Strategies

For all modes, we used two binning methods, (A) fixed bin size and (B) fixed number of bins. These methods are described below for the time based binning mode only to avoid repetition. A similar philosophy applies to the case of spike count based binning.

2.2.2.1. Fixed number of bins

In this method, the total number of bins per sample is fixed or static. As a result, in the time mode of binning, the longer samples produce longer bins than shorter samples (as shown in **Figure 3**). If the number of bins per sample is fixed at B_{sta}, and the

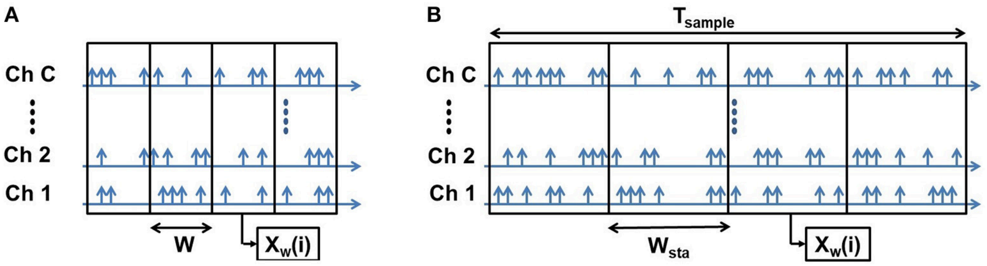


FIGURE 3 | Fixed number of bins: Both the short **(A)** and long **(B)** samples have the same number of bins but the bin width (W) is shorter for short samples and longer for long samples.

corresponding bin width is w_{sta} for a sample, the total duration of the sample, T_{sample} is given by:

$$T_{sample} = w_{sta} \times B_{sta} \tag{2.2.1}$$

In this method, we explicitly set the value of B_{sta} and w_{sta} is determined by:

$$w_{sta} = T_{sample}/B_{sta}. \tag{2.2.2}$$

If the total number of spikes per sample is denoted as N_{sample} and the average number of spikes/bin/channel is denoted by $\overline{n}_{spikes}$, we can write:

$$N_{sample} = B_{sta} \times C \times \overline{n}_{spikes} \tag{2.2.3}$$

The output of each bin ($Xw(i)$) is cascaded to produce the feature vector F = [Xw(1) Xw(2)...Xw(B)]. So the dimension of the feature vector is $C \times B_{sta}$. Thus, there is a clear trade-off between the feature vector size and temporal resolution of the bins. Higher temporal resolution leads to a larger feature vector size and therefore higher classification complexity and vice-versa. The primary disadvantage of this method is that it requires a priori information about the duration of total spike count of the sample before the binning. So, the entire sample needs to be stored first and afterwards binning is done on the sample. Thus, the memory requirement of this method is quite high and the latency is equal to the sample duration. Finally, use of a dynamic bin size removes inter-sample variability of temporal resolution by performing an intrinsic normalization. The longer samples are compressed as a result of longer bin sizes while shorter samples expanded as a result of shorter bin sizes. This is the feature extraction method used in previous work such as Abdollahi and Liu (2011).

In the spike count mode, the total number of spikes N_{sample} summed across all channels and time is divided into a fixed number of bins (B_{sta}) leading to a limit (N_{sample}/B_{sta}) on total number of spikes per bin. Whenever this limit is reached, it defines the formation of a bin. Spike counts in all channels are frozen to create a feature vector and this process repeats.

2.2.2.2. Fixed bin size

In the fixed bin size method, the size of bins is predetermined in terms of time duration or spike count based on the mode of binning. As a result, the longer samples produce larger number of bins while shorter samples produce smaller number of bins (as shown in **Figure 4**).

Denoting the number of bins per sample using this strategy as B_{fix}, setting the bin width to w_{fix} and using the same notations as the previous method, we can write:

$$T_{sample} = w_{fix} \times B_{fix} \tag{2.2.4}$$

In this method, we explicitly set the value of w_{fix} and the corresponding value of B_{fix} is determined by:

$$B_{fix} = T_{sample}/w_{fix} \tag{2.2.5}$$

The total number of spikes per sample is given by:

$$N_{sample} = B_{fix} \times C \times \overline{n}_{spikes} \tag{2.2.6}$$

As the number of bins produced by the samples (B_{fix}) is different for different samples and the ELM classification algorithm requires a fixed feature vector size, we needed to find an optimum number of bins that produce overall high accuracy irrespective of sample duration. Larger number of bins results in increased feature vector size which in turn makes the classification task more difficult and computationally expensive while smaller number of bins result in feature vectors that sample the spike recordings coarsely and thus, miss the finer variations over the sample durations. Our initial experiments suggested that, for number of bins 8-12 the classification accuracy is optimum. Therefore, we decided to fix the number of bins to 10. So, The dimension of the feature vector is $10 \times C$. Based on the bin size and total sample duration, one of two cases can occur:

Case I: $B_{fix} \geq 10$

If the sample produced more than 10 bins, we will keep the output of only first 10 bins to produce the feature vectors while ignoring the rest. These bins are then cascaded to produce the feature vector $F = [Xw(1)Xw(2)...Xw(10)]$. In this case,

$$T_{sample} \geq w_{fix} \times 10 \tag{2.2.7}$$

For this case, we only use a fraction of total spikes to produce the feature vector. If the number of spikes used is given by N_{used}, we

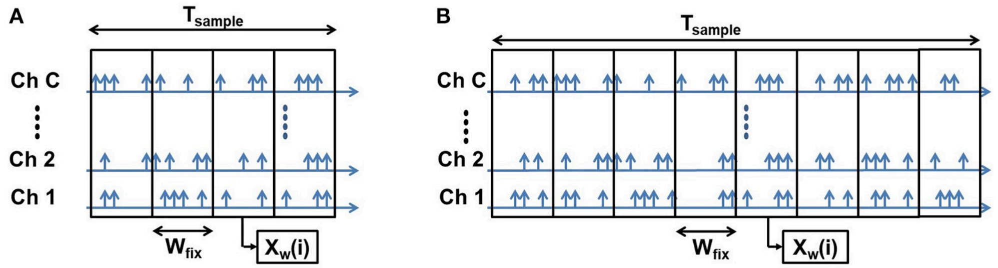


FIGURE 4 | Fixed Bin Size: Both the short **(A)** and long **(B)** samples have the same bin width (W). A short sample produces smaller number of bins and a long sample produces larger number of bins.

can write:

$$N_{used} = 10 \times C \times \overline{n}_{spikes} \leq B_{fix} \times C \times \overline{n}_{spikes} = N_{sample} \qquad (2.2.8)$$

Case II: $B_{fix} < 10$
For the samples that produce less than 10 bins for a given bin size, zero padding is used to produce the feature vectors. In this case,

$$T_{sample} < w_{fix} \times 10$$

For this case, we use all the spikes in the sample to produce the feature vector. So,

$$N_{used} = B_{fix} \times C \times \overline{n}_{spikes} = N_{sample} \qquad (2.2.9)$$

So, generalizing the two cases, we can express N_{used} as:

$$N_{used} = min\{10 \times C \times \overline{n}_{spikes}, B_{fix} \times C \times \overline{n}_{spikes}\} \qquad (2.2.10)$$

There is no need to store the sample in memory for this method since the feature vectors are directly produced from the samples with predetermined bin sizes. Thus, memory required for this method is quite low. As we require only 10 bin outputs to form a feature vector, the latency is independent of the sample duration unlike the previous strategy. The primary drawback of this strategy is that to obtain fixed feature vector sizes, we have to use a fixed number of bins (10 in our case) to produce the feature vectors and therefore, for larger samples, the rest of the bin outputs are discarded. So, there is a loss of information in this strategy. Moreover, as the bin size is fixed, this method does not provide any input duration normalization like the earlier strategy. A similar fixed spike count based frame size strategy has been used by Moeys et al. (2016) for feature extraction.

2.3. Classification Methods

2.3.1. Extreme Learning Machine: Algorithm

The ELM is a three layer feedforward neural network introduced in Huang et al. (2006) shown in **Figure 5A**. The output of the ELM network with L hidden neurons is given by:

$$\mathbf{o} = \sum_{\mathbf{i}}^{\mathbf{L}} \beta_{\mathbf{i}} \mathbf{H}_{\mathbf{i}} = \sum_{\mathbf{i}}^{\mathbf{L}} \beta_{\mathbf{i}} \mathbf{g}(\mathbf{w}_{\mathbf{i}}^{\mathbf{T}} \mathbf{x} + \mathbf{b}_{\mathbf{i}}) \qquad (2.3.1)$$

where x is a d-dimensional input vector, b_i is the bias of individual neurons, w_i and β_i are input and output weights respectively. $g(.)$ is the non-linear activation function (sigmoid function is commonly used) and h_i is the output of the ith hidden neuron. While the weights w_i and b_i are chosen from any random distribution and need not be tuned, the output weights β_i need to be tuned during training. So the basic task in this architecture is to find the least square solution of β given targets of training data:

$$\mathbf{Minimize}_{\beta} : ||\mathbf{H}\beta - \mathbf{T}||^{\mathbf{2}}, \qquad (2.3.2)$$

where T is the target of training data. The optimal solution of β is given by

$$\tilde{\beta} = H^{\dagger T}, \qquad (2.3.3)$$

where $H^{\dagger}$ is the Moore-Penrose pseudoinverse of H (Penrose, 1955). The simplest method to find $H^{\dagger}$ is using orthogonal projection:

$$\begin{aligned} H^{\dagger} &= (H^T H)^{-1} H^T \; if \; H^T H \; is \; non-singular \\ H^{\dagger} &= H^T (HH^T)^{-1} \; if \; HH^T \; is \; non-singular. \end{aligned} \qquad (2.3.4)$$

Moreover, taking advantage of the concepts from ridge regression (Hoerl and Kennard, 1970), a constant is added to the diagonal of $H^T H$ or HH^T which results in a solution that is more stable and has better generalization performance. C is a tunable hyperparameter. Several regularization techniques have been explored for determining the optimal value of H to reduce training time and number of hidden neurons (Huang et al., 2012). The simple architecture of the ELM network makes it a suitable candidate for hardware implementation.

2.3.2. Extreme Learning Machine: Hardware

For the classification task, we have used software ELM as well as hardware measurements on the neuromorphic ELM chip described in Yao and Basu (2017).

The digital implementations of ELM can benefit from the software simulations of the ELM shown in this paper. The architecture of the ELM chip is shown in **Figure 5B**. The 128 input digital values are converted to analog currents using

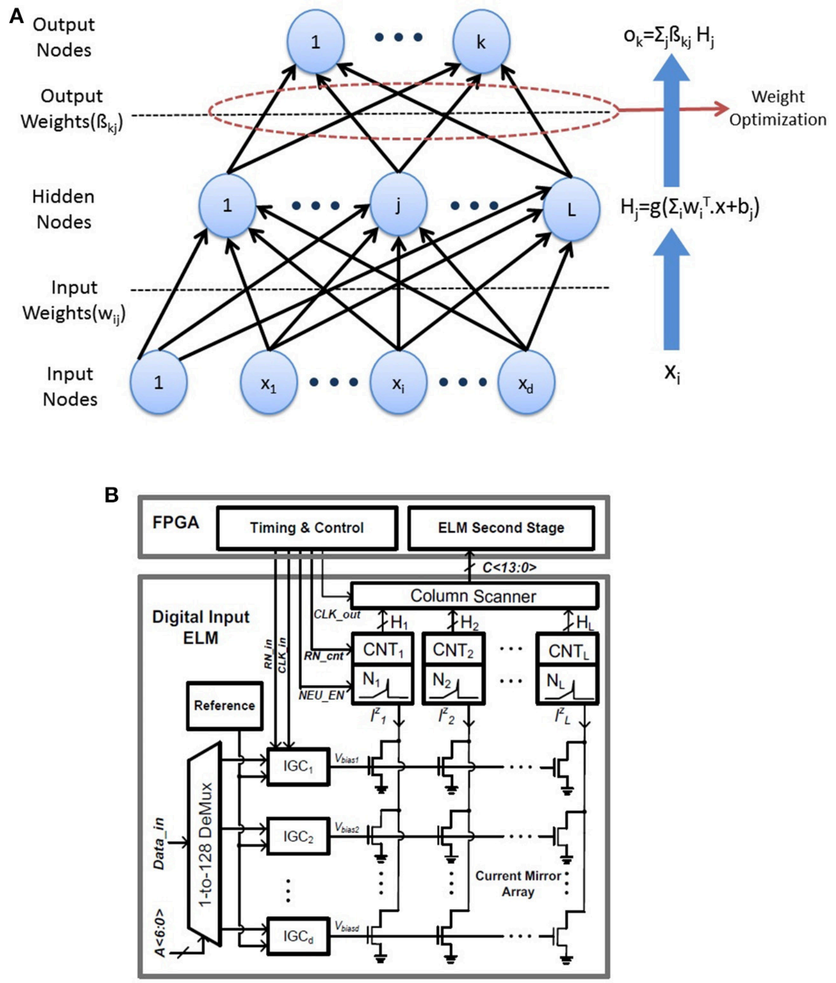


FIGURE 5 | (A) ELM network architecture: The weights w_{ij} in the first layer are random and fixed while only the second layer weights need to be trained. **(B)** Architecture of the neuromorphic ELM IC (adapted from Patil et al., 2015).

current mode DACs which are multiplied by random weights in a 128 × 128 current mirror array (CMA). The random weights are generated by the physical mismatch of transistors in the CMA. The 128 output currents are converted to spikes using an array of 128 integrate and fire neurons. The corresponding firing rates are obtained by an array of digital counters while the second stage of ELM is performed in digital on a FPGA. While the software ELM uses random weights with a uniform random distribution, the chip generates random weights w_{ij} with lognormal distribution. This is due to the exponential relation of current and threshold voltage (V_T) in the sub-threshold regime which leads to mismatch induced weights of the form

$$w = e^{\Delta V_T/U_T} \tag{2.3.5}$$

where ΔV_T denotes mismatch between threshold voltages of a pair transistors forming a current mirror. However, lognormal distributions have positive mean and software simulations show that zero mean weights result in higher classification accuracy. Hence, a simple digital post-processing is used on the outputs to obtain zero mean random numbers. Instead of directly feeding the chip output h_i to the second stage, the difference h'_i of neighboring neurons were used. So, the modified output of the hidden layer is given by:

$$h'_i = h_i - h_{(i+1)mod(128)},\ i = 1, 2, .., 128 \tag{2.3.6}$$

As shown in Patil et al. (2015), any weight distribution w_{ij} can become a zero mean distribution w'_{ij} using this technique. We will refer to this as log difference weight for the rest of this paper. Finally, instead of using typical non-linearities like sigmoid or tanh as $g(.)$, we have used an absolute value (abs) function as the preferred non-linearity. While software simulations show similar or slightly better classification accuracy for an absolute value non-linearity compared to typical non-linearities, it has several other advantages over them. Absolute value is a non-saturating non-linearity and so feature vectors need not be normalized before being passed to the ELM unlike saturating non-linearities.This reduces the computational burden. Moreover, the hardware implementation of abs non-linearity is much simpler than sigmoid or similar non-linearities.

3. HARDWARE COMPLEXITY

In this section we will discuss the hardware complexity comprising computations and memory requirements for the classifier and the two feature extraction methods described earlier. For our calculation, we assume that the time stamp of a spike is encoded using 32 bits and the channel address of the spike is 6 bits. The average number of spikes per sample is assumed to be N_{sample} and the spike counter size is $b_{counter}$ bits. The number of computations (N_{comp}) can be written as the sum of two components:

$$N_{comp} = N_{feature} + N_{ELM} \tag{3.0.1}$$

where $N_{feature}$ is the number of computations for feature extraction while N_{ELM} is the number of computations required for classification by ELM.

The total memory required (M_{total}) can be written as sum of two components:

$$M_{total} = M_{feature} + M_{ELM} \tag{3.0.2}$$

where $M_{feature}$ is the memory required for feature extraction while M_{ELM} is the memory required for classification by ELM.

3.1. Feature Extraction

3.1.1. Fixed Number of Bins

For the fixed number of bins method, the entire sample needs to be stored first and bin sizes are to be determined later. So, the memory required to store the spike information of an entire sample (time stamp and channel count) is

$$M_{samples} = 38 \times N_{sample}\ bits \tag{3.1.1}$$

Now, if the number of bins is B_{sta}, a total of $B_{sta} \times C$ counters are required to count the spikes and produce the feature vector. Therefore, the memory required to store a feature vector is given by:

$$M_{feature\_vector} = B_{sta} \times C \times b_{count}\ bits \tag{3.1.2}$$

So, from Equations 3.1.1 and 3.1.2 the total memory requirement for fixed number of bins method is

$$\begin{aligned} M_{feature} &= 38 \times N_{sample} + B_{sta} \times C \times b_{count}\ bits \\ &= 38 \times B_{sta} \times C \times \overline{n}_{spikes} + B_{sta} \times C \times b_{count}\ bits \end{aligned} \tag{3.1.3}$$

In terms of computations, there will be a counter increment for each spike resulting in N_{sample} operations per sample. Also, for each spike, the time stamp needs to be compared with the bin boundary to determine when to reset counters. Hence the total number of operations per sample is given by:

$$N_{feature} = N_{sample} + N_{sample} = 2N_{sample} \tag{3.1.4}$$

3.1.2. Fixed Bin Size

For the fixed bin size method, the feature vectors are produced directly from the sample as the bin sizes are pre-determined. Thus, there is no need for storing the sample in memory. The only memory required in fixed bin size method is for storing the feature vectors. Since we cascade 10 bin outputs to produce a feature vector in this method, using calculations similar to above, we get:

$$M_{feature} = M_{feature\_vector} = 10 \times C \times b_{count}\ bits \tag{3.1.5}$$

Finally, the total number of operations per sample is the total number of counter increments which is equal to the number of spikes used to produce the feature vector. So,

$$N_{feature} = N_{used} = min\{10 \times C \times \overline{n}_{spikes}, B_{fix} \times C \times \overline{n}_{spikes}\}, \tag{3.1.6}$$

For the fixed bin size method, the memory requirement is significantly less than the fixed number of bins method as there is no need for storing the entire sample before feature extraction. Furthermore, pre-determined bin sizes enable this method to be compatible with real-time speech recognition systems. The significant advantage of this method over the fixed number of bins method in terms of memory and energy requirements is further quantified in section 4.3.

3.2. Classification

N_{ELM} again has two parts due to multiply and accumulate (MAC) in the first and second layers of the network. Hence, N_{ELM} is given by the following:

$$N_{ELM} = D \times L + L \times C_o \tag{3.2.1}$$

where C_o is the number of output classes, D is the dimension of the feature vector and L is the number of hidden nodes. For our classification problem, number of output classes $C_o = 11$. Moreover, calculating log difference weights requires some additional subtractions ($= L$). Hence, the final value of N_{ELM} is given by:

$$N_{ELM} = D \times L + L \times C_o + L \tag{3.2.2}$$

Finally, the amount of memory (M_{ELM}) needed by the classifier is given by:

$$M_{ELM} = D \times L \times b_W + L \times C_o \times b_\beta \tag{3.2.3}$$

where b_W and b_β denote the number of bits to represent the first and second layer weights.

The energy requirement for the ELM in the custom implementation will depend on the energy required for each of these operations. Since multiplications are dominant, E_{MAC} is the prime concern. Since it has been shown that $E^{ana}_{MAC} < E^{dig}_{MAC}$ for the first stage with maximum number of multiplies (Chen et al., 2016), we have used an analog neuromorphic ELM hardware in this work. However, the findings of this work are applicable to a digital implementation of ELM on ASIC or on a microprocessor.

4. RESULTS

4.1. Software Simulations

In this section, we show the classification accuracies for different pre-processing strategies described in section 2.2 using a software ELM with uniform random weights and log difference weights. Though there are 64 (max. channel count) channels available in AEREAR2, only the first 54 channels were active for all the samples, therefore C = 54. All the results were obtained by averaging the classification accuracies over five randomized 90–10% train-test splits .

4.1.1. Fixed Number of Bins (1B, 2B)

For the fixed number of bins method, we have used $B_{sta} = 5, 10, 20,$ and 30 bins per sample for both time based and spike count based modes with number of hidden nodes in the classifier varying from $L = 500$ to $3,000$. The results for this experiment are plotted for both uniform random and log difference weights in **Figures 6A,B** for time based and in **Figures 6C,D** for spike based binning respectively. It can be seen that, for both modes, $B_{sta} = 10$ bins per sample produced maximum overall classification accuracy of around 96% for uniform random and 93.5% for log difference weights respectively. Also, the accuracies tend to initially increase with increasing values of L but eventually saturate and start decreasing due to over-fitting.

4.1.2. Fixed Bin Size (1A, 2A)

For the fixed bin size method (1A, 2A in **Table 1**), we have used 10–40ms bin sizes for time based binning and 300 spikes/bin to 600 spikes/bin bin sizes for spike count based binning with number of hidden nodes varying from 500 to 3,000.The results for this experiment are plotted for both uniform random and log difference weights in **Figures 7A,B** for time based and in **Figures 7C,D** for spike based binning respectively. It can be seen that, for time based mode, the maximum overall classification accuracy was obtained for 40 ms. We tried a bin size of up to 80 ms and found that the accuracy decreases beyond 40 ms. This is probably due to the fact that, while larger bin sizes ensure less loss of information at the end of a digit, it produces very small number of bins for shorter samples which results in their misclassification. For spike count based mode the maximum overall classification accuracy was obtained for 400 spikes/bin. Interestingly, even with fixed bin size features, we can obtain classification accuracies ∼95% for time based binning in both cases of uniform and log difference weights. Hence, this points

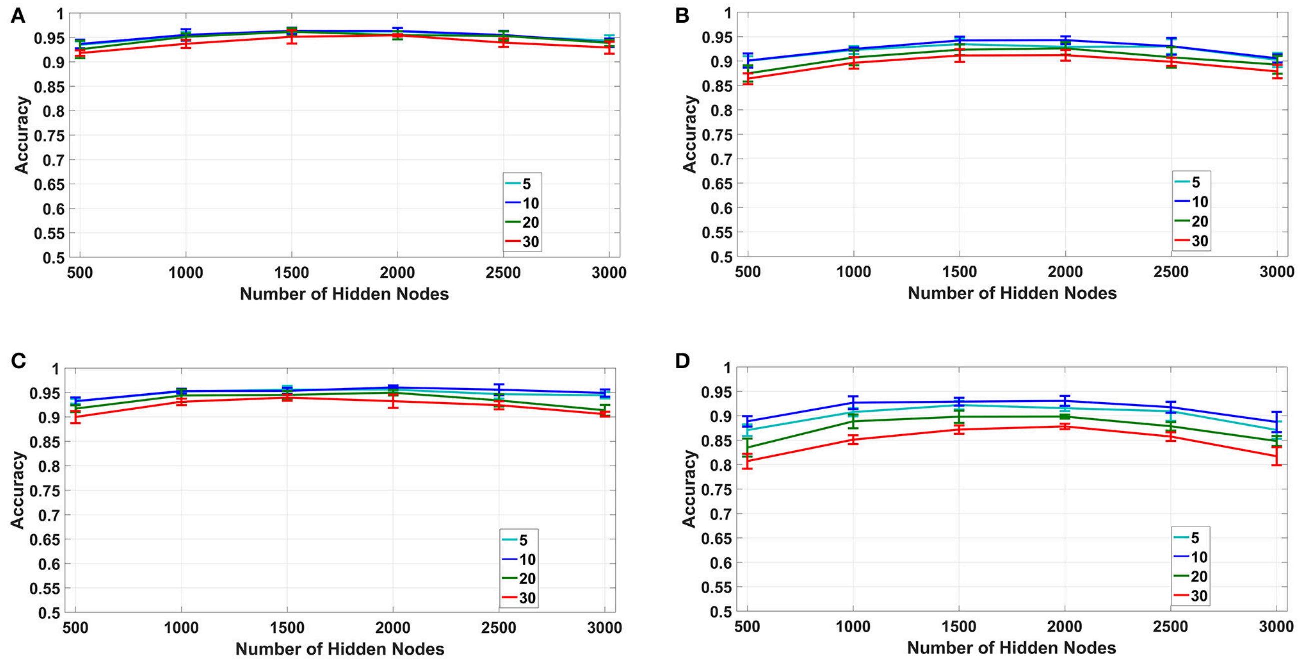


FIGURE 6 | Fixed number of bins: Accuracy vs. number of hidden nodes for different number of bins. **(A,B)**: Time based binning (1B):10 bins per sample shows highest overall accuracy. **(C,D)**: Spike count based binning (2B): 10 bins per sample shows highest overall accuracy.

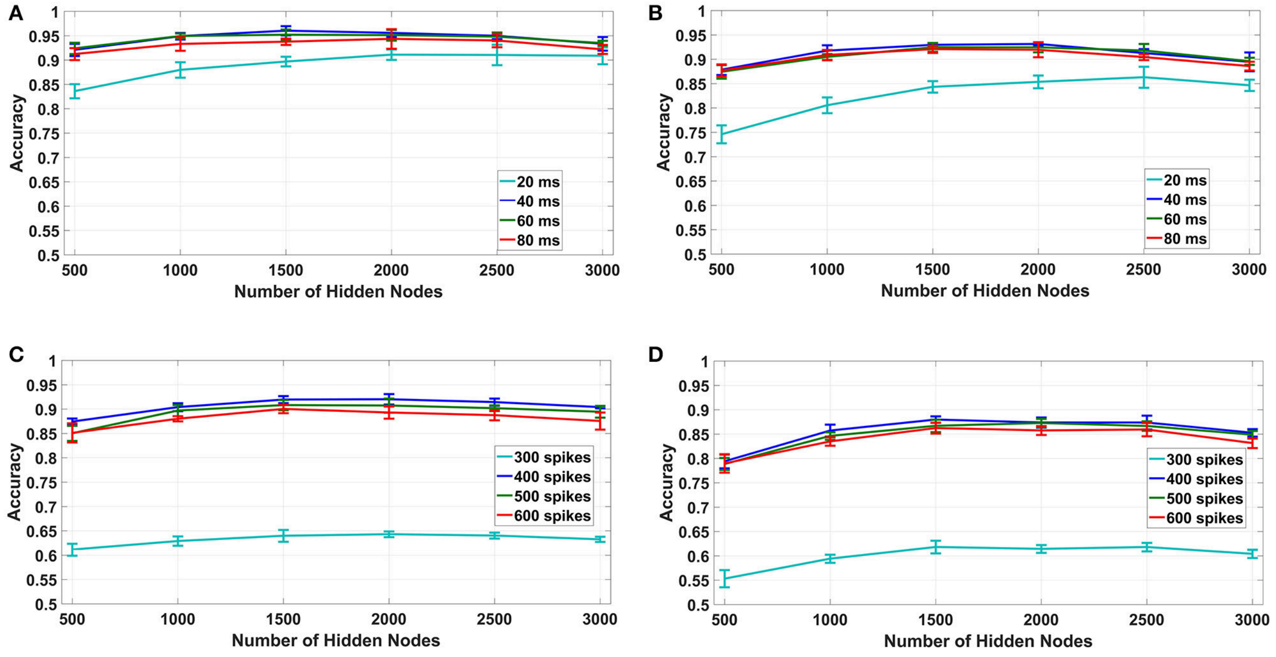


FIGURE 7 | Fixed bin size: accuracy vs. number of hidden nodes for different bin sizes. **(A,B)**: Time based binning (1A): 40 ms bin size shows highest overall accuracy. **(C,D)**: Spike count based binning (2A): 400 spikes/bin shows highest overall accuracy.

to a method for low hardware complexity feature extraction that also allows usage of analog sub-threshold ELM circuits with log difference weights. Second, the trend of increasing accuracies with increasing temporal bin size is due to the ELM being able to access larger parts of the speech sample. Lastly, the difference between spike count based binning and time based binning is very large in this case indicating that spike count alone is not a good distinguishing feature for fixed bin size.

4.1.3. Combined Binning

Out of the two binning strategies described in this paper, the fixed bin size method is more convenient to implement from a hardware perspective. Moreover, the memory and energy requirements of the fixed bin size method are much less than its counterpart as discussed in section 4.3. But as we have shown in section 4.1.2, the best case accuracy of the fixed bin size method is typically 2–3% less than that of fixed number of bins method. This is due to two factors: lack of input temporal normalization and loss of information due to discarded bins. To increase the accuracy of the fixed bin size method, we adopted a combined binning approach as shown in **Figure 8A**. In this fixed bin size strategy, the input data is processed in parallel using both time based and spike count based binning.The feature vectors produced are applied to their respective ELMs and the ELM outputs are combined (added) in the decision layer. The final output class is defined as the strongest class based on both strategies. **Figures 8B,C** compares the best case accuracies of time based binning (40 ms bin size), spike count based binning (400 spikes/bin bin size) and combined binning mode (combination of both). The combined binning mode not only outperforms both the time and spike count based modes, but also shows accuracies similar to the best case accuracies of fixed number of bins method for both type of weights.The reasons for this increased accuracy is further discussed in section 5.

4.2. Hardware Measurements

Finally, the proposed feature extraction methods were tested on a neuromorphic ELM IC described in Yao and Basu (2017) by feeding the chip with feature vectors produced by the methods described above. Due to the long testing times needed, we only tested the best accuracy cases of time based binning (40 ms bin size), spike count based binning (400 spikes/bin bin size) and combined binning (combination of the two). The accuracies obtained are shown in **Figure 9**. The optimum accuracy obtained by time based binning is slightly higher than that of spike count based binning while combined binning approach outperforms both of the methods. However, comparing this result with the earlier software simulations, we notice two differences. First, the accuracies obtained are slightly less than software and second, the accuracy increases with increasing *L*.

Possible reasons for this reduction in accuracy and its subsequent increase with increasing *L* are discussed in section 5.

4.3. Memory and Energy Requirement (Highest Accuracy Case Is Marked Red)

In this section, we will determine the memory and energy requirements of different post processing methods described. We have used the formulae derived in section 3 to determine the memory requirement and the computational complexity of different strategies. Moreover, we used the specifications of Apollo2 Ultra-Low Power Microcontroller for calculating pre-processing energy requirement ($10\mu A/MHz$ at $3.3V$[1]) and specifications of the neuromorphic ELM chip for calculating the classification energy requirement ($0.47pJ/MAC$, Yao and Basu, 2017). **Tables 2**, **3** show the memory requirement, computational complexity and average energy per sample of fixed number of bins and fixed bin size strategies assuming 1500 hidden nodes for the ELM. If we compare the best accuracy cases of both fixed bin size and fixed number of bins methods, these results show that fixed binning requires ∼50× less memory for feature extraction (∼ 3× overall) and ∼ 30% less energy compared to that of fixed number of bins method. Furthermore, as the combined binning requires approximately twice the memory and computational complexity than that of the simple time or spike count based binning methods, we can conclude that the combined binning strategy is able to produce accuracies similar to fixed number of bins method using ∼ 25× less memory for feature extraction (∼ 1.5× overall). Moreover, since the neuromorphic ELM chip uses mismatch induced random weights for the first layer of the ELM, no memory is required to store the first layer weights. Only, the second layer trained weights need to be stored in memory. The minimum resolution of the second layer weights (b_β) required for no loss of accuracy is found to be 8 bits.

[1] http://ambiqmicro.com/apollo-ultra-low-power-mcu/apollo2-mcu/

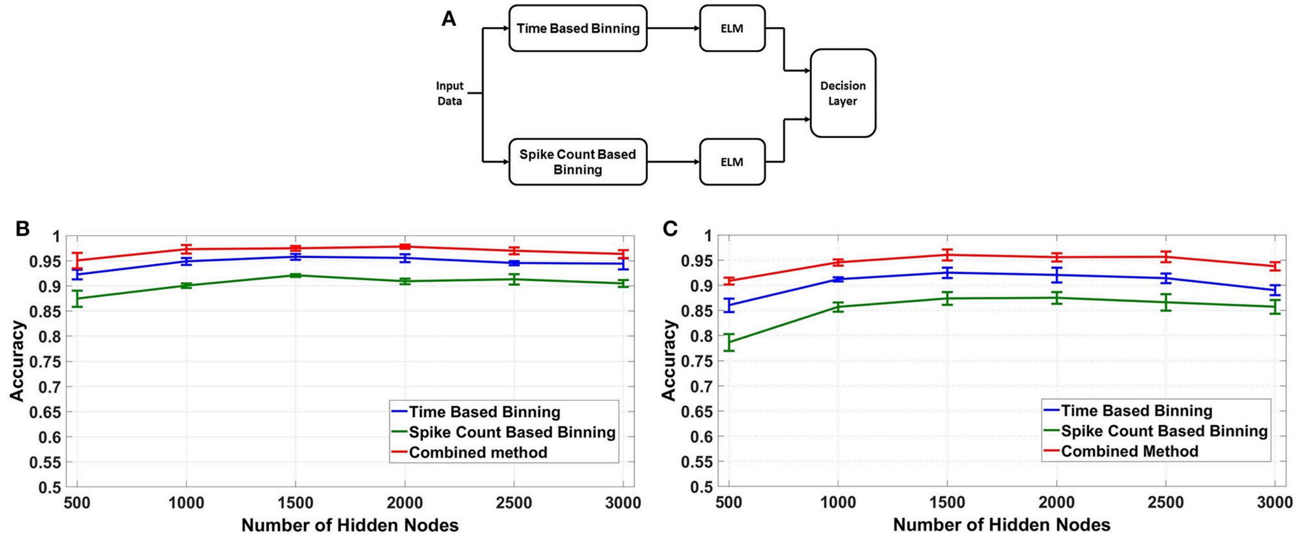


FIGURE 8 | (A) Combined Binning architecture for fixed bin size case by fusing the decisions of two ELMs operating in time based and spike count based modes respectively. **(B,C)** Comparison of binning modes, fixed bin size: Accuracy vs. Number of Hidden Nodes using different binning modes for fixed bin size, Combined Mode shows highest overall accuracy, comparable to fixed number of bins.

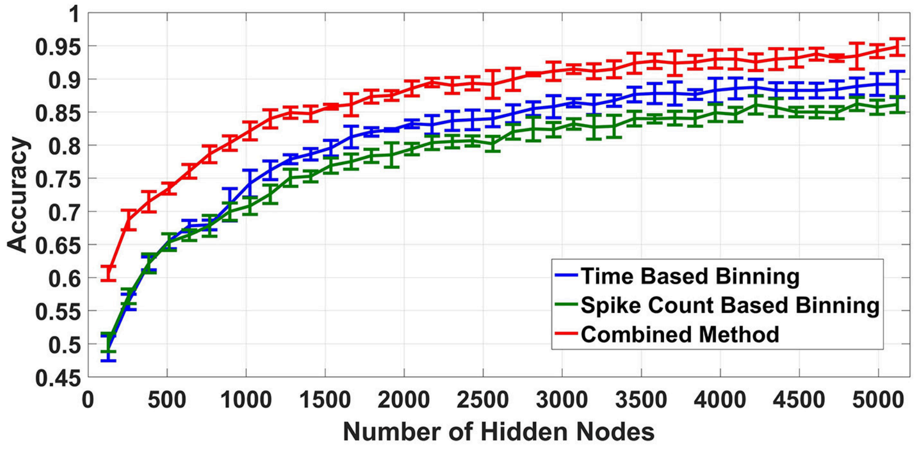


FIGURE 9 | Hardware classification accuracies for different binning strategies, Combined Binning strategy shows highest classification accuracy.

5. DISCUSSION

5.1. Hardware vs. Software ELM

One key observation from the results obtained is that the hardware ELM requires larger number of hidden nodes to obtain accuracies similar to the software simulations (compare **Figure 8** and **Figure 9**). While software simulations required around 2,000 hidden nodes to obtain optimum accuracy, the hardware required more than 5,000 hidden nodes to obtain comparable accuracies.This discrepancy can be ascribed to the higher correlation between input weights in the ELM IC. In an ideal ELM, the input weights of are assumed to be random and so, the correlation between successive columns of weights should be low. But in the ELM IC, the correlation between successive columns of weights are relatively higher due to chip architecture. Since the DACs converting the input digital number to a current is shared for each row, mismatch between the DACs introduce a systematic mismatch between rows. This systematic variation of the input weight matrix results in increased correlation between columns of input weights. **Figure 10** shows the histogram of inter column correlation coefficients for hardware weights and software simulated log normal weights. Greater correlation between hardware weights can alternatively thought of as a reduction in effective number of uncorrelated weights and thereby, a reduction in number of uncorrelated hidden nodes compared to software simulations. Therefore, the "effective" number of hidden nodes in hardware case is in fact smaller than the number of hidden nodes used in the IC. This explains the requirement of higher number of hidden nodes in hardware to match the performance of software simulations.

Another significant observation about the experimental results is that the combined strategy consistently outperforms both time based binning and spike count based binning methods for software as well as hardware simulations. This can be attributed to the synergy produced by combining two disparate representations of the input data (time based features and spike count based features) using a decision layer. To prove the importance of using two different representations, we have obtained the average confusion matrices for both time based

TABLE 2 | Memory and energy requirements for fixed number of bins method (1B,2B).

Bins/ Sample	5	10	20	30
Memory Required (Feature Extraction) (Kbits)	213	215	219	223
Memory Required (ELM Layer 2) (Kbits)	132	132	132	132
No.of Ops/sample (Feature Extraction) (Kops)	11	11	11	11
No. of MACs/sample (ELM Layer 1) (KMACs)	405	810	1,620	2,430
No. of MACs/sample (ELM Layer 2) (KMACs)	18	18	18	18
Energy Required (nJ/sample)	3,061	3,251	3,632	4,013

TABLE 3 | Memory and energy requirements for fixed bin size method (1A, 2A). Highest accuracy cases are marked red.

	Time based binning				Spike count based binning			
Bin Size	10 ms	20 ms	30 ms	40 ms	300 spikes /bin	400 spikes /bin	500 spikes /bin	600 spikes /bin
Memory Required (Feature Extraction) (Kbits)	4.3	4.3	4.3	4.3	4.3	4.3	4.3	4.3
Memory Required (ELM Layer 2) (Kbits)	132	132	132	132	132	132	132	132
No.of Ops/sample (Feature Extraction) (Kops)	0.7	1.4	1.8	2	2	3	4	5
No. of MACs/sample (ELM Layer 1) (KMACs)	810	810	810	810	810	810	810	810
No. of MACs/sample (ELM Layer 2) (KMACs)	18	18	18	18	18	18	18	18
Energy Required (nJ/sample)	2,232	2,301	2,340	2,360	2,360	2,459	2,558	2,657

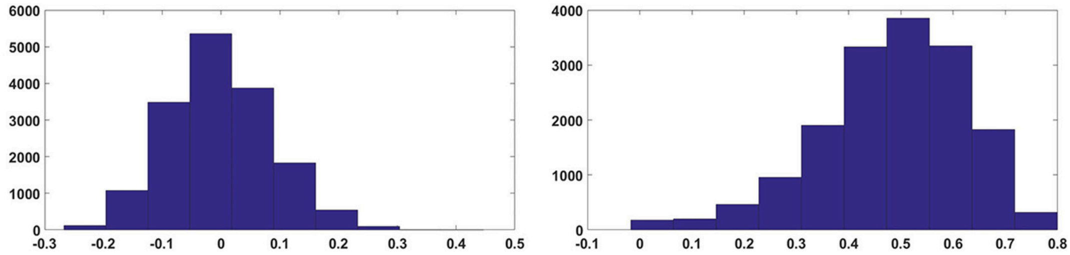


FIGURE 10 | Histogram of correlation coefficients of input weights.

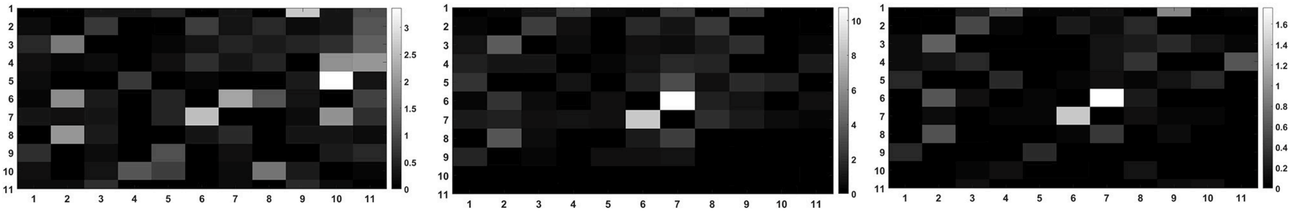


FIGURE 11 | Confusion matrices for different binning strategies exhibit peaks at different locations for time based and spike count based binning. Hence, a combination of these two methods can eliminate some of these errors.

binning and spike count based binning using several randomized training and testing sets. The resulting confusion matrices are plotted alongside the confusion matrix for the combined strategy in **Figure 11**. It can be clearly seen from the confusion matrices that while some of the peaks of the confusion matrices are at the same locations for both time based and spike count based methods, a significant number of minor peaks are at different locations. Therefore, a significant number of those misclassifications occurring for only one of the two binning methods are correctly classified in the combined strategy. This claim has been further quantitatively analyzed in Appendix.

5.2. Comparison With Other Methods

Next, we compare our results with reported accuracies in existing literature using the N-TIDIGITS18 dataset. For fixed bin size strategy, Neil and Liu (2016) obtained an accuracy of 87.65% using CNN and an accuracy of 82.82% using GRU RNN. Anumula et al. (2018) also obtained 88.6% accuracy using GRU RNN and 86.1% accuracy using LSTM RNN for the same feature extraction technique. For fixed number of bins strategy, Abdollahi and Liu (2011) obtained an accuracy of 95.08% using SVM. Thus, we can see that the accuracies reported in this paper outperform those obtained using fixed bin size or fixed number of bins techniques in existing literature. The best case accuracies obtained in this paper are comparable to that of MFCC based features in previous works [using MFCC based features, (Abdollahi and Liu, 2011) obtained an accuracy of 96.83% using SVM while (Anumula et al., 2018) obtained an accuracy of 97.90% using GRU RNN]. However, this comparison is imperfect since we need to account for the power needed in generating mode complex features like MFCC. Tsai et al. (2017) has shown that the power required for MFCC feature extraction is 122 mW on FPGA based implementation and 62.3 mW on ARM based implementation for TIDIGITS dataset using a 32 ms frame size. This is significantly higher than that of feature extraction techniques described in this paper (**Tables 2**, **3**). Also, it is difficult to compare power dissipation of RNN approaches since very few hardware implementations of these networks are reported. As one example, Gao et al. (2018) reports a Delta RNN network that uses ≈453K operations per frame of 25 ms (excluding FFT operations to generate features) which is quite comparable to the number of operations needed by the ELM first stage. However, it should be noted that the ELM first stage operations were simple random multiplications which could be easily implemented in low pwoer using analog techniques while the same cannot be said for the RNN.

5.3. Real-Time Detection of Word Occurrence

For the classification of the dataset so far we have assumed that the start and end of a digit is clearly marked for both training and testing data. But for real time applications, this assumption will not hold. So, we have decided to employ a sliding window technique for automatic detection of start and end of a digit. For the spike N-TIDIGITS18 dataset we have used, no noise was added to the waveforms of the original TIDIGITS dataset. So, the detection of start and end of the digit will become a relatively trivial task. However, the more challenging task is to detect the start and end of the signal in presence of noise. Therefore, we have implemented a threshold-based start and end detection using a sliding window assuming presence of noise. The algorithm detects the start of a digit if the total spike count within the window is higher than the given threshold and rejects the frame as noise if the total spike count is less than the threshold. Once the start of a digit is detected, the upcoming spikes are assumed to be part of the digit until the total spike count within a window is less than the threshold for a certain number of consecutive windows. At this point, the last window where the spike count was higher than the threshold is assumed to be the end of the digit. This ensures that the false end detection is avoided in case there are low spike count windows within the digit. We have set the threshold as a certain % of average spike count per window over all samples and the number of consecutive low spike count windows required to determine the end of a digit is a parameter dependent on the sliding window size.

We have tested this algorithm on best accuracy cases of both fixed number of bins strategy (time based binning, 10 bins/sample) and fixed bin size strategy (time based binning, bin size = 40 ms). We used a non-overlapping sliding window size of 40 ms and 2 consecutive windows with sub-threshold spike count for end detection. For fixed bin size strategy, the accuracy remained same for 10% threshold level and decreased by 0.8% for 20% threshold level. For fixed number of bins strategy, the reductions in accuracy were 2.5% and 3.6% respectively for 10% and 20% threshold level respectively. The diminished effect of start and end detection on the classification accuracy for fixed bin size strategy can be attributed to its indifference toward digit duration and thereby exact start and end time unlike its counterpart. Thus, the fixed bin size strategy seems relatively more noise robust.

In this proposed algorithm, the loss of accuracy stems from three sources, (a) loss of bins at the beginning, (b) loss of bins at the end and (c) loss of part of the digits due to false detection. For the fixed bin size case, only (c) is the major contributor to loss in accuracy while for fixed bin size case, all three factors contribute to the accuracy loss. Moreover, this sliding window technique introduces some additional latency depending upon the number of sub-threshold spike count windows used for end detection.

6. CONCLUSION

In this paper, we have presented several low-complexity feature extraction techniques to construct an end-to-end speech recognition system using a neuromorphic spiking cochlea and neuromorphic ELM IC. Moreover, the computational complexity, power requirement and memory requirement of the proposed techniques were calculated. Furthermore, we have used both software and hardware simulations of the neuromorphic ELM IC to obtain high classification accuracies (∼96%) for the N-TIDIGITS18 dataset.

The proposed fixed number of bins and fixed bin size methods presented a clear trade-off between classification accuracy and hardware overhead where using fixed number of bins gives ∼2-3 % higher accuracy with ∼ 3× more hardware overhead compared to the fixed bin size method. Our strategy of combining two different feature space representations of the input data gives high classification accuracy while using ∼ 25× less memory compared to the fixed number of bins method. So far, the feature extraction block of our proposed architecture is simulated in software only. In future, we plan to implement the feature extraction block using a microcontroller to produce a fully hardware based neuromorphic speech recognition system based on the low-power component prototypes Yang et al. (2016). Moreover, we plan to use our proposed architecture for other speech and audio recognition problems including speaker identification.

AUTHOR CONTRIBUTIONS

All the authors have contributed in varying degrees to different aspects of this paper. JA contributed in data analysis, software simulations, drafting and revising the manuscript. AP has contributed in experiment design and data collection using ELM IC. XL has contributed in experiment design and data acquisition using silicon cochlea. YC has contributed in data analysis and hardware data collection using ELM IC. S-CL has contributed in overall conception and design of the experiments and revising the manuscript. AB has also contributed in conception and design of the experiments and drafting and revising the manuscript.

REFERENCES

Abdollahi, M., and Liu, S.-C. (2011). "Speaker-independent isolated digit recognition using an aer silicon cochlea," in *2011 IEEE Biomedical Circuits and Systems Conference (BIOCAS)* (San Diego, CA: IEEE), 269–272.

Akusok, A., Björk, K.-M., Miche, Y., and Lendasse, A. (2015). High-performance extreme learning machines: a complete toolbox for big data applications. *IEEE Access* 3, 1011–1025. doi: 10.1109/ACCESS.2015.2450498

Anumula, J., Neil, D., Delbruck, T., and Liu, S.-C. (2018). Feature representations for neuromorphic audio spike streams. *Front. Neurosci.* 12:23. doi: 10.3389/fnins.2018.00023

Chakrabartty, S., and Liu, S.-C. (2010). "Exploiting spike-based dynamics in a silicon cochlea for speaker identification," in *2010 IEEE International Symposium on Circuits and Systems (ISCAS)* (Paris: IEEE), 513–516.

Chan, V., Liu, S.-C., and van Schaik, A. (2007). AER EAR: A matched silicon cochlea pair with address event representation interface. *IEEE Trans. Circ. Syst. I Regul. Pap.* 54, 48–59. doi: 10.1109/TCSI.2006.887979

Chen, Y., Yao, E., and Basu, A. (2016). A 128-channel extreme learning machine-based neural decoder for brain machine interfaces. *IEEE Trans. Biomed. Circ. Syst.* 10, 679–692. doi: 10.1109/TBCAS.2015.2483618

Deng, J., Fruhholz, S., Zhang, Z., and Schuller, B. (2017). Recognizing emotions from whispered speech based on acoustic feature transfer learning. *IEEE Access* 5, 5235–5246. doi: 10.1109/ACCESS.2017.2672722

Eliasmith, C. and Anderson, C. H. (2004). *Neural Engineering: Computation, Representation, and Dynamics in Neurobiological Systems*. MIT press.

Eliasmith, C., Stewart, T. C., Choo, X., Bekolay, T., DeWolf, T., Tang, Y., et al. (2012). A large-scale model of the functioning brain. *Science* 338, 1202–1205. doi: 10.1126/science.1225266

Gao, C., Neil, D., Ceolini, E., Liu, S.-C., and Delbruck, T. (2018). "Deltarnn: a power-efficient rnn accelerator," in *Twenty-Sixth ACM/SIGDA International Symposium on Field-Programmable Gate Arrays (FPGA)* (Monterey, CA: ACM).

Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A.-R., Jaitly, N., et al. (2012). Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. *IEEE Signal Process. Mag.* 29, 82–97. doi: 10.1109/MSP.2012.2205597

Hoerl, A. E., and Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics* 12, 55–67.

Huang, G.-B., Zhou, H., Ding, X., and Zhang, R. (2012). Extreme learning machine for regression and multiclass classification. *IEEE Trans. Syst. Man Cybern. Part B (Cybern)*, 42, 513–529. doi: 10.1109/TSMCB.2011.2168604

Huang, G.-B., Zhu, Q.-Y., and Siew, C.-K. (2006). Extreme learning machine: theory and applications. *Neurocomputing* 70, 489–501. doi: 10.1016/j.neucom.2005.12.126

Leonard, R. (1984). "A database for speaker-independent digit recognition," in *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP '84*, Vol. 9 (San Diego, CA), 328–331.

Li, C.-H., Delbruck, T., and Liu, S.-C. (2012). "Real-time speaker identification using the AEREAR2 event-based silicon cochlea," in *2012 IEEE International Symposium on Circuits and Systems (ISCAS)* (Seoul: IEEE) , 1159–1162.

Liu, S.-C., and Delbruck, T. (2010). Neuromorphic sensory systems. *Curr. Opin. Neurobiol.* 20, 288–295. doi: 10.1016/j.conb.2010.03.007

Liu, S.-C., van Schaik, A., Minch, B. A., and Delbruck, T. (2014). Asynchronous binaural spatial audition sensor with 2 × 64 × 4 channel output. *IEEE Trans. Biomed. Circ. Syst.* 8, 453–464. doi: 10.1109/TBCAS.2013.2281834

Mead, C. (1990). Neuromorphic electronic systems. *Proc. IEEE* 78, 1629–1636.

Moeys, D. P., Corradi, F., Kerr, E., Vance, P., Das, G., Neil, D., et al. (2016). "Steering a predator robot using a mixed frame/event-driven convolutional neural network," in *2016 Second International Conference on Event-based Control, Communication, and Signal Processing (EBCCSP)* (Krakow: IEEE), 1–8.

Neil, D., and Liu, S.-C. (2016). "Effective sensor fusion with event-based sensors and deep network architectures," in *Circuits and Systems (ISCAS), 2016 IEEE International Symposium on* (Montréal, QC: IEEE), 2282–2285.

Patil, A., Shen, S., Yao, E., and Basu, A. (2015). "Random projection for spike sorting: decoding neural signals the neural network way," in *2015 IEEE Biomedical Circuits and Systems Conference (BioCAS)* (Atlanta, GA: IEEE), 1–4.

Penrose, R. (1955). "A generalized inverse for matrices," in *Mathematical Proceedings of the Cambridge Philosophical Society*, Vol. 51 (Cambridge, UK: Cambridge University Press), 406–413.

Song, Y., Crowcroft, J., and Zhang, J. (2012). Automatic epileptic seizure detection in EEGs based on optimized sample entropy and extreme learning machine. *J. Neurosci. Methods* 210, 132–146. doi: 10.1016/j.jneumeth.2012.07.003

Stewart, T., Choo, F.-X., and Eliasmith, C. (2012). "Spaun: A perception-cognition-action model using spiking neurons," in *Proceedings of the Cognitive Science Society*, Vol. 34 (Sapporo).

Stewart, T. C. (2012). *A Technical Overview of the Neural Engineering Framework*. University of Waterloo.

Tsai, W.-Y., Barch, D. R., Cassidy, A. S., DeBole, M. V., Andreopoulos, A., Jackson, B. L., et al. (2017). Always-on speech recognition using truenorth, a reconfigurable, neurosynaptic processor. *IEEE Trans. Comput.* 66, 996–1007. doi: 10.1109/TC.2016.2630683

Yang, M., Chien, C.-H., Delbruck, T., and Liu, S.-C. (2016). A 0.5V 55μw 64×2 channel binaural silicon cochlea for event-driven stereo-audio sensing. *IEEE J. Solid State Circ.* 51, 2554–2569. doi: 10.1109/JSSC.2016.2604285

Yao, E., and Basu, A. (2017). "VLSI Extreme Learning Machine: A design space exploration," in *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 25, 60–74.

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

APPENDIX

Further Discussion on Confusion Matrices

To quantitatively analyze our hypothesis in section 5 that the correlation matrices produced by time based binning and spike count based binning have peaks at different locations, we have used correlation coefficients. We have calculated the correlation coefficients between confusion matrices produced by time (and spike count) based binning for different randomizedtraining and testing sets. We have also obtained the cross-correlation coefficients between confusion matrices produced by time and spike count based binning for same training and testing sets. The spread of the correlation coefficients obtained is shown using the box-plots in **Figure A1**. It is quite evident from the box-plots that confusion matrices produced by the same feature extraction method for different training and testing sets are highly correlated while confusion matrices produced by different feature extraction methods for same training and testing set have lower correlation.

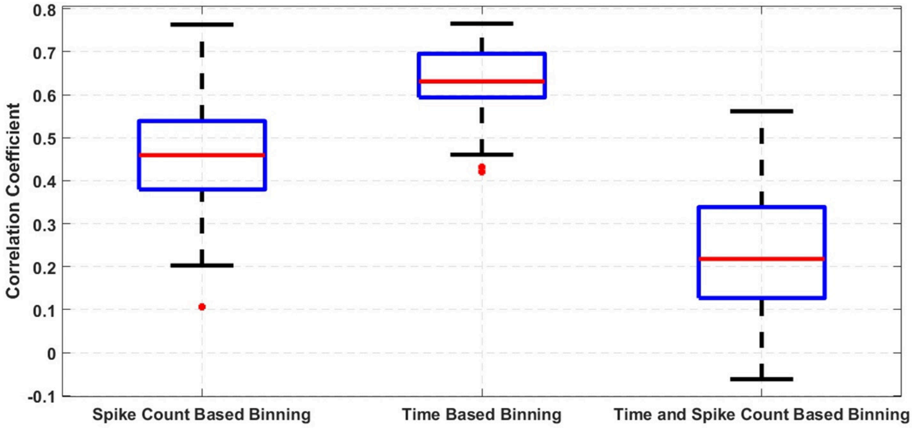


FIGURE A1 | Correlation between confusion matrices.

ORIGINAL RESEARCH
published: 19 November 2018
doi: 10.3389/fnins.2018.00836

A Spiking Neural Network Framework for Robust Sound Classification

Jibin Wu[1], *Yansong Chua*[2]*, *Malu Zhang*[1], *Haizhou Li*[1,2] *and Kay Chen Tan*[3]

[1] *Department of Electrical and Computer Engineering, National University of Singapore, Singapore, Singapore,* [2] *Institute for Infocomm Research, A*STAR, Singapore, Singapore,* [3] *Department of Computer Science, City University of Hong Kong, Kowloon Tong, Hong Kong*

Edited by:
Shih-Chii Liu,
ETH Zürich, Switzerland

Reviewed by:
Daniel Neil,
University of Zurich, Switzerland
Charles Augustine,
Intel, United States

***Correspondence:**
Yansong Chua
chuays@i2r.a-star.edu.sg

Specialty section:
This article was submitted to Neuromorphic Engineering, a section of the journal Frontiers in Neuroscience

Received: *29 March 2018*
Accepted: *26 October 2018*
Published: *19 November 2018*

Citation:
Wu J, Chua Y, Zhang M, Li H and Tan KC (2018) A Spiking Neural Network Framework for Robust Sound Classification. Front. Neurosci. 12:836. doi: 10.3389/fnins.2018.00836

Environmental sounds form part of our daily life. With the advancement of deep learning models and the abundance of training data, the performance of automatic sound classification (ASC) systems has improved significantly in recent years. However, the high computational cost, hence high power consumption, remains a major hurdle for large-scale implementation of ASC systems on mobile and wearable devices. Motivated by the observations that humans are highly effective and consume little power whilst analyzing complex audio scenes, we propose a biologically plausible ASC framework, namely SOM-SNN. This framework uses the unsupervised self-organizing map (SOM) for representing frequency contents embedded within the acoustic signals, followed by an event-based spiking neural network (SNN) for spatiotemporal spiking pattern classification. We report experimental results on the RWCP environmental sound and TIDIGITS spoken digits datasets, which demonstrate competitive classification accuracies over other deep learning and SNN-based models. The SOM-SNN framework is also shown to be highly robust to corrupting noise after multi-condition training, whereby the model is trained with noise-corrupted sound samples. Moreover, we discover the early decision making capability of the proposed framework: an accurate classification can be made with an only partial presentation of the input.

Keywords: spiking neural network, self-organizing map, automatic sound classification, maximum-margin Tempotron classifier, noise robust multi-condition training

1. INTRODUCTION

Automatic sound classification generally refers to the automatic identification of ambient sounds in the environment. Environmental sounds, complementary to visual cues, informs us of our surrounding environment and is an essential part of our daily life. ASC technologies enable a wide range of applications including, but not limited to content-based sound classification and retrieval (Guo and Li, 2003), audio surveillance (Rabaoui et al., 2008), sound event classification (Dennis et al., 2011) and disease diagnosis (Kwak and Kwon, 2012).

The conventional ASC systems are inspired by automatic speech recognition systems, which typically comprise of acoustic signal pre-processing, feature extraction and classification (Sharan and Moir, 2016). As shown in **Figure 1**, signal pre-processing can be further sub-categorized into pre-emphasis (high-frequency components are amplified), segmenting (continuous acoustic signals are segmented into overlapping short frames), and windowing (a window function is applied

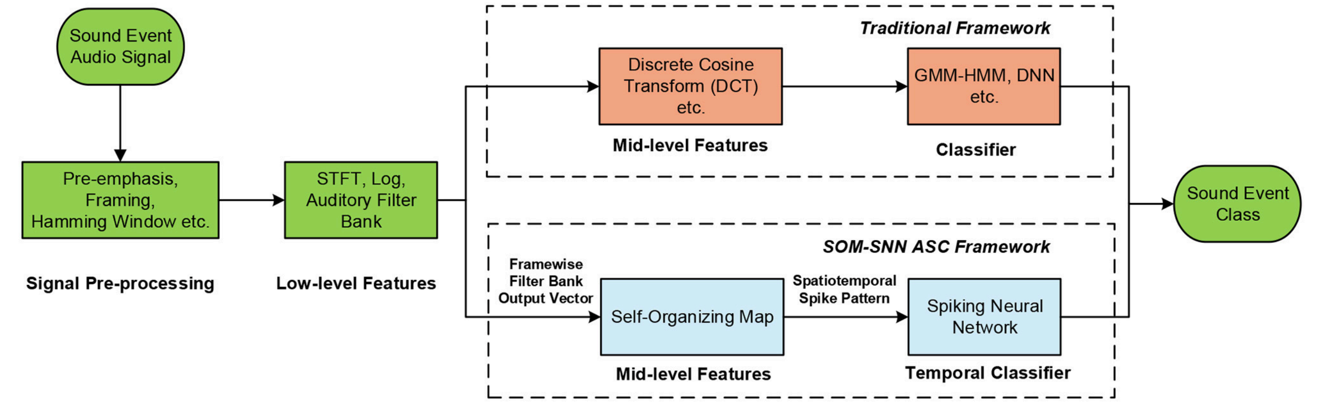


FIGURE 1 | Overview of the proposed SOM-SNN ASC framework, which uses the SOM as a mid-level feature representation of frequency contents in the sound frames, and classifies the spatiotemporal spike patterns using SNNs.

to reduce the effect of spectral leakage). Several feature representations for acoustic signals have been proposed over the years for capturing frequency contents and temporal structures of acoustic signals (Mitrović et al., 2010). The most frequently used features are the Mel-Frequency Cepstral Coefficients (MFCC) (Chu et al., 2009) and Gammatone Cepstral Coefficients (GTCC) (Leng et al., 2012). Both these features mimic the human auditory system, as they are more sensitive to changes in the low-frequency components. These frame-based features are then used to train a GMM-HMM or deep learning models in a classification task.

Despite the significant performance improvement in recent years driven by deep learning models and the abundance of training data, two major challenges remain to prevent the large-scale adoption of such frame-based ASC systems on mobile and wearable devices. First of all, high-performance computing, which typically entails high power consumption, is commonly unavailable on such devices. Secondly, the performance of state-of-the-art GMM-HMM and deep learning models, with MFCC or GTCC feature as input, degrades significantly with increased background noise.

We note that in comparison to existing machine learning techniques, human performs much more efficiently and robustly in various auditory perception tasks, whereby different frequency components of the acoustic signal are asynchronously encoded using sparse and highly parallel spiking impulses. Remarkably, even though spiking impulses in biological neural systems are transmitted at rates of several orders of magnitude slower than signals in modern transistors, humans perceive complex audio scenes with much lower energy consumption (Merolla et al., 2014). Moreover, human learn to distinguish sounds with only sparse supervision, currently formulated as zero-shot or one-shot learning (Fei-Fei et al., 2006; Palatucci et al., 2009) in machine learning. These observations of human auditory perception motivate us to explore and design a biologically plausible event-based ASC system.

Event-based computation, as observed in the human brain and nervous systems, relies on asynchronous and highly parallel spiking events to efficiently encode and transmit information. In contrast to traditional frame-based machine vision and auditory systems, event-based biological systems represent and process information in a much more energy efficient manner whereby energy is only consumed during spike generation and transmission. Spiking neural network (SNN) is one such class of neural networks motivated by event-based computation. For training the SNN on a temporal pattern classification task, many temporal learning rules have been proposed. Depending on how the error function is formulated, they can be categorized into either spike-time based (Ponulak and Kasiński, 2010; Yu et al., 2013a) or membrane-potential based (Gütig and Sompolinsky, 2006; Gütig, 2016; Zhang et al., 2017). For spike-time based learning rules, the main objective is to minimize the time difference between the actual and desired output spike patterns by updating the synaptic weights. In contrast, membrane potential based learning rules use the voltage difference between the actual membrane potential and the firing threshold to guide synaptic weight updates.

Recently, there are growing interests in integrating event-based sensors, such as the DVS (Delbrück et al., 2010), DAVIS (Brandli et al., 2014) and DAS (Liu et al., 2014), with event-based neuromorphic processors such as TrueNorth (Merolla et al., 2014) and SpiNNaker (Furber et al., 2013) for more energy efficient applications (Serrano-Gotarredona et al., 2015; Amir et al., 2017).

In this work, we propose a novel SNN framework for automatic sound classification. We adopt a biologically plausible auditory front-end (using logarithmic mel-scaled filter banks that resemble the functionality of the human cochlea) to first extract low-level spectral features. After which, the unsupervised self-organizing map (SOM) (Kohonen, 1998) is used to generate an effective and sparse mid-level feature representation. The best-matching units (BMUs) of the SOM are activated over time and the corresponding spatiotemporal spike patterns are generated, which represent the characteristics of each sound event. Finally, a newly developed Maximum-Margin Tempotron temporal learning rule (membrane-potential based) is used to classify the spike patterns into different sound categories.

This paper furthers our recent research, which focused on speech recognition (Wu et al., 2018a). In this work, we look into the SOM-SNN properties, system architecture and its robustness against noise in a sound event classification task. We also perform a comparative study with the state-of-the-art deep learning techniques. The main contributions of this work are threefold:

- We propose a biologically plausible event-based ASC framework, namely the SOM-SNN. In this framework, the unsupervised SOM is utilized to represent the frequency contents of environmental sounds, while the SNN learns to distinguish these sounds. This framework achieves competitive classification accuracies compared with deep learning and other SNN-based models on the RWCP and TIDIGITS datasets. Additionally, the proposed framework is shown to be highly robust to corrupting noise after multi-condition training (McLoughlin et al., 2015), whereby the model is trained with noise-corrupted sound samples.
- We propose a new Maximum-Margin Tempotron temporal learning rule, which incorporates the Tempotron (Gütig and Sompolinsky, 2006) with the maximum-margin classifier (Cortes and Vapnik, 1995). This newly introduced hard margin ensures a better separation between positive and negative classes, thereby improving the classification accuracy of the SNN classifier.
- We discover the early decision making capability of the proposed SNN-based classifier, which arises naturally from the Maximum-Margin Tempotron learning rule. The earliest possible discriminative spatiotemporal feature is identified automatically in the SNN classifier, and an output spike is immediately triggered by the correct output neuron. Consequently, an input pattern could be classified with high accuracy when only part of it is presented. Under the same test conditions, the SNN-based classifier consistently outperforms other traditional artificial neural networks (ANNs), [i.e., the Recurrent Neural Network (RNN) (Graves et al., 2013) and Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997)] in a temporal pattern classification task. It, therefore, shows great potential for real-world applications, whereby acoustic signals maybe intermittently distorted by noise: the classification decision can be robustly made based on the input portion with less distortion.

2. METHODS

In this section, we first describe the components of the proposed SOM-SNN framework. Next, we present the experiments designed to evaluate the classification performance and noise robustness of the proposed framework. Finally, we compare it with other state-of-the-art ANN- and SNN-based models.

2.1. Auditory Front-end

Human auditory front-end consists of the outer, middle and inner ear. In the outer ear, sound waves travel through air and arrive at the pinna, which also embeds the location information of the sound source. From the pinna, the sound signals are then transmitted via the ear canal, which functions as a resonator, to the middle ear. In the middle ear, vibrations (induced by the sound signals) are converted into mechanical movements of the ossicles (i.e., malleus, incus, and stapes) through the tympanic membrane. The tensor tympani and stapedius muscles, which are connected to the ossicles, act as an automatic gain controller to moderate mechanical movements under the high-intensity scenario. At the end of the middle ear, the ossicles join with the cochlea via the oval window, where mechanical movements of the ossicles are transformed into fluid pressure oscillations which move along the basilar membrane in the cochlea (Bear et al., 2016).

The cochlea is a wonderful anatomical work of art. It functions as a spectrum analyzer which displaces the basilar membrane at specific locations that correspond to different frequency components in the sound wave. Finally, displacements of the basilar membrane activate inner hair cells via nearby mechanically gated ion channels, converting mechanical displacements into electrical impulse trains. The spike trains generated at the hair cells are transmitted to the cochlear nuclei through dedicated auditory nerves. Functionally, the cochlear nuclei act as filter banks, which also normalize activities of saturated auditory nerve fibers over different frequency bands. Most of the auditory nerves terminate at the cochlear nuclei where sound information is still identifiable. Beyond the cochlear nuclei, in the auditory cortex, it remains unclear how information is being represented and processed (Møller, 2012).

The understanding of the human auditory front-end has a significant impact on machine hearing research and inspires many biologically plausible feature representations of acoustic signals, such as the MFCC and GTCC. In this paper, we adopt the MFCC representation. As shown in **Figure 2**, we pre-processed the sound signals by first applying pre-emphasis to amplify high-frequency contents, then segmenting the continuous sound signals into overlapped frames of suitable length so as to better capture the temporal variations of the sound signal, and finally applying the Hamming window on these frames to reduce the effect of spectral leakage. To extract the spectral contents in the acoustic stimuli, we perform Short-Time Fourier-Transform (STFT) on the sound frames and compute the power spectrum. After that, we apply 20 logarithmic mel-scaled filters on the resulting power spectrum, generating a compressed feature representation for each sound frame. The mel-scaled filter banks emulate the human perception of sound that is more discriminative toward the low frequency as compared to the high frequency components.

2.2. Feature Representation Using SOM

Feature representation is critical in all ASC systems; state-of-the-art ASC systems input low-level MFCC or GTCC features into the GMM-HMM or deep learning models so as to extract higher-level representations. In our initial experiments, we observe that existing SNN temporal learning rules cannot discriminate latency (Yu et al., 2013b) or population (Bohte et al., 2002) encoded mel-scaled filter bank outputs effectively. Therefore, we propose to use the biologically inspired SOM to form a mid-level feature representation of the sound frames. The neurons in the SOM form distinctive synaptic filters that organize themselves tonotopically and compete to

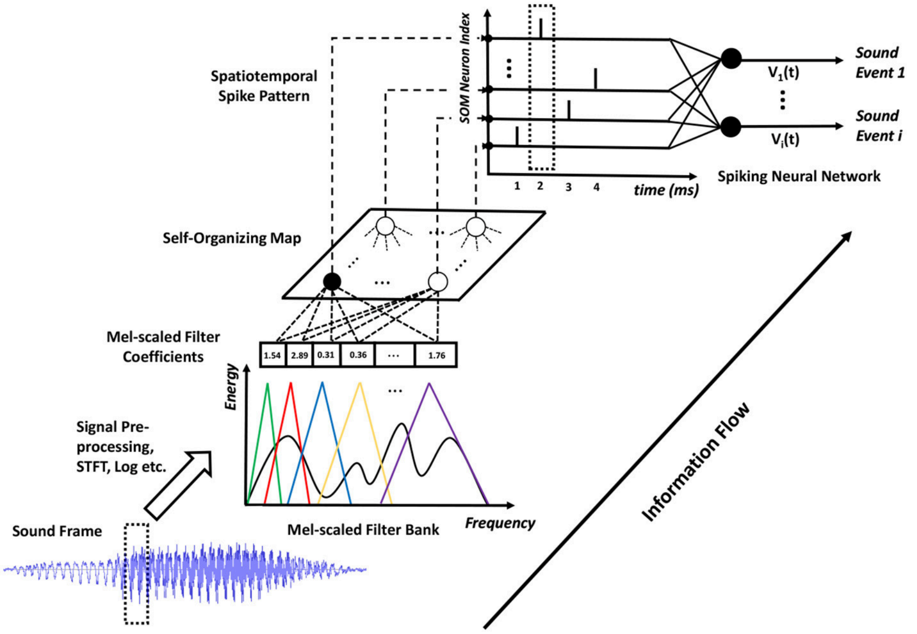


FIGURE 2 | The details of the proposed SOM-SNN ASC framework. The sound frames are pre-processed and analyzed using mel-scaled filter banks. Then, the SOM generates discrete BMU activation sequences which are further converted into spike trains. All such spike trains form a spatiotemporal spike pattern to be classified by the SNN.

represent the filter bank output vectors. Such tonotopically organized feature maps have been found in the human auditory cortex in many physiological experiments (Pantev et al., 1995).

As shown in **Figure 2**, all neurons in the SOM are fully connected to the filter bank and receive mel-scaled filter outputs (real-valued vectors). The SOM learns acoustic features in an unsupervised manner, whereby two mechanisms: competition and cooperation, guide the formation of a tonotopically organized neural map. During training, the neurons in the SOM compete with each other to best represent the input frame. The best-matching unit (BMU), with its synaptic weight vector closest to the input vector in the feature space, will update its weight vector to become closer to the input vector. Additionally, the neurons surrounding the BMU will cooperate with it by updating their weight vectors to move closer to the input vector. The magnitude of the weight update of neighboring neurons is inversely proportional to its distance to the BMU, effectively facilitating the formation of neural clusters. Eventually, the synaptic weight vectors of neurons in the SOM follow the distribution of input feature vectors and organize tonotopically, such that adjacent neurons in the SOM will have similar weight vectors.

During the evaluation, as shown in **Figure 2**, the SOM (through the BMU neuron) emits a single spike at each sound frame sampling interval. The sparsely activated BMUs encourage pattern separation and enhance power efficiency. The spikes triggered over the duration of a sound event form a spatiotemporal spike pattern, which is then classified by the SNN into one of the sound classes. The mechanisms of SOM training and testing are provided in **Algorithm 1** (see more details Kohonen, 1998). This classical work (Kohonen, 1998) trained the SOM for a phoneme recognition task, which then used a set of hand-crafted rules to link sound clusters of the SOM to actual phoneme classes. In this work, we use an SNN-based classifier to automatically categorize the spatiotemporal spike patterns into different sound events.

2.3. Supervised Temporal Classification

2.3.1. Neuron Model

For the SNN-based temporal classifier, we adopt the leaky integrate-and-fire neuron model (Gütig and Sompolinsky, 2006), which utilizes the kernel function to describe the effect of pre-synaptic spikes on the membrane potential of post-synaptic neurons. When there is no incoming spike, the post-synaptic neuron i remains at its resting potential V_{rest}. Each incoming spike from the pre-synaptic neuron j at t_j will induce a post-synaptic potential (PSP) on the post-synaptic neuron as described by the following kernel function:

$$K(t-t_j) = K_0\left(\exp(-\frac{t-t_j}{\tau_m}) - \exp(-\frac{t-t_j}{\tau_s})\right)\theta(t-t_j) \quad (7)$$

where K_0 is a normalization factor that ensures the maximum value of the kernel $K(t-t_j)$ is 1. τ_m and τ_s correspond to the membrane and synaptic time constants, which jointly determine the shape of the kernel function. In addition, $\theta(t-t_j)$ represents

Algorithm 1: The Self-Organizing Map Algorithm

```
Input:
The randomly initialized weight vector $w_i(0)$ for neuron i = 1, ..., $M \cdot N$, where $M$ and $N$ are the length and width of the SOM
The training set that is formed by framewise filter bank output vectors
The initial width of the neighborhood function $\sigma(0) = \sqrt{M^2 + N^2}/2$
The number of training epochs $E$, initial learning rate $\eta_0$ and time constant of the time-varying width $\tau_1 = E/\log[\sigma(0)]$
Output:
The final weight vectors $w_i(E)$ for neuron $i$ = 1, ..., $M \cdot N$
Train:
for $e \in [0, 1, 2, ..., E-1]$ do
    1. Randomly choose an input vector $x_{train} = [x_1, x_2, x_3, \ldots, x_n]$ from the training set, where $n$ is the total number of mel-scaled filters
    2. Determine the winner neuron $k$ that has a weight vector closest to the current input vector $x_{train}$:
        $k = \arg\min_i ||w_i(e) - x_{train}||$    (1)
    3. Update the learning rate $\eta(e)$, the time-varying width $\sigma(e)$ and the Gaussian neighborhood function $h_{i,k}(e)$ for all neurons $i = 1, ..., m$:
        $\eta(e) = \eta_0 \cdot \exp(-e/E)$    (2)
        $\sigma(e) = \sigma(0) \cdot \exp(-e/\tau_1)$    (3)
        $h_{i,k}(e) = \exp\{-||w_i(e) - w_k(e)||^2/[2 \cdot \sigma(e)^2]\}$    (4)
    4. Update $w_i(e+1)$ for all neurons $i = 1, ..., M \cdot N$:
        $w_i(e+1) = w_i(e) + \eta(e) \cdot h_{i,k}(e) \cdot [x_{train} - w_i(e)]$    (5)
Test:
Given any input vector $x_{test}$ from the testing set, label it with the winner neuron $k$ that has weight vector closest to $x_{test}$:
    $k = \arg\min_i ||w_i(E) - x_{test}||$    (6)
```

the Heaviside function to ensure that only pre-synaptic spikes emitted before time t are considered.

$$\theta(x) = \begin{cases} 1, & \textit{if } x \geq 0 \\ 0, & \textit{otherwise} \end{cases} \tag{8}$$

At time t, the membrane potential of the post-synaptic neuron i is determined by the weighted sum of all PSPs triggered by incoming spikes before time t:

$$V_i(t) = \sum_j w_{ji} \sum_{t_j < t} K(t - t_j) + V_{rest} \quad \forall t \in [0, T] \tag{9}$$

where w_{ji} is the synaptic weight between the pre-synaptic neuron j and post-synaptic neuron i, and T is the duration of the simulation. Whenever the membrane potential $V_i(t)$ of the post-synaptic neuron i reaches the firing threshold, it emits a spike. For the single-spike based classifier used in this work, the membrane potential of the post-synaptic neuron then smoothly relaxes back to V_{rest} after spiking by shunting all subsequent input spikes (i.e., input spikes arriving after the post-synaptic spike, have no effect on the membrane potential of the post-synaptic neuron). Since these input spikes would not contribute to any learning in the single-spike based classifier, the unnecessary post-spike computations can be safely ignored.

2.3.2. Maximum-Margin Tempotron Learning Rule

For the classification of spatiotemporal patterns as illustrated by the SNN in **Figure 2**, we use a modified version of the biologically plausible Tempotron (Gütig and Sompolinsky, 2006) learning rule to train the classifier, which has been successfully used in several ASC tasks (Dennis et al., 2013; Xiao et al., 2017). The original Tempotron rule is designed for a binary classification task, such that a neuron emits a spike when it observes a spike pattern from its desired class, and remains quiescent otherwise. For a multi-class classification task, we adopt the one-against-all strategy to train one output neuron to respond to each class.

During training, for neuron i that represents the ith class, we treat all training samples with class label i as positive samples, and all others as negative. During testing, we monitor the membrane potential of all output neurons and classify the test sample as follows: (1) If no output neuron fires over the sample duration, we select the output neuron with the highest membrane potential as the correct class. (2) If only a single output neuron fires, the class label corresponding to this neuron is selected. (3) Otherwise, if two or more neurons fire, we label the test sample with the earliest firing neuron, which signals the detection of the earliest local discriminative feature (a property of the Tempotron).

The Tempotron learning rule follows a stochastic gradient descent method for synaptic weight updates: the desired output neuron triggers a weight update whenever it fails to fire on samples with matching class label or when the wrong output neurons fire erroneously on samples from other classes. When the desired output neuron i fails to fire, long-term potentiation (LTP) update with cost function V_{thr} - $V_{t_i^{\max}}$ is triggered. Similarly, long-term depression (LTD) update with cost function $V_{t_i^{\max}}$ - V_{thr} is triggered when the wrong output neuron fires erroneously. The Tempotron update rule is defined as follows:

$$\Delta w_{ij} = \begin{cases} \lambda \sum\limits_{t_j^{(f)} < t_i^{\max}} K(t_i^{\max} - t_j^{(f)}), & \textit{if LTP} \\ -\lambda \sum\limits_{t_j^{(f)} < t_i^{\max}} K(t_i^{\max} - t_j^{(f)}), & \textit{if LTD} \\ 0, & \textit{otherwise} \end{cases} \tag{10}$$

where λ denotes a constant learning rate and $t_i^{\max}$ refers to the time instant when the postsynaptic neuron i reaches its maximum membrane potential over the pattern duration. The $t_j^{(f)}$ are spike times of spike emitted by the pre-synaptic neuron j. The synaptic weights are only updated at the time instant of $t_i^{\max}$. For LTD weight update, the $t_i^{\max}$ is also the spike time since the post-spike computations are ignored.

Inspired by the maximum-margin classifier (Cortes and Vapnik, 1995), we introduce a hard margin Δ to the V_{thr} and denote the new learning rule as the Maximum-Margin Tempotron. During the training phase, the Δ term is either added to or deducted from the V_{thr} of the desired or wrong output neurons, respectively. Consequently, for the desired neuron i, a spike is generated at t if

$$V_i(t) = V_{thr} + \Delta \quad and \quad \frac{d}{dt}V_i(t) > 0 \tag{11}$$

For the other (wrong) neurons, a spike is generated if

$$V_i(t) = V_{thr} - \Delta \quad and \quad \frac{d}{dt}V_i(t) > 0 \tag{12}$$

The desired output neuron will fire only when it has observed strong evidence that causes its $V_{t_{max}}$ to rise above V_{thr} by a margin of Δ. Similarly, the other neurons will be discouraged to fire and maintain its membrane potential by a margin Δ below V_{thr}. This additional margin Δ imposes a harder constraint during training and encourages the SNN classifier to find more discriminative features in the input spike patterns. Therefore, during testing, when the hard margin Δ is removed from $V_i(t)$ as described in Equation (9), the neurons are encouraged to respond with the desired spiking activities. This strategy helps to prevent overfitting and improves classification accuracy.

2.4. Multi-condition Training

Although state-of-the-art deep learning based ASC models perform reasonably well under the noise-free condition, it remains a challenging task for these models to recognize sound robustly in noisy real-world environments. To address this challenge, we investigated training the proposed SOM-SNN model with both clean and noisy sound data, as per the multi-condition training strategy.

The motivation for such an approach is that with training samples collected from different noisy backgrounds, the trained model will be encouraged to identify the most discriminative features and become more robust to noise. This methodology has been proven to be effective for Deep Neural Network (DNN) and SVM models under the high noise condition, with some trade-off in performance for clean sound data (McLoughlin et al., 2015). Here, we investigate its generalizability to SNN-based temporal classifiers under noisy environments.

2.5. Training and Evaluation

Here, we first introduce two standard benchmark datasets used to evaluate the classification accuracies of the proposed SOM-SNN framework, which are made up of environmental sounds and human speech. After which, we describe the experiments conducted on the RWCP dataset to evaluate model performance pertaining to the effectiveness of feature representation using the SOM, early decision making capability and noise robustness of the classifier.

2.5.1. Evaluation Datasets

The Real World Computing Partnership (RWCP) (Nishiura and Nakamura, 2002) sound scene dataset was recorded in a real acoustic environment at a sampling rate of 16 kHz. For a fair comparison with other SNN-based systems (Dennis et al., 2013; Xiao et al., 2017), we used the same 10 sound event classes from the dataset: “cymbals,” “horn,” “phone4,” “bells5,” “kara,” “bottle1,” “buzzer,” “metal15,” “whistle1,” “ring.” The sound clips were recorded as isolated samples with duration of 0.5s to 3s at high SNR. There are also short lead-in and lead-out silent intervals in the sound clips. We randomly selected 40 sound clips from each class, of which 20 are used for training and the remaining 20 for testing, giving a total of 200 training and 200 testing samples.

The TIDIGITS (Leonard and Doddington, 1993) dataset consists of reading digit strings of varying lengths, and the speech signals are sampled at 20 kHz. The TIDIGITS dataset is a publicly available dataset from the Linguistic Data Consortium, which is one of the most commonly available speech datasets used for benchmarking speech recognition algorithms. This dataset consists of spoken digit utterances from 111 male and 114 female speakers. We used all of the 12,373 continuous spoken digit utterances for the SOM training and the rest of the 4,950 isolated spoken digit utterances for the SNN training and testing. Each speaker contributes two isolated spoken digit utterances for all 11 classes (i.e., “zeros” to “nine” and “oh”). We split the isolated spoken digit utterances randomly with 3,950 utterances for training and the remaining 1,000 utterances for testing.

2.5.2. SOM-SNN Framework

The SOM-SNN framework, as shown in **Figure 2**, consists of three processing stages organized in a pipeline. These stages are trained separately and then evaluated in a single, continuous process. For the auditory front-end, we segment the continuous sound samples into frames of 100 ms length with 50 ms overlap between neighboring frames for the RWCP dataset. In contrast, we use a frame length of 25 ms with 10 ms overlap for the TIDIGITS dataset. These values are determined empirically to sufficiently discriminate the signals without excessive computational load. We utilize 20 mel-scaled filters for the spectral analysis, ranging from 200 to 8,000 Hz and 200 to 10,000 Hz respectively for the RWCP and TIDIGITS datasets.

The number of filters is again empirically determined, such that more filters do not improve classification accuracy.

For feature representation learning in the SOM, we utilize the SOM available in the MATLAB Neural Network Toolbox. The Euclidean distance is used to determine the BMUs, which are subsequently converted into spatiotemporal spike patterns. The output spikes from the SOM are generated per sound frame, with an interval as determined by the frame shift (i.e., 50 ms for RWCP dataset and 15 ms for TIDIGITS dataset). We study the effect of different hyperparameters including SOM map size, number of training epochs and number of activated neurons per incoming frame. Their effects on classification accuracy are presented in section 3.3.

We initialize the SNN by setting the threshold V_{thr}, the hard margin Δ and learning rate λ to 1.0, 0.5 and 0.005 respectively. The time constants of the SNN have determined empirically such that the PSP duration is optimal for the particular dataset, and we set τ_m to 750, 225 ms and τ_s to 187.5, 56.25 ms for the RWCP and TIDIGITS datasets, respectively. We train all the SNNs for 10 epochs by when convergence is observed. The initial weights for the neurons in the SNN classifier are drawn randomly from the Gaussian distribution with a mean of 0 and standard deviation of 10^{-3}. Parameters used in all our experiments are as above unless otherwise stated.

2.5.3. Traditional Artificial Neural Networks

To facilitate comparison with other traditional ANN models trained on the RWCP dataset, we implement four common neural network architectures, namely the Multi-Layer Perceptron (MLP) (Morgan and Bourlard, 1990), the Convolutional Neural Network (CNN) (Krizhevsky et al., 2012), the Recurrent Neural Network (RNN) (Graves et al., 2013) and the Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) using the Pytorch library. For a fair comparison, we implement the MLP with 1 hidden layer of 500 ReLU units, and the CNN with two convolution layers of 128 feature maps each followed by 2 fully-connected layers of 500 and 10 ReLU units. The input frames to the MLP and CNN are concatenated over time into a spectrogram image. Since the number of frames for each sound clip varies from 20 to 100 and cannot be processed directly by the MLP or CNN, we bilinearly rescale these spectrogram images into a consistent dimension of 20 × 64.

We implement both the RNN and LSTM with two hidden layers containing 100 hidden units each, and a dropout layer with a probability of 0.5 is applied after the first hidden layer to prevent overfitting. The input to the RNN and LSTM are the 20-dimensional filter bank output vectors. The weights for all networks are initialized with orthogonal conditions as suggested in (Saxe et al., 2013). The deep learning networks are trained with the cross-entropy criterion and optimized using the Adam (Kingma and Ba, 2014) optimizer. The learning rate is decayed to 99% of the original value after every epoch, and all networks are trained for 100 epochs, except for the CNN (50 epochs), by when convergence is observed. Simulations are repeated 10 times for each model, with random weight initialization.

To study the synergy between SOM and deep learning models (i.e., RNN and LSTM), we use the mid-level features of the SOM as inputs to train the RNN and LSTM, respectively denoted as SOM-RNN and SOM-LSTM. These features are obtained by converting the BMU that corresponds to each sound frame into a one-hot vector and concatenating them over time to form a sparse representation of each sound clip. We trained the SOM-RNN and SOM-LSTM models with the same set-up as the RNN and LSTM mentioned above.

2.5.4. Noise Robustness Evaluation

2.5.4.1. Environmental noise

We generate noise-corrupted sound samples by adding "Speech Babble" background noise from the NOISEX-92 dataset (Varga and Steeneken, 1993) to the clean RWCP sound samples. This selected background noise represents a non-stationary noisy environment with predominantly low-frequency contents, hence making a fair comparison with the noise robustness tests performed in LSF-SNN (Dennis et al., 2013) and LTF-SNN models (Xiao et al., 2017). For each training or testing sound sample, a random noise segment of the same duration is selected from the noise file and added at 4 different SNR levels of 20, 10, 0 and -5 dB separately, giving a total of 1,000 training and 1,000 testing samples. The SNR ratio is calculated based on the energy level of each sound sample and the corresponding noise segment in our experiments. Training is performed over the whole training set, while the testing set is evaluated separately at different SNR levels.

We perform multi-condition training on all the MLP, CNN, RNN, LSTM and SOM-SNN models. Additionally, we also conduct experiments whereby the models are trained with clean sound samples but tested with noise-corrupted samples (the mismatched condition).

2.5.4.2. Neuronal Noise

We also consider the effect of neuronal noise which is known to exist in the human brain, emulated by spike jittering and deletion. Given that the human auditory system is highly robust to these noises, it motivates us to investigate the performance of the proposed framework under such noisy conditions.

For spike jittering, we add Gaussian noise with zero mean and standard deviation σ to the spike timing t of all input spikes entering the SNN classifier. The amount of jitter is determined by σ which we sweep from 0.1 T to 0.8 T, where T is the spike generation period. In addition, we also consider spike deletion, where a certain fraction of spikes are corrupted by noise and not delivered to the SNN. For both types of neuronal noise, we trained the model without any noise and then tested it with jittered (of varying standard deviation σ) or deleted (of varying ratio) input spike trains.

3. RESULTS

In this section, we first present the classification results of the proposed SOM-SNN framework for the two benchmark datasets and then compare them with other baseline models. Next, we discuss its early decision-making capability, the effectiveness of using the SOM for feature representation and its underlying hyperparameters, as well as the key differences

TABLE 1 | Comparison of the classification accuracy of the proposed SOM-SNN framework against other ANNs and SNN-based frameworks on the RWCP dataset.

Model	Accuracy (%)
MLP	99.45
CNN	99.85
RNN	95.35
LSTM	98.40
SOM-RNN	97.20
SOM-LSTM	98.15
LSF-SNN (Dennis et al., 2013)	98.50
LTF-SNN (Xiao et al., 2017)	97.50
SOM-SNN (ReSuMe)	97.00
SOM-SNN (Maximum-Margin Tempotron)	99.60

The average results over 10 experimental runs with random weight initialization are reported.

TABLE 2 | Comparison of the classification accuracy of the proposed SOM-SNN framework against other baseline frameworks on the TIDIGITS dataset.

Model	Accuracy (%)
Single-layer SNN and SVM (Tavanaei and Maida, 2017a)[a]	91.00
Spiking CNN and HMM (Tavanaei and Maida, 2017b)[a]	96.00
AER Silicon Cochlea and SVM (Abdollahi and Liu, 2011)[b]	95.58
AER Silicon Cochlea and Deep RNN (Neil and Liu, 2016)[b]	96.10
AER Silicon Cochlea and Phased LSTM (Anumula et al., 2018)[b]	91.25
Liquid State Machine (Zhang et al., 2015)[c]	92.30
MFCC and GRU RNN (Anumula et al., 2018)[c]	97.90
SOM and SNN (this work)[c]	97.40

[a] *Evaluate on the Aurora dataset which was developed from the TIDIGITS dataset.*
[b] *The data was collected by playing the audio files from the TIDIGITS dataset to the AER Silicon Cochlea Sensor.*
[c] *Evaluate on the TIDIGITS dataset.*

between the feedforward SNN-based and RNN-based systems for a temporal classification task. Finally, we demonstrate the improved classification capability of the modified Maximum-Margin Tempotron learning rule and the robustness of the framework against environmental and neuronal noises.

3.1. Classification Results

3.1.1. RWCP Dataset

As shown in **Table 1**, the SOM-SNN model achieved a test accuracy of 99.60%, which is competitive compared with other deep learning and SNN-based models. As described in the experimental set-up, the MLP and CNN models are trained using spectrogram images of fixed dimensions, instead of explicitly modeling the temporal transition of frames. Despite their high accuracy on this dataset, it may be challenging to use them for classifying sound samples of long duration; the temporal structures will be affected inconsistently due to the necessary rescaling of the spectrogram images (Gütig and Sompolinsky, 2009). On the other hand, the RNN and LSTM models capture the temporal transition explicitly. These models are however hard to train for long sound samples due to the vanishing and exploding gradient problem (Greff et al., 2017).

LSF-SNN (Dennis et al., 2013) and LTF-SNN (Xiao et al., 2017) classify the sound samples by first detecting the spectral features in the power spectrogram, and then encoding these features into a spatiotemporal spike pattern for classification by a SNN classifier. In our framework, the SOM is used to learn the key features embedded in the acoustic signals in an unsupervised manner, which is more biologically plausible. Neurons in the SOM become selective to specific spectral features after training, and these features learned by the SOM are more discriminative as shown by the superior SOM-SNN classification accuracy compared with the LSF-SNN and LTF-SNN models.

3.1.2. TIDIGITS Dataset

As shown in **Table 2**, it is encouraging to note that the SOM-SNN framework achieves an accuracy of 97.40%, outperforming all other bio-inspired systems on the TIDIGITS dataset. In Anumula et al. (2018), Abdollahi and Liu (2011), and Neil and Liu (2016), novel systems are designed to work with spike streams generated directly from the AER silicon cochlea sensor. This event-driven auditory front-end generates spike streams asynchronously from 64 bandpass filters spanning over the audible range of the human cochlea. Anumula et al. (Abdollahi and Liu, 2011) provide a comprehensive overview of the asynchronous and synchronous features generated from these raw spike streams, once again highlighting the significant role of discriminative feature representation in speech recognition tasks.

Tavanaei et al. (Tavanaei and Maida, 2017a,b) proposes two biologically plausible feature extractors constructed from SNNs trained using the unsupervised spike-timing-dependent plasticity (STDP) learning rule. The neuronal activations in the feature extraction layer are then transformed into a real-valued feature vector and used to train a traditional classifier, such as the HMM or SVM models. In our work, the features are extracted using the SOM and then used to train a biologically plausible SNN classifier. These different biologically inspired systems represent an important step toward an end-to-end SNN-based automatic speech recognition system.

We note that the traditional RNN based system offers a competitive accuracy of 97.90% (Anumula et al., 2018); our proposed framework, however, is fundamentally different from traditional deep learning approaches. It is worth noting that the network capacity and classification accuracy of our framework can be further improved using multi-layer SNNs.

3.2. Early Decision Making Capability

We note that the SNN-based classifier can identify temporal features within the spatiotemporal spike pattern and generate an output spike as soon as enough discriminative evidence is accumulated. This cumulative decision-making process is more biologically plausible, as it mimics how human makes decisions. A key benefit of such a decision-making process is low latency. As shown in **Figure 3A**, the SNN classifier makes a decision before the whole pattern has been presented. On average, the decision is made when only 50% of the input is presented.

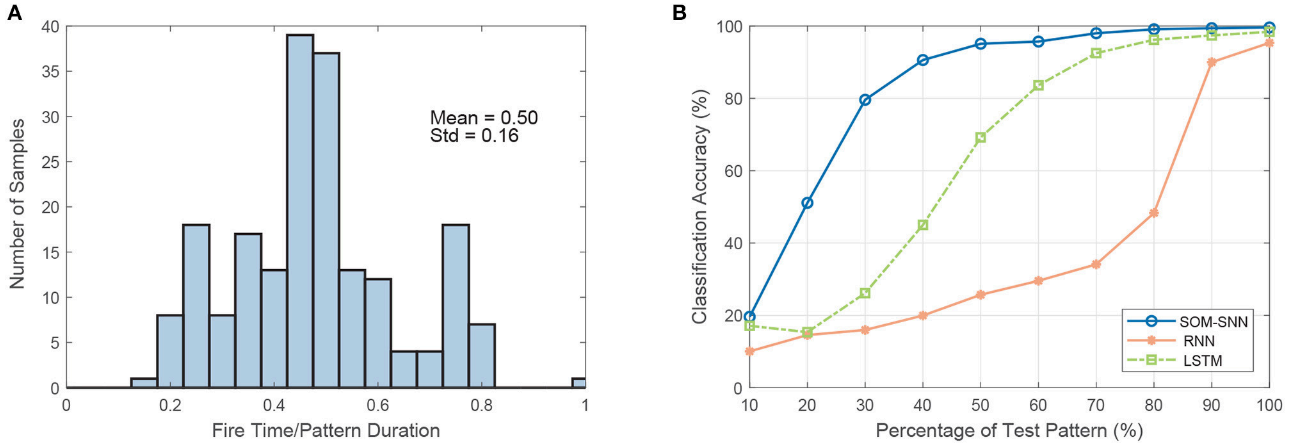


FIGURE 3 | The demonstration of the early decision making capability of the SNN-based classifier. **(A)** The distribution of the number of samples as a function of the ratio of decision time (spike timing) to sample duration on the RWCP test dataset. On average, the SNN-based classifier makes the classification decision when only 50% of the pattern is presented. **(B)** Test accuracy as a function of the percentage of test pattern input to different classifiers (classifiers are trained with full training patterns).

Additionally, we conduct experiments on the SOM-SNN, RNN, and LSTM models, whereby they are trained on the full input patterns but tested with only a partial presentation of the input. The training label is provided to the RNN and LSTM models at the end of each training sequence by default as it is not clear beforehand when enough discriminative features have been accumulated. Likewise, the training labels are provided at the end of input patterns for the SNN classifier. For testing, we increase the duration of the test input pattern presented from 10 to 100% of the actual duration, starting from the beginning of each pattern. As shown in **Figure 3B**, the classification accuracy as a function of the input pattern percentage increases more rapidly for the SNN model. It achieves a satisfactory accuracy of 95.1% when only 50% of the input pattern is presented, much higher than the 25.7 and 69.2% accuracy achieved by the RNN and LSTM models respectively. For the RNN and LSTM models to achieve early decision-making capability, one may require that the models be trained with partial inputs or output labels provided at every time-step. Therefore, SNN-based classifiers demonstrate great potential for real-time temporal pattern classification, compared with state-of-the-art deep learning models such as the RNN and LSTM.

3.3. Feature Representation of the SOM

To visualize the features extracted by the SOM, we plot the BMU activation sequences and their corresponding trajectories on the SOM for a set of randomly selected samples from class "bell5," "bottle1," and "buzzer" in **Figure 4**. We observe low intra-class variability and high inter-class variability in both the BMU activation trajectories and sequences, which are highly desirable for pattern classification. Furthermore, we perform tSNE clustering on the concatenated input vectors entering the SOM and the BMU trajectories generated by the SOM. In **Figure 5A** (input vectors entering the SOM), it can be seen that samples from the same class are distributed over several clusters in 2D space (e.g., class 7, 10). The corresponding BMU vectors, however, merge into a single cluster as shown in **Figure 5B**, suggesting lower intra-class variability achieved by the SOM. The class boundaries for the BMU trajectories may now be drawn as shown in **Figure 5B**, suggesting high inter-class variability. The outliers in **Figure 5B** maybe an artifact due to the uniform rescaling performed on BMU trajectories, a necessary step for tSNE clustering.

We note that the time-warping problem exists in the BMU activation sequences, whereby the duration of sensory stimuli fluctuates from sample to sample within the same class. However, the SNN-based classifier is robust to such fluctuations as shown in the classification results. The decision to fire for a classifying neuron is made based on a time snippet of the spiking pattern; such is the nature of the single spike-based temporal classifier. As long as the BMU activation sequence stays similar, duration fluctuations of input sample will not affect the general trajectory of the membrane potential in each output neuron; the right classification decision, therefore, can be guaranteed. Hence, those outliers in **Figure 5B** underlying the time-warping problem may not necessarily lead to poor classification.

To investigate whether the feature dimension reduction of the SOM is necessary for the SNN classifier to learn different sound categories, we performed experiments that directly input the spike trains of the latency-encoded (20 neurons) (Yu et al., 2013b) or population-encoded (144 neurons) (Bohte et al., 2002) mel-scaled filter bank outputs into the SNN for classification. We find that the SNN classifier is unable to classify such low-level spatiotemporal spike patterns, and only achieve 10.2 and 46.5% classification accuracy for latency- and population-encoded spike patterns, respectively. For both latency- and population-encoded spike patterns, as all encoding neurons spike in every sound frame, albeit with different timing, the synaptic weights therefore either all strengthen or all weaken in the event of misclassification as defined in the Tempotron learning rule.

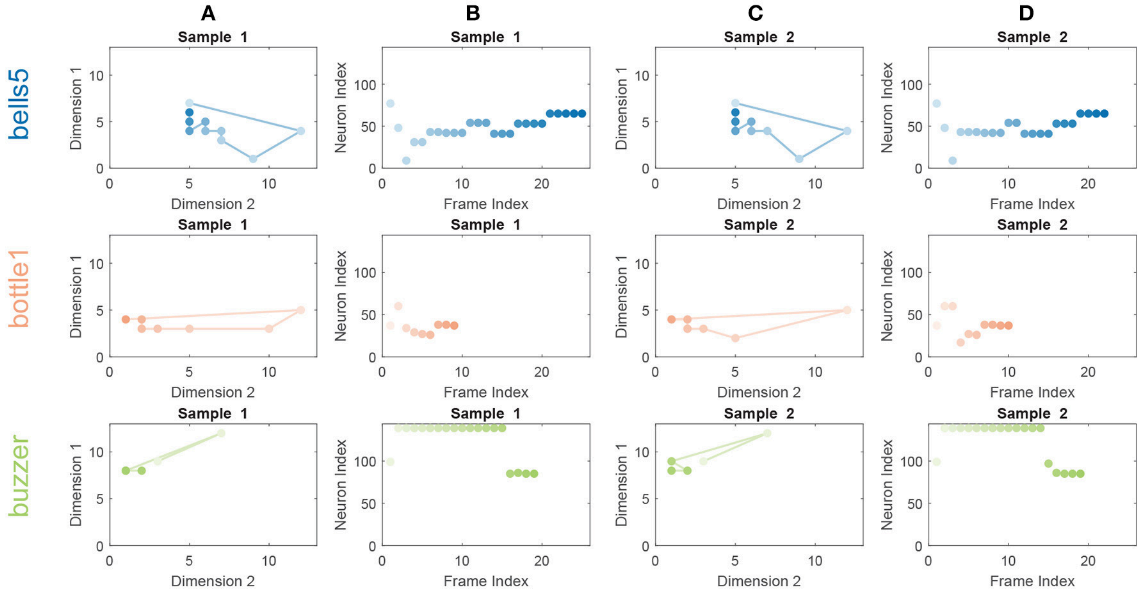


FIGURE 4 | BMU activation trajectories of the SOM **(A,C)** and BMU activation sequences **(B,D)** for randomly selected sound samples from classes "bell5" (Row 1), "bottle1" (Row 2) and "buzzer" (Row 3) of a trained 12 × 12 SOM on the RWCP dataset. For BMU activation trajectories, the lines connect activated BMUs from frame to frame. The activated BMUs are highlighted from light to dark over time. For BMU activation sequences, the neurons of the SOM are enumerated along the y-axis and color matched with neurons in the BMU activation trajectories. The low intra-class variability and high inter-class variability for the BMU activation trajectories and sequences are observed.

Such synchronized weight updates make it challenging for the SNN classifier to find discriminative features embedded within the spike pattern.

As summarized in the section 1, the learning rules for the SNN can be categorized into either membrane-potential based or spike-time based; the Maximum-Margin Tempotron learning rule belongs to the former. To study the synergy between the SOM-based feature representation and spike-time based learning rule, we conducted an experiment using the ReSuMe (Ponulak and Kasiński, 2010) learning rule to train the SNN classifier. For a fair comparison with the Maximum-Margin Tempotron learning rule, we use one output neuron to represent each sound class and each neuron has a single desired output spike. To determine the desired spike timing for each output neuron, we first present all training spiking patterns from the corresponding sound class to the randomly initialized SNN; and monitor the membrane potential trace of the desired output neuron during the simulation. We note the time instant when the membrane potential trace reaches its maximum (denoted as T_{max}) for each sound sample, revealing the most discriminative local temporal feature. We then use the mean of T_{max} across all 20 training samples as the desired output spike time. As shown in **Table 1**, the SNN trained with ReSuMe rule achieves a classification accuracy of 97.0%, which is competitive with other models. This, therefore, demonstrates the compatibility of features extracted by the SOM and spike-time based learning rules, whereby the intra-class variability of sound samples is circumvented by SOM feature extraction such that a single desired spike time for each class suffices.

We note that the SOM functions as an unsupervised sparse feature extractor that provides useful, discriminative input to downstream ANN classifiers. As shown in **Table 1**, the classification accuracy of the SOM-RNN model is better than that of the RNN model alone, and the accuracy of the SOM-LSTM model is also comparable to that of the LSTM model. Additionally, we also notice faster training convergence for both the SOM-RNN and SOM-LSTM models compared to those without the SOM, requiring approximately 25% less number of epochs. This observation may be best explained by the observations made in **Figure 4**, whereby only a subset of the SOM neurons are involved in the spiking patterns of any sound sample (with low intra-class variability and high inter-class variability) which in itself is highly discriminative.

To analyze the effect of different hyperparameters in the SOM on classification accuracy, we perform the following experiments:

Neural Map Size. We sweep the SOM neural map size from 2 × 2 to 16 × 16. As shown in **Figure 6**, we notice improved SNN classification accuracy with larger neural map, which suggests that a larger SOM captures more discriminative features and therefore generates more discriminative spiking patterns for different sound classes. However, the accuracy plateaus once the number of neurons exceeds 120. We suspect that with more neurons the effect of the time-warping problem starts to dominate, leading to more misclassification. Hence, the optimum neural map size has to be empirically determined.

Number of Training Epochs. We sweep the number of training epochs used for the SOM from 100 to 1,000 with an interval of 100. We observe improvements in classification

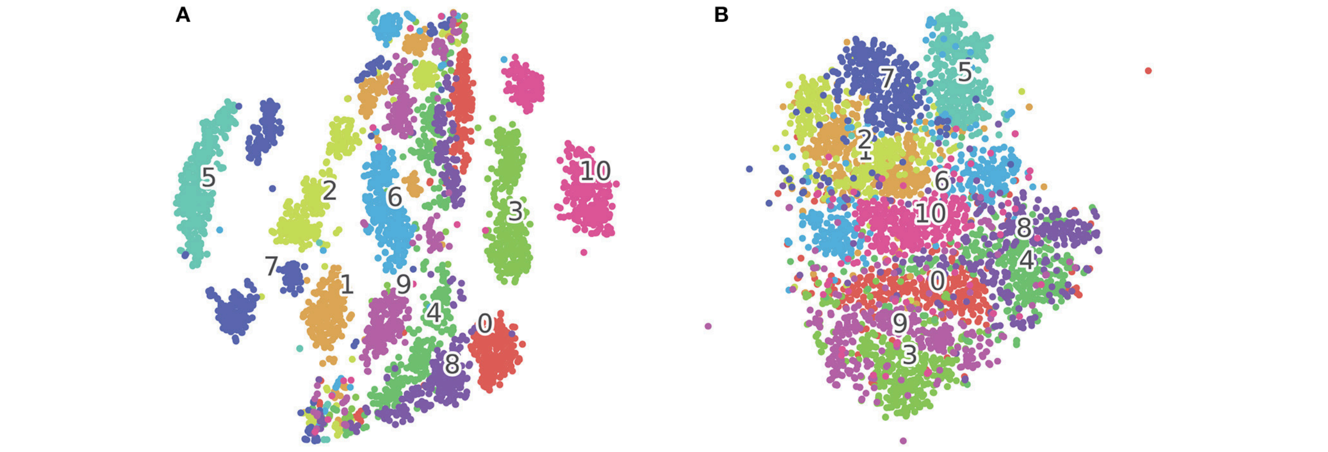


FIGURE 5 | (A) tSNE clustering for concatenated input vectors entering the SOM. **(B)** tSNE clustering for BMU trajectories output from the SOM. Each dot on the figure corresponds to one test sample in the TIDIGITS dataset, the numbers in the figure correspond to class centroids. The samples (e.g., Class 7 and 10) within the same class get closer after being processed by the SOM as shown in this 2D visualization.

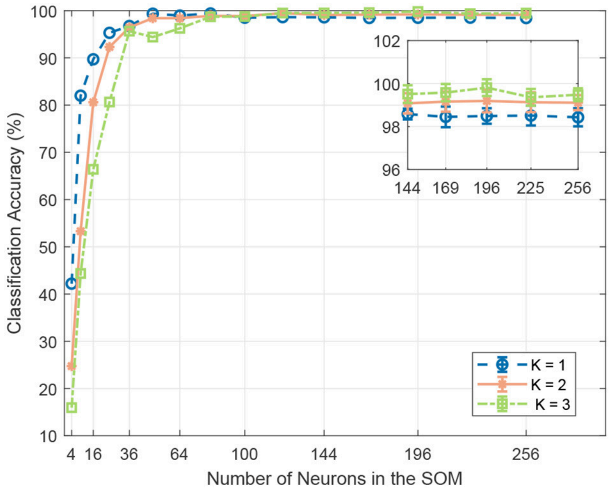


FIGURE 6 | The effect of the SOM neural map size and number of BMUs per frame on classification accuracy. A larger neural map can capture more feature variations and generate more discriminative spiking patterns for different sound events. However, the accuracy plateaus once the number of neurons exceeds 121. As shown in the inset, for neural maps of size above 121, increasing the number *K* of BMUs for each frame enhances system robustness with redundancy and improves classification accuracy.

accuracy of the SNN classifier, with more training epochs of the SOM, which plateaus at 400 for the RWCP dataset.

Number of Activated Neurons. We perform experiments with different number of activated output neurons $K = [1, 2, 3]$ for each sound frame. Specifically, the distances between the SOM output neurons' synaptic weight vectors and the input vector are computed, and the top K neurons with the closest weight vectors will emit a spike. The neural map sizes are swept from 2×2 to 16×16, with number of training epochs fixed at 400. As shown in **Figure 6**, with more activated output neurons in the SOM, the SNN achieves lower classification accuracy for neural map size below 100, while achieving higher accuracy for neural map size larger than that. It can be explained by the fact that for smaller neural maps, given the same number of feature clusters, fewer neurons are allocated to each cluster. Now, with more activated neurons per frame, either fewer clusters can be represented, or the clusters are now less distinguishable from each other. Either way, inter-class variability is reduced, and classification accuracy is adversely affected. This capacity constraint is alleviated with a larger neural map, whereby neighboring neurons are usually grouped into a single feature cluster. As shown in the inset of **Figure 6**, for neural map size larger than 100, more activated neurons per frame improves the feature representation with some redundancy and lead to better classification accuracy. However, it should be noted that with more activated neurons per frame, there are more output spikes generated in the SOM, hence increasing energy consumption. Therefore, a trade-off between classification accuracy and energy consumption has to be made for practical applications.

3.4. Tempotron Learning Rule With Hard Maximum-Margin

As described in section 2, we modify the original Tempotron learning rule by adding a hard margin Δ to the firing threshold V_{thr}. With this modification, we note that the classification accuracy of the SNN increases by 2% consistently with the same SOM dimensions.

To demonstrate how the hard margin Δ improves classification, we show two samples which have been misclassified by the SNN classifier trained with the original Tempotron rule (**Figures 7A,B**), but correctly classified by the Maximum-Margin Tempotron rule (**Figures 7C,D**). In **Figure 7A**, both output neurons (i.e., "ring" and "bottle1") are selective to the discriminative local feature occurring between 2 and 10 ms. While in **Figure 7B**, the discriminative local feature is overlooked by the desired

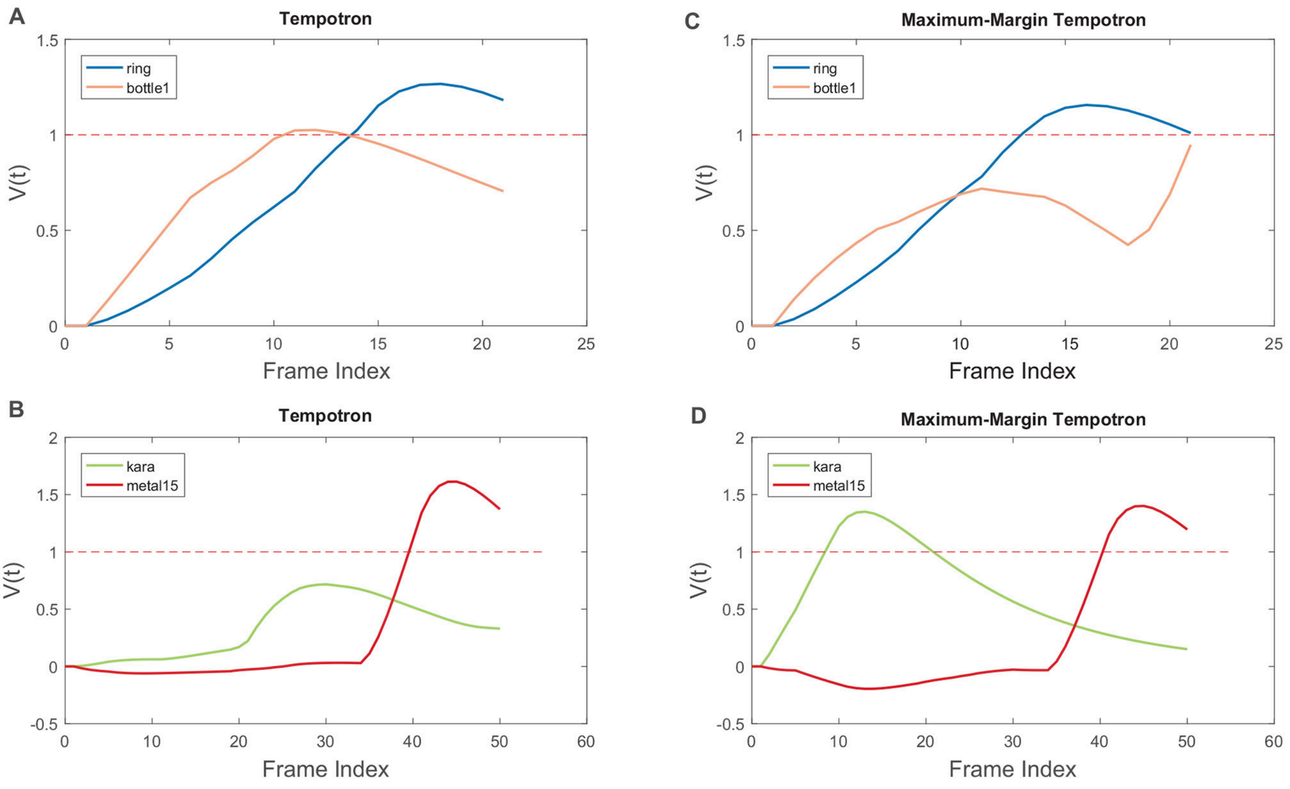


FIGURE 7 | Selected samples misclassified by the Tempotron learning rule, while classified correctly by the modified Maximum-Margin Tempotron learning rule. Sample from the "ring" class misclassified as "bottle1" **(A)**, while correctly classified with Maximum-Margin Tempotron learning rule **(C)**. Sample from the "kara" class misclassified as "metal15" **(B)**, while correctly classified with Maximum-Margin Tempotron learning rule **(D)**.

output neuron, possibly due to the time-warping, and the output neuron representing another class fires erroneously afterward.

When trained with the additional hard margin Δ, the negative output neuron representing the "bottle1" class is suppressed and prevented from firing (**Figure 7C**). Similarly, the negative output neuron representing the "metal15" class is also slightly suppressed, while the positive output neuron representing the "kara" class undergoes LTP and correctly crosses the V_{thr} (**Figure 7D**). Therefore, the additional hard margin Δ ensures a better separation between the positive and negative classes and improves classification accuracy.

Since the relative ratio between the hard margin Δ and the firing threshold V_{thr} is an important hyper-parameter, we investigate its effect on the classification accuracy using the RWCP dataset by sweeping it from 0 to 1.2 with an interval of 0.1. The experiments are repeated 20 times for each ratio value with random weight initialization. For simplicity, we only study the symmetric cases whereby the hard margin has the same absolute value for both positive and negative neurons. For the case when the ratio is 0, the learning rule is reduced to the standard Tempotron rule. As shown in **Figure 8**, the hard margin Δ improves the classification accuracy consistently for ratios below 1.0, and the best accuracy is achieved with a ratio of 0.5. The accuracy drops significantly for ratio above 0.9, suggesting a high level of margin may interfere with learning and lead to brittle models.

3.5. Robustness to Noise

3.5.1. Environmental Noise

We report the classification accuracies over 10 runs with random weight initialization in **Tables 3**, **4** for mismatched and multi-condition training respectively.

We note that under the mismatched condition, the classification accuracy for all models degrades dramatically with an increasing amount of noise and falls below 50% with SNR at 10 dB. The LSF-SNN and LTF-SNN models use local key points on the spectrogram as features to represent the sound sample, and are therefore robust to noise under such conditions. However, the biological evidence for such spectrogram features is currently lacking.

As shown in **Table 4**, multi-condition training effectively addresses the problem of performance degradation under noisy conditions, whereby MLP, CNN, LSTM, and SOM-SNN models have achieved classification accuracies above 95% even at the challenging 0 dB SNR. Similar to observations made in McLoughlin et al. (2015), we note that the improved robustness to noise comes with a trade-off in terms of accuracy for clean sounds, as demonstrated in the results for the ANN models.

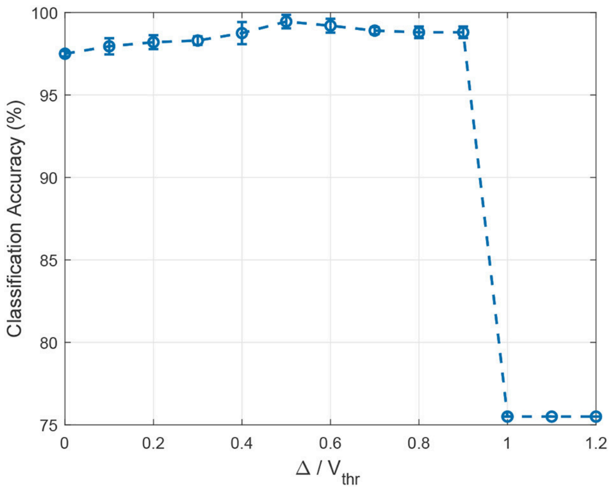


FIGURE 8 | The effect of the ratio between the hard margin Δ and the firing threshold V_{thr} on classification accuracy. For $\Delta/V_{thr} = 0$, the learning rule is reduced to the standard Tempotron rule. The hard margin Δ improves the classification accuracy for ratios below 1.0, while the accuracy drops significantly afterward. The best accuracy is achieved with a ratio of 0.5 on the RWCP dataset.

However, the classification accuracies improve across the board for the SOM-SNN model under all acoustic conditions using the multi-condition training, achieving an accuracy of 98.7% even for the challenging case of -5 dB SNR. The SOM-SNN model hence offers an attractive alternative to other models especially when a single trained model has to operate under varying noise levels.

3.5.2. Spike Jittering

As shown in **Figure 9A**, the SOM-SNN model is shown to be highly robust to spike jittering and maintains a high accuracy independent of the number of neurons activated per sound frame in the SOM. We suspect that given only a small subset of neurons in the SOM are involved for each sound class, the requirement of the SNN for precise spike timing is relaxed.

3.5.3. Spike Deletion

As shown in **Figure 9B**, the SOM-SNN model maintains a high classification accuracy when spike deletion is performed on the input to the SNN. As only a small subset of pre-synaptic neurons in the SOM deliver input spikes to the SNN for each sound class, with high inter-class variability, the SNN classifier is still able to classify correctly even with some input spike deletion. The peak membrane potential value is used in some cases to make the correct classification.

4. DISCUSSION

In this paper, we propose a biologically plausible SOM-SNN framework for automatic sound classification. This framework integrates the auditory front-end, feature representation learning and temporal classification in a unified framework. Biological plausibility is a key consideration in the design of our framework, which distinguishes it from many other machine learning frameworks.

The SOM-SNN framework is organized in a modular manner, whereby acoustic signals are pre-processed using a biologically plausible auditory front-end, the mel-scaled filter bank, for frequency content analysis. This framework emulates the functionality of the human cochlea and the non-linearity of human perception of sound (Bear et al., 2016). Although it is still not clear how information is represented and processed in the auditory cortex, it has been shown that certain neural populations in the cochlear nuclei and primary auditory cortex are organized in a tonotopic fashion (Pantev et al., 1995; Bilecen et al., 1998). Motivated by this, the biologically plausible SOM is used for the feature extraction and representation of mel-scaled filter bank outputs. The selectivity of neurons in the SOM emerges from unsupervised training and organizes in a tonotopic fashion, whereby adjacent neurons share similar weight vectors. The SOM effectively improves pattern separation, whereby each sound frame originally represented by a 20-dimensional vector (mel-scaled filter bank output coefficients) is translated into a single output spike. The resulting BMU activation sequences are shown to have the property of low intra-class variability and high inter-class variability. Consequently, the SOM provides an effective and sparse representation of acoustic signals as observed in the auditory cortex (Hromádka et al., 2008). Additionally, the feature representation of the SOM was shown to be useful inputs for RNN and LSTM classifiers in our experiments.

Although the SOM is biologically inspired by cortical maps in the human brain, it lacks certain characteristics of the biological neuron, such as spiking output and access to only local information. Other studies (Rumbell et al., 2014; Hazan et al., 2018) have shed light on the feasibility of using spiking neurons and spike-timing dependent plasticity (STDP) learning rule (Song et al., 2000) to model the SOM. We would investigate how we may integrate the spiking-SOM and the SNN classifier for classification tasks in the future.

Acoustic signals exhibit large variations not only in their frequency contents but also in temporal structures. State-of-the-art machine learning based ASC systems model the temporal transition explicitly, using the HMM, RNN or LSTM, while our work focuses on building a biologically plausible temporal classifier based on the SNN. For efficient training, we use supervised temporal learning rules, namely the membrane-potential based Maximum-Margin Tempotron and spike-timing based ReSuMe. The Maximum-Margin Tempotron (combining the Tempotron rule with the maximum-margin classifier) ensures a better separation between the positive and negative classes, improving classification accuracy in our experiments. As demonstrated in our experiments, the SOM-SNN framework achieves comparable classification results on both the RWCP and TIDIGITS datasets against other deep learning and SNN-based models.

We further discover that the SNN-based classifier has an early decision making capability: making a classification decision when only part of the input is presented. In our

TABLE 3 | Average classification accuracy of different models under the mismatched-condition.

SNR	MLP	CNN	RNN	LSTM	SOM-SNN
Clean	99.45 ± 0.35%	**99.85 ± 0.23%**	95.35 ± 1.06%	98.40 ± 0.86%	99.60 ± 0.15%
20 dB	55.05 ± 4.30%	61.5 ± 4.71%	25.15 ± 8.86%	47.20 ± 5.36%	**79.15±3.70%**
10 dB	32.10 ± 8.38%	**42.70 ± 5.84%**	11.85 ± 2.06%	34.50 ± 10.61%	36.25±1.25%
0 dB	24.60 ± 4.94%	**28.40 ± 6.60%**	10.10 ± 1.64%	22.35 ± 6.63%	26.50 ± 1.29%
-5 dB	18.40 ± 4.58%	**22.65 ± 5.08%**	9.20 ± 1.98%	16.60 ± 7.00%	19.55 ± 0.16%
Average	45.92%	51.02%	30.33%	43.81%	**52.21%**

Experiments are conducted over 10 runs with random weight initialization.
The bold values indicate the best classification accuracies under different SNR.

TABLE 4 | Average classification accuracy of different models with multi-condition training.

SNR	MLP	CNN	RNN	LSTM	SOM-SNN
Clean	96.10 ± 1.18%	97.60 ± 0.89%	94.30 ± 3.04%	98.15 ± 0.71%	**99.80 ± 0.22%**
20 dB	98.45 ± 0.61%	99.50 ± 0.22%	94.30 ± 2.70%	99.10 ± 0.89%	**100.00 ± 0.00%**
10 dB	99.35 ± 0.45%	99.70 ± 0.33%	95.25 ± 2.49%	99.05 ± 1.25%	**100.00 ± 0.00%**
0 dB	98.20 ± 1.45%	99.45 ± 0.75%	93.65 ± 2.82%	95.80 ± 3.93%	**99.45 ± 0.55%**
-5 dB	92.50 ± 1.53%	98.35 ± 0.78%	86.85 ± 5.20%	91.35 ± 4.82%	**98.70 ± 0.48%**
Average	96.92%	98.92%	92.87%	96.69%	**99.59%**

Experiments are conducted over 10 runs with random weight initialization.
The bold values indicate the best classification accuracies under different SNR.

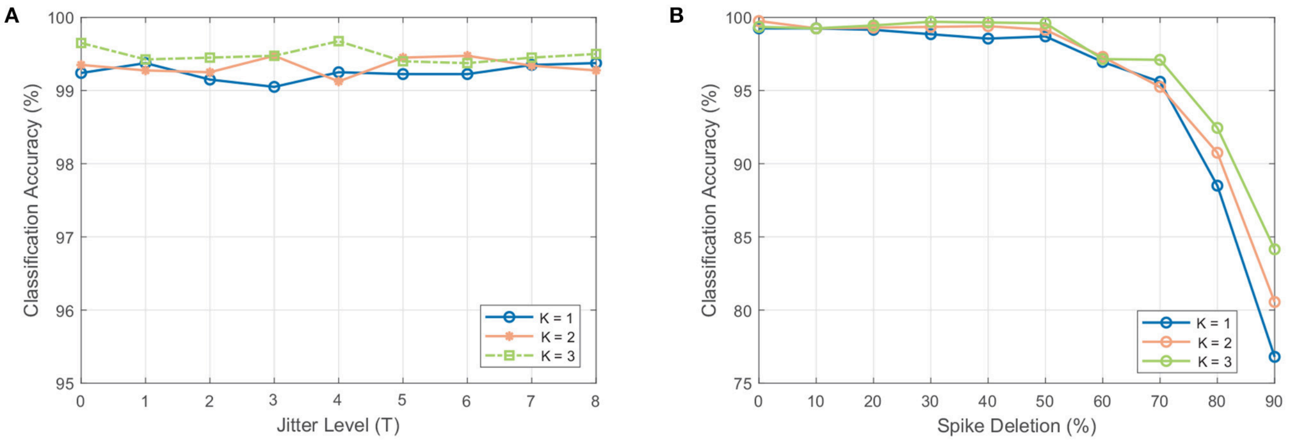


FIGURE 9 | The effect of spike jittering and spike deletion on the classification accuracy. **(A)** Classification accuracy as a result of spike jitter added at the input to the SNN classifier. The amount of jitter is added as a fraction of the spike generation period *T* (i.e., 50 ms used for the RWCP dataset). The classifier is robust to spike jitter, maintaining a high accuracy with different amount of jitter. **(B)** Classification accuracy as a result of spike deletion at the input to the SNN classifier. The accuracy of the classifier remains stable for spike deletion ratio below 60% and decays with increased spike deletion.

experiments, the SNN-based classifier achieves an accuracy of 95.1%, significantly higher than those of the RNN and LSTM (25.7% and 69.2% respectively) when only 50% of the input pattern is presented. This early decision making capability can be further exploited in noisy environments, as exemplified by the cocktail party problem (Haykin and Chen, 2005). The SNN-based classifier can potentially identify discriminative temporal features and classify accordingly from a time snippet of the acoustic signals that are less distorted, which is desirable for an environment with fluctuating noise.

Environmental noise poses a significant challenge to the robustness of any sound classification systems: the accuracy of many such systems degrade rapidly with an increasing amount of noise as shown in our experiments. Multi-condition training, whereby the model is trained with noise-corrupted sound samples, is shown to overcome this challenge effectively. In contrast to the DNN and SVM classifiers (McLoughlin et al., 2015), there is no trade-off in performance for clean sounds in the SOM-SNN framework with multi-condition training; probably because the classification decision is made based on local temporal patterns. Additionally, noise is also known to exist in the central nervous system (Schneidman, 2001; van Rossum et al., 2003) which can be simulated by spike jittering and deletion. Notably, the SOM-SNN framework is shown to be highly robust to such noises introduced to spike inputs arriving at the SNN classifier.

The SNN classifier makes a decision based on a single local discriminative feature which often only lasts for a fraction of the pattern duration, as a direct consequence of the Maximum-Margin Tempotron learning rule. We expect improved accuracy when more such local features within a single spike pattern are utilized for classification, which may be learned using the multi-spike Tempotron (Gütig, 2016; Yu et al., 2018). The accuracy of the SOM-SNN model trained with the ReSuMe learning rule may also be improved by using multiple spike times. However, defining these desired spike times is a challenge exacerbated by increasing intra-class variability. Although the existing single-layer SNN classifier has achieved promising results on both benchmark datasets, it is not clear how the proposed framework may scale for more challenging datasets. Recently, there is progress made in training multi-layer SNNs (Lee et al., 2016; Neftci et al., 2017; Wu et al., 2018b), which could significantly increase model capacity and classification accuracy. For future work, we would investigate how to incorporate these multi-spike and multi-layer SNN classifiers into our framework for more challenging large-vocabulary speech recognition tasks.

For real-life applications such as audio surveillance, we may add inhibitory connections between output neurons to reset all neurons once the decision has been made (i.e., a winner-takes-all mechanism). This allows output neurons to compete once again and spike upon receipt of a new local discriminative spike pattern. The firing history of all output neurons can then be analyzed so as to understand the audio scene.

The computational cost and memory bandwidth requirements of our framework would be the key concerns in a neuromorphic hardware implementation. As the proposed framework is organized in a pipelined manner, the computational cost could be analyzed independently for the auditory front-end, SOM and SNN classifier. For the auditory front-end, our implementation is similar to that of the MFCC. As evaluated in Anumula et al. (2018), the MFCC implementation is computationally more costly compared to the spike trains generated directly from the neuromorphic cochlea sensor. Our recent work (Pan et al., 2018) proposes a novel time-domain frequency filtering scheme which addresses the cost issue in MFCC implementation. We expect the SOM to be the main computational bottleneck of the proposed framework. For each sound frame, the calculation of the Euclidean distance of synaptic weights from the input vector is done for each SOM neuron. Additionally, the distances are required to be sorted so as to determine the best-matching units. However, this computational bottleneck can be addressed with the spiking-SOM implementation (Rumbell et al., 2014; Hazan et al., 2018), whereby the winner neuron spikes the earliest and inhibits all other neurons from firing (i.e., a winner-takes-all mechanism) and hence by construction, the BMU. The spiking-SOM also facilitates the implementation of the whole framework on a neuromorphic hardware. In tandem with the SNN classifier, a fully SNN-based framework when implemented would translate to significant power saving.

As for memory bandwidth requirements, the synaptic weight matrices connecting the auditory front-end with the SOM and the SOM with the SNN classifier are the two major components for memory storage and retrieval. For the synaptic connections between the auditory front-end and the SOM, the memory bandwidth increases quadratically with the product of the number of neurons in the SOM and the dimensionality of the filter banks. Since the number of output neurons is equal to the total number of classes and hence fixed, the memory bandwidth only increases linearly with the number of neurons in the SOM. Therefore, the number of neurons in the SOM should be carefully designed for a particular application considering the trade-off between classification accuracy and hardware efficiency.

AUTHOR CONTRIBUTIONS

JW performed all the experiments. All authors contributed to the experiments design, results interpretation and writing.

FUNDING

This research is supported by Programmatic Grant No. A1687b0033 from the Singapore Government's Research, Innovation and Enterprise 2020 plan (Advanced Manufacturing and Engineering domain).

REFERENCES

Abdollahi, M., and Liu, S. C. (2011). "Speaker-independent isolated digit recognition using an aer silicon cochlea," in *2011 IEEE Biomedical Circuits and Systems Conference (BioCAS)* (La Jolla, CA), 269–272.

Amir, A., Taba, B., Berg, D., Melano, T., McKinstry, J., Di Nolfo, C., et al. (2017). "A low power, fully event-based gesture recognition system," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Honolulu), 7388–7397.

Anumula, J., Neil, D., Delbruck, T., and Liu, S. C. (2018). Feature representations for neuromorphic audio spike streams. *Front. Neurosci.* 12:23. doi: 10.3389/fnins.2018.00023

Bear, M., Connors, B., and Paradiso, M. (2016). *Neuroscience: Exploring the Brain, 4th Edn*. Philadelphia, PA: Wolters Kluwer.

Bilecen, D., Scheffler, K., Schmid, N., Tschopp, K., and Seelig, J. (1998). Tonotopic organization of the human auditory cortex as detected by bold-fmri. *Hear. Res.* 126, 19–27. doi: 10.1016/S0378-5955(98)00139-7

Bohte, S. M., Kok, J. N., and La Poutre, H. (2002). Error-backpropagation in temporally encoded networks of spiking neurons. *Neurocomputing* 48, 17–37. doi: 10.1016/S0925-2312(01)00658-0

Brandli, C., Berner, R., Yang, M., Liu, S. C., and Delbruck, T. (2014). A 240× 180 130 db 3 μs latency global shutter spatiotemporal vision sensor. *IEEE J. Solid-State Circ.* 49, 2333–2341. doi: 10.1109/JSSC.2014.2342715

Chu, S., Narayanan, S., and Kuo, C.-C. J. (2009). Environmental sound recognition with time-frequency audio features. *IEEE Trans. Audio Speech Lang. Process.* 17, 1142–1158. doi: 10.1109/TASL.2009.2017438

Cortes, C., and Vapnik, V. (1995). Support-vector networks. *Mach. Learn.* 20, 273–297. doi: 10.1007/BF00994018

Delbrück, T., Linares-Barranco, B., Culurciello, E., and Posch, C. (2010). "Activity-driven, event-based vision sensors," in *Proceedings of 2010 IEEE International Symposium on Circuits and Systems* (Paris), 2426–2429.

Dennis, J., Tran, H. D., and Li, H. (2011). Spectrogram image feature for sound event classification in mismatched conditions. *IEEE Signal Process. Lett.* 18, 130–133. doi: 10.1109/LSP.2010.2100380

Dennis, J., Yu, Q., Tang, H., Tran, H. D., and Li, H. (2013). "Temporal coding of local spectrogram features for robust sound recognition," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing* (Vancouver, BC), 803–807.

Fei-Fei, L., Fergus, R., and Perona, P. (2006). One-shot learning of object categories. *IEEE Trans. Patt. Anal. Mach. Intell.* 28, 594–611. doi: 10.1109/TPAMI.2006.79

Furber, S. B., Lester, D. R., Plana, L. A., Garside, J. D., Painkras, E., Temple, S., et al. (2013). Overview of the spinnaker system architecture. *IEEE Trans. Comput.* 62, 2454–2467. doi: 10.1109/TC.2012.142

Graves, A., Mohamed, A., and Hinton, G. E. (2013). "Speech recognition with deep recurrent neural networks," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing (IEEE)* (Vancouver, BC), 6645–6649.

Greff, K., Srivastava, R. K., Koutnik, J., Steunebrink, B. R., and Schmidhuber, J. (2017). Lstm: a search space odyssey. *IEEE Trans. Neural Netw. Learn. Syst.* 28, 2222–2232. doi: 10.1109/TNNLS.2016.2582924

Guo, G., and Li, S. Z. (2003). Content-based audio classification and retrieval by support vector machines. *IEEE Trans. Neural Netw.* 14, 209–215. doi: 10.1109/TNN.2002.806626

Gütig, R. (2016). Spiking neurons can discover predictive features by aggregate-label learning. *Science* 351:aab4113. doi: 10.1126/science.aab4113

Gütig, R., and Sompolinsky, H. (2006). The tempotron: a neuron that learns spike timing-based decisions. *Nat. Neurosci.* 9:420. doi: 10.1038/nn1643

Gütig, R., and Sompolinsky, H. (2009). Time-warp–invariant neuronal processing. *PLoS Biol.* 7:e1000141. doi: 10.1371/journal.pbio.1000141

Haykin, S., and Chen, Z. (2005). The cocktail party problem. *Neural Comput.* 17, 1875–1902. doi: 10.1162/0899766054322964

Hazan, H., Saunders, D., Sanghavi, D. T., Siegelmann, H., and Kozma, R. (2018). "Unsupervised learning with self-organizing spiking neural networks," in *2018 International Joint Conference on Neural Networks (IJCNN)* (Rio de Janeiro).

Hochreiter, S., and Schmidhuber, J. (1997). Long short-term memory. *Neural Comput.* 9, 1735–1780. doi: 10.1162/neco.1997.9.8.1735

Hromádka, T., DeWeese, M. R., and Zador, A. M. (2008). Sparse representation of sounds in the unanesthetized auditory cortex. *PLoS Biol.* 6:e16. doi: 10.1371/journal.pbio.0060016

Kingma, D., and Ba, J. (2014). Adam: a method for stochastic optimization. arXiv:1412.6980.

Kohonen, T. (1998). The self-organizing map. *Neurocomputing* 21, 1–6. doi: 10.1016/S0925-2312(98)00030-7

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems* (Stateline), 1097–1105.

Kwak, C., and Kwon, O. W. (2012). Cardiac disorder classification by heart sound signals using murmur likelihood and hidden markov model state likelihood. *IET Signal Process.* 6, 326–334. doi: 10.1049/iet-spr.2011.0170

Lee, J. H., Delbruck, T., and Pfeiffer, M. (2016). Training deep spiking neural networks using backpropagation. *Front. Neurosci.* 10:508. doi: 10.3389/fnins.2016.00508

Leng, Y. R., Tran, H. D., Kitaoka, N., and Li, H. (2012). Selective gammatone envelope feature for robust sound event recognition. *IEICE Trans. Inform. Syst.* 95, 1229–1237. doi: 10.1587/transinf.E95.D.1229

Leonard, R. G., and Doddington, G. (1993). *Tidigits Speech Corpus*. Philadelphia, PA: Linguistic Data Consortium.

Liu, S. C., van Schaik, A., Minch, B. A., and Delbruck, T. (2014). Asynchronous binaural spatial audition sensor with 2644 channel output. *IEEE Trans. Biomed. Circ. Syst.* 8, 453–464. doi: 10.1109/TBCAS.2013.2281834

McLoughlin, I., Zhang, H., Xie, Z., Song, Y., and Xiao, W. (2015). Robust sound event classification using deep neural networks. *IEEE/ACM Trans. Audio Speech Lang. Process.* 23, 540–552. doi: 10.1109/TASLP.2015.2389618

Merolla, P. A., Arthur, J. V., Alvarez-Icaza, R., Cassidy, A. S., Sawada, J., Akopyan, F., et al. (2014). A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science* 345, 668–673. doi: 10.1126/science.1254642

Mitrović, D., Zeppelzauer, M., and Breiteneder, C. (2010). Features for content-based audio retrieval. *Adv. Comput.* 78, 71–150. doi: 10.1016/S0065-2458(10)78003-7

Møller, A. R. (2012). *Hearing: Anatomy, Physiology, and Disorders of the Auditory System*. Plural Publishing.

Morgan, N., and Bourlard, H. (1990). "Continuous speech recognition using multilayer perceptrons with hidden markov models," in *1990 IEEE International Conference on Acoustics, Speech and Signal Processing (IEEE)* (Albuquerque, NM), 413–416.

Neftci, E. O., Augustine, C., Paul, S., and Detorakis, G. (2017). Event-driven random back-propagation: enabling neuromorphic deep learning machines. *Front. Neurosci.* 11:324. doi: 10.3389/fnins.2017.00324

Neil, D., and Liu, S. C. (2016). "Effective sensor fusion with event-based sensors and deep network architectures," in *2016 IEEE International Symposium on Circuits and Systems (ISCAS)* (Montreal, QC), 2282–2285.

Nishiura, T., and Nakamura, S. (2002). "An evaluation of sound source identification with rwcp sound scene database in real acoustic environments," in *Proceedings IEEE International Conference on Multimedia and Expo*, Vol. 2, (Lausanne), 265–268.

Palatucci, M., Pomerleau, D., Hinton, G. E., and Mitchell, T. M. (2009). "Zero-shot learning with semantic output codes," in *Advances in Neural Information Processing Systems* (Vancouver, BC), 1410–1418.

Pan, Z., Chua, Y., Wu, J., and Li, H. (2018). "An event-based cochlear filter temporal encoding scheme for speech signals," in *2018 International Joint Conference on Neural Networks (IJCNN)* (Rio de Janeiro), 1–8.

Pantev, C., Bertrand, O., Eulitz, C., Verkindt, C., Hampson, S., Schuierer, G., and Elbert, T. (1995). Specific tonotopic organizations of different areas of the human auditory cortex revealed by simultaneous magnetic and electric recordings. *Electroencephalogr. Clin. Neurophysiol.* 94, 26–40. doi: 10.1016/0013-4694(94)00209-4

Ponulak, F., and Kasiński, A. (2010). Supervised learning in spiking neural networks with ReSuMe: sequence learning, classification, and spike shifting. *Neural Comput.* 22, 467–510. doi: 10.1162/neco.2009.11-08-901

Rabaoui, A., Davy, M., Rossignol, S., and Ellouze, N. (2008). Using one-class svms and wavelets for audio surveillance. *IEEE Trans. Inform. Forens. Secur.* 3, 763–775. doi: 10.1109/TIFS.2008.2008216

Rumbell, T., Denham, S. L., and Wennekers, T. (2014). A spiking self-organizing map combining stdp, oscillations, and continuous learning. *IEEE Trans. Neural Netw. Learn. Syst.* 25, 894–907. doi: 10.1109/TNNLS.2013.2283140

Saxe, A. M., McClelland, J. L., and Ganguli, S. (2013). Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. arXiv:1312.6120.

Schneidman, E. (2001). *Noise and Information in Neural Codes*. PhD thesis, Hebrew University.

Serrano-Gotarredona, T., Linares-Barranco, B., Galluppi, F., Plana, L., and Furber, S. (2015). "Convnets experiments on spinnaker," in *2015 IEEE International Symposium on Circuits and Systems (ISCAS)* (Lisbon), 2405–2408.

Sharan, R. V., and Moir, T. J. (2016). An overview of applications and advancements in automatic sound recognition. *Neurocomputing* 200, 22–34. doi: 10.1016/j.neucom.2016.03.020

Song, S., Miller, K. D., and Abbott, L. F. (2000). Competitive hebbian learning through spike-timing-dependent synaptic plasticity. *Nat. Neurosci.* 3:919. doi: 10.1038/78829

Tavanaei, A., and Maida, A. (2017a). "Bio-inspired multi-layer spiking neural network extracts discriminative features from speech signals," in *International Conference on Neural Information Processing* (Guangzhou: Springer), 899–908.

Tavanaei, A., and Maida, A. (2017b). A spiking network that learns to extract spike signatures from speech signals. *Neurocomputing* 240, 191–199. doi: 10.1016/j.neucom.2017.01.088

van Rossum, M. C., O'Brien, B. J., and Smith, R. G. (2003). Effects of noise on the spike timing precision of retinal ganglion cells. *J. Neurophysiol.* 89, 2406–2419. doi: 10.1152/jn.01106.2002

Varga, A., and Steeneken, H. J. (1993). Assessment for automatic speech recognition: II. noisex-92: a database and an experiment to study the effect of additive noise on speech recognition systems. *Speech Commun.* 12, 247–251. doi: 10.1016/0167-6393(93)90095-3

Wu, J., Chua, Y., and Li, H. (2018a). "A biologically plausible speech recognition framework based on spiking neural networks," in *2018 International Joint Conference on Neural Networks (IJCNN)* (Rio de Janeiro), 1–8.

Wu, Y., Deng, L., Li, G., Zhu, J., and Shi, L. (2018b). Spatio-temporal backpropagation for training high-performance spiking neural networks. *Front. Neurosci.* 12:331. doi: 10.3389/fnins.2018.00331

Xiao, R., Yan, R., Tang, H., and Tan, K. C. (2017). *A Spiking Neural Network Model for Sound Recognition*, Singapore: Springer.

Yu, Q., Li, H., and Tan, K. C. (2018). Spike timing or rate? Neurons learn to make decisions for both through threshold-driven plasticity. *IEEE Trans. Cybern.* 1–12. doi: 10.1109/TCYB.2018.2821692

Yu, Q., Tang, H., Tan, K. C., and Li, H. (2013a). Precise-spike-driven synaptic plasticity: learning hetero-association of spatiotemporal spike patterns. *PLoS ONE* 8:e78318. doi: 10.1371/journal.pone.0078318

Yu, Q., Tang, H., Tan, K. C., and Li, H. (2013b). Rapid feedforward computation by temporal encoding and learning with spiking neurons. *IEEE Trans. Neural Netw. Learn. Syst.* 24, 1539–1552. doi: 10.1109/TNNLS.2013.2245677

Zhang, M., Qu, H., Belatreche, A., and Xie, X. (2017). EMPD: an efficient membrane potential driven supervised learning algorithm for spiking neurons. *IEEE Trans. Cogn. Dev. Syst.* 10, 151–162. doi: 10.1109/TCDS.2017.2651943

Zhang, Y., Li, P., Jin, Y., and Choe, Y. (2015). A digital liquid state machine with biologically inspired learning and its application to speech recognition. *IEEE Trans. Neural Netw. Learn. Syst.* 26, 2635–2649. doi: 10.1109/TNNLS.2015.2388544

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers
in Neuroscience

ORIGINAL RESEARCH
published: 03 April 2018
doi: 10.3389/fnins.2018.00194

Biologically-Inspired Spike-Based Automatic Speech Recognition of Isolated Digits Over a Reproducing Kernel Hilbert Space

Kan Li and José C. Príncipe*

Computational NeuroEngineering Laboratory, Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL, United States

This paper presents a novel real-time dynamic framework for quantifying time-series structure in spoken words using spikes. Audio signals are converted into multi-channel spike trains using a biologically-inspired leaky integrate-and-fire (LIF) spike generator. These spike trains are mapped into a function space of infinite dimension, i.e., a Reproducing Kernel Hilbert Space (RKHS) using point-process kernels, where a state-space model learns the dynamics of the multidimensional spike input using gradient descent learning. This kernelized recurrent system is very parsimonious and achieves the necessary memory depth via feedback of its internal states when trained discriminatively, utilizing the full context of the phoneme sequence. A main advantage of modeling nonlinear dynamics using state-space trajectories in the RKHS is that it imposes no restriction on the relationship between the exogenous input and its internal state. We are free to choose the input representation with an appropriate kernel, and changing the kernel does not impact the system nor the learning algorithm. Moreover, we show that this novel framework can outperform both traditional hidden Markov model (HMM) speech processing as well as neuromorphic implementations based on spiking neural network (SNN), yielding accurate and ultra-low power word spotters. As a proof of concept, we demonstrate its capabilities using the benchmark TI-46 digit corpus for isolated-word automatic speech recognition (ASR) or keyword spotting. Compared to HMM using Mel-frequency cepstral coefficient (MFCC) front-end without time-derivatives, our MFCC-KAARMA offered improved performance. For spike-train front-end, spike-KAARMA also outperformed state-of-the-art SNN solutions. Furthermore, compared to MFCCs, spike trains provided enhanced noise robustness in certain low signal-to-noise ratio (SNR) regime.

Keywords: spike-based learning, noise-robust automatic speech recognition (ASR), keyword spotting, kernel adaptive filtering (KAF), reproducing kernel Hilbert space (RKHS), kernel method, neuromorphic computation

Edited by:
Shih-Chii Liu,
ETH Zürich, Switzerland

Reviewed by:
Shantanu Chakrabartty,
Washington University in St. Louis,
United States
Malcolm Slaney,
Google, United States
Behtash Babadi,
University of Maryland, College Park,
United States

***Correspondence:**
Kan Li
likan@ufl.edu

Specialty section:
This article was submitted to
Neuromorphic Engineering,
a section of the journal
Frontiers in Neuroscience

Received: *06 October 2017*
Accepted: *12 March 2018*
Published: *03 April 2018*

Citation:
Li K and Príncipe JC (2018)
Biologically-Inspired Spike-Based
Automatic Speech Recognition of
Isolated Digits Over a Reproducing
Kernel Hilbert Space.
Front. Neurosci. 12:194.
doi: 10.3389/fnins.2018.00194

1. INTRODUCTION

Automatic speech recognition (ASR) or the task of translating audio signal into text is an especially challenging problem due to both the non-stationarity of speech signal and the large variations in its spatiotemporal representation. Particularly, the variability in the temporal dimension of speech signal prevents state-of-the-art pattern classifiers such as support vector machines (SVMs)

(Scholkopf and Smola, 2001), which are limited to static patterns or fixed (constant) dimension inputs, from being implemented in a straightforward manner. Compounding the issue is that performance often degrades significantly under noisy environments.

Figure 1 illustrates a typical ASR system. Following pre-processing, which includes speech/non-speech detection and filtering, feature extraction is performed on the post-processed speech signal to form a compact representation. Desirable speech features should emphasize linguistic information over extraneous content such as the speaker's age, emotion, gender, etc. The most commonly used features in speech recognition systems are Mel-frequency cepstral coefficients (MFCCs) (Davis and Mermelstein, 1980). The extraction process involves segmenting the speech signal into quasi-stationary short-time frames of 20–40 ms, overlapped every 10 ms (i.e., frame-rate of 100 fps). For each frame, a Mel-scale filter bank is applied to its power spectrum estimate. The MFCCs are defined as the discrete cosine transform (DCT) of the log energies in the corresponding frequency bands. They measure the power spectrum envelope in each frame, which correlates to the shape of the vocal tract, providing an appropriate representation of the sound or phone being produced.

At the heart of an ASR system is the decoder. Feature vectors are decoded into linguistic units that make up speech, using acoustic models learned from recordings and their corresponding transcripts. Linguistic and pronunciation knowledge are often used to improve the decoding performance (Kuhn and Mori, 1990; Bengio et al., 2003; Mikolov et al., 2010). The standard approach to tackle ASR is to impose a statistical framework by scoring each speech signal with words in a vocabulary on a probability scale, with the most likely word selected as the ASR output. The hidden Markov model (HMM) was the most widely used acoustic model for speech recognition (Rabiner, 1989) until recent years and is still used for many practical applications. Under this statistical framework, the observations or speech feature vectors are modeled as acoustic signals generated by a stationary process, while the transition probabilities in the hidden states account for the time-varying nature of speech. Current advances in accuracy achieved with deep learning (DL) (Hinton et al., 2012) are mismatched with mobile devices and resource-constrained systems, due to difficulty of training, power, and footprint requirements. Conventionally, these applications utilize cloud-based solutions, where processing is performed on large remote servers. However, this imposes additional demands on quality of service. There are many mobile applications where the on-device acoustic model output accuracy is insufficient.

Figure 2 shows a typical discrete HMM, parametrized by an initial state distribution $\pi = \{\pi_i = \Pr(S_1 = s_i)\}$, a state transition probability matrix $\mathbf{A} = \{a_{i,j} = \Pr(S_t = s_j | S_{t-1} = s_i)\}$, an observation distribution $\mathbf{B} = \{b_i(\boldsymbol{u}_t) = \Pr(\boldsymbol{u}_t | S_t = s_i)\}$, where $\mathcal{U} = \{\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_f\}$ is an f-frame observation sequence, and $S = \{s_1, s_2, \cdots, s_L\}$ is the underlying state sequence of length L, which forms a first-order Markov chain. The Gaussian mixture model (GMM) is typically used to approximate the observation distribution $\mathbf{B}$. An HMM $(\pi, \mathbf{A}, \mathbf{B})$ can be estimated using the Baum-Welch (BW) algorithm (Baum et al., 1970), a special case of the expectation-maximization (EM) algorithm (Dempster et al., 1977). In ASR, one HMM is trained for each speech unit (e.g., phone, syllable, word, etc.,) in the vocabulary. A test utterance is compared to all trained HMMs, in order to determine the likelihood that it was generated by a particular HMM. This framework represents an unsupervised learning paradigm. As a maximum-likelihood estimation (MLE) method, it relies on strong assumptions on the statistical properties of the observed phenomenon, but lacks discriminative power among different models.

Since humans naturally and very efficiently decode speech and perform better than most ASR systems, especially in noisy environments, it is only logical for researchers to turn to biological inspiration in the design of ASR systems. As a matter of fact, MFCC already makes use of the psychoacoustic properties of the auditory system (the Mel scale imitates the cochlea by employing linearly and logarithmically distributed filters along the frequency axis, with the cutoff at 1 kHz), a fine tuned preprocessing step in the human auditory system. The pressure waves originating from the cochlea are translated into spike trains by the peripheral auditory neurons, which travel through nerve fibers to the auditory cortex. The computation in this complex and hierarchical structure is carried out via action potential timing information. Computing with spikes is therefore an important aspect to bio-inspired ASR.

There has been limited research in spike train representation for spoken word recognition (Hopfield and Brody, 2001; Verstraeten et al., 2005; Wade et al., 2010; Zhang et al., 2015). The state-of-the-art spike-based ASR systems are based on spiking neural network (SNN) such as liquid state machines (LSMs) (Maass et al., 2002). LSM utilizes a large randomly initialized network with recurrent connections, also referred to as a dynamic reservoir or liquid. The parameters of the liquid remain fixed, and only a readout layer is adapted through training to optimally project the network or liquid states onto the desired output. The LSM falls under a general framework called reservoir computing (RC), which is further identified as an echo state network (ESN) (Jaeger, 2001) for continuous valued inputs and LSM for spike train inputs. The primary advantage of the LSM approach is that it does not require consideration for time dependency of the learning task, since all temporal processing is performed implicitly in the recurrent neural circuit. RC is free from the problems associated with gradient-based recurrent neural networks training such as local optima, slow convergence, and high computational complexity. However, performance depends largely on the reservoir hyperparameters that need to be cross-validated appropriately to find an optimal solution, without which RC is a less reliable convex universal learning machine (CULM) than conventional adaptive networks using kernel adaptive filtering (Príncipe and Chen, 2015). Furthermore, producing a constant output for time-varying liquid state is a major challenge for LSM, since its memory-less readout has to transform the transient and non-stationary states of the liquid into a stable output without the assistance of stable states or attractors (Maass et al., 2002).

In our previous work (Li and Príncipe, 2016), we introduced a novel online kernel adaptive filtering algorithm: the kernel

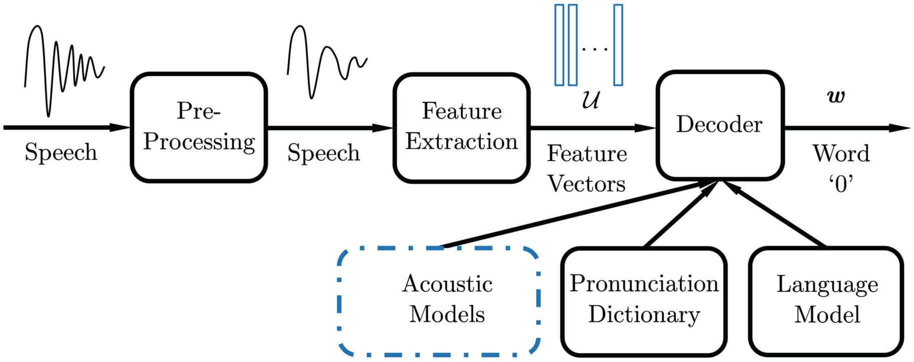


FIGURE 1 | Automatic speech recognition system diagram.

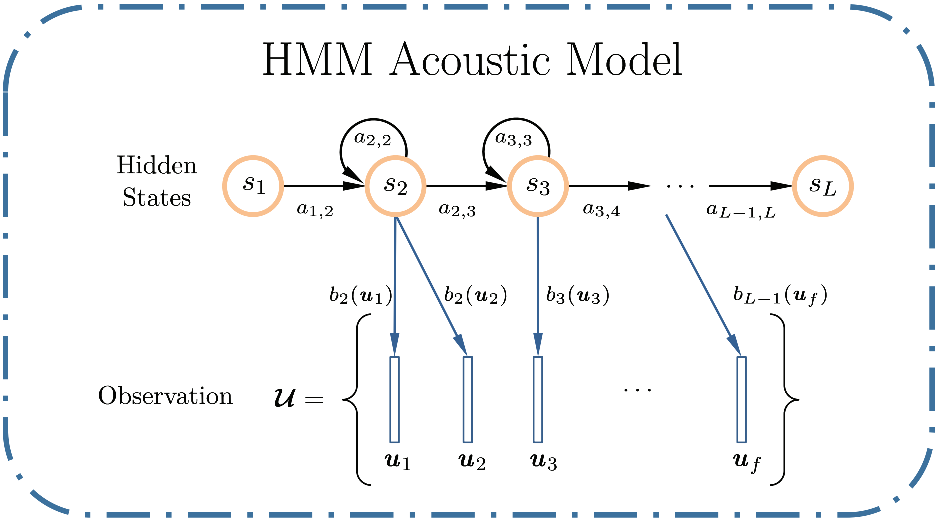


FIGURE 2 | Example of an *L*-state left-to-right discrete HMM used for ASR, with two non-emitting states: s_1 and s_L. For each emitting state, the HMM can only remain in the same state or move to the next state on its right.

adaptive autoregressive-moving-average (ARMA) or KAARMA. We demonstrated this kernelized recurrent network's ability to model dynamical systems and as a bit-stream classifier using the benchmark Tomita grammars. Specifically, we showed that KAARMA-based solutions can outperform LSMs on spike data, which opened the door for many novel neuroscience applications (Dura-Bernal et al., 2016). Furthermore, we have successfully applied the methods to model flight dynamics of insects and plant growth patterns (Li and Príncipe, 2017a,b; Li et al., 2017). Since speech production is both nonlinear and non-stationary in nature, KAARMA can deliver computationally efficient solutions for ASR as we demonstrate below.

In this paper, we propose a novel paradigm to work with spike timing information. Instead of projecting the input spike train nonlinearly into a much higher dimensional space using a recurrent interconnection of spiking neurons as is done with LSM, we project the input spike trains into an infinite dimensional function space (RKHS) using positive definite functions, where we train a linear state-space model with a very small order using backpropagation and the kernel trick. The theory of adaptive signal processing is greatly enhanced through the integration with the theory of RKHS. By performing classical linear methods in an infinite-dimensional feature space, online kernel learning (Kivinen et al., 2004), such as kernel-Adaline (Frieß and Harrison, 1999), kernel recursive least-squares (KRLS) (Engel et al., 2004), kernel least mean square (Liu et al., 2008), and extended-KRLS (Liu et al., 2009) algorithms provide general nonlinear solutions in the original input space. It also gives rise to kernel Kalman implementations, such as using subspace kernel principal component analysis (Ralaivola and d'Alche Buc, 2005) and statistical embedding (Zhu et al., 2014) to model nonlinear dynamics.

A major advantage of the KAARMA algorithm is that it works with functions in the RKHS and changing the kernel function does not impact the underlying learning algorithm. Therefore, KAARMA is agnostic to the type of input and can be applied to static data using Gaussian kernels, or directly to spike trains, by designing an appropriate spike kernel (Park et al., 2012, 2013). In this paper we use a biologically-inspired auditory filterbank and a LIF neuron model to convert the continuous-amplitude signal output from each channel of the filterbank into a sparse spike train representation, to create a multichannel spike train, encoding the signal-structure changes in each frequency band. The spike trains are then segmented using a sliding window into frames of fixed duration and frame rate or stride, similar to conventional speech processing. A special designed temporal kernel then maps the spike-train frames to an RKHS by estimating the distance between successive frames of the spike trains, using their precise spike timings. Inference is performed not on individual frames, but on sequences of spike-train frames without assumption on the sequence length. Nonlinear ARMA networks have theoretical capability to model dynamics of arbitrary complexity. This methodology suggests a novel way to apply spike-based computation using a recurrent neural coding algorithm in RKHS as an alternative for a biologically-inspired robust ASR system. Without any feature engineering step, we evaluate how well this spike-based KAARMA ASR performs compared to conventional amplitude-based MFCC-KAARMA and other SNN solutions. We also evaluate the inherent noise-robustness of the spike-train sparse representation, due to the smoothing effect of the integration operation in the LIF neuronal model.

The rest of this paper is organized as follows. In section 2, we briefly introduce the KAARMA methodology. We present its application for bio-inspired spike-based ASR in section 3. Performances of the proposed KAARMA classifiers are evaluated in section 4. Section 5 concludes this paper.

2. METHODS

We would like to model and learn the temporal evolution of speech time-series acoustical features' structure encoded in spike trains. The goal, here, is a bio-inspired ASR system where as much of the traditional speech pipeline as possible is replaced by a recurrent network architecture. Specifically, we wish to evaluate an end-to-end spike-based keyword spotting system, without hand-designed feature extraction algorithm, past the spike-generation stage. Furthermore, we wish to use a unifying framework that does not depend on input signal type. For example, conventional artificial neural network and SNN have completely different output and learning mechanisms due to the non-differentiable activation functions associated with discrete spikes. To accomplish this, we apply the theory of RKHS to map the inputs into a function space and construct a recurrent network in this space. This way, the learning algorithm is defined not in terms of the input representation (continuous-valued attributes vs. discrete spikes), but in terms of dot products between respective infinite-dimensional features, where they can be computed in closed form using the kernel trick. Thus, we are free to choose the input representation independently with an appropriate reproducing kernel, and changing the input-kernel pair does not impact the learning algorithm itself. An additional drawback of conventional speech pipeline is alignment, specifically frame-level training targets. We can resolve all the issues mentioned by modeling speech as a dynamical system and treating isolated word recognition as a grammatical inference task trained on sequences and not on individual frames, using the kernel adaptive ARMA algorithm.

2.1. Kernel Adaptive ARMA Algorithm

Here, we briefly introduce the KAARMA algorithm for isolated-word speech recognition or keyword spotting, while the adaptation of parameters is presented in the Appendix (see Supplementary Material) for completeness. For a more in-depth derivation, please refer to Li and Príncipe (2016).

A dynamical system approach studies the evolution of observables over time according to specific rules. We can trace it to a classical Newtonian root: the forces are much simpler to describe than planetary motions. Under this framework, even seemingly-chaotic time series actually follow an easy to explain hidden order, and a dynamical model allows us to find such attracting behavior. Rule discovery provides a compact and convenient way to analyze and model a class of equivalent trajectories but with large variations in realization.

First, let us define a dynamical system using a state-space representation with a general continuous nonlinear state-transition function $\mathbf{g}(\cdot,\cdot)$ and an observation function $\mathbf{h}(\cdot)$:

$$\mathbf{x}_i = \mathbf{g}(\mathbf{s}_{i-1}, \mathbf{u}_i), \tag{1}$$

$$\mathbf{y}_i = \mathbf{h}(\mathbf{x}_i) \overset{\Delta}{=} \mathbf{h} \circ \mathbf{g}(\mathbf{s}_{i-1}, \mathbf{u}_i), \tag{2}$$

with input vector $\mathbf{u}_i \in \mathbb{R}^{n_u}$, hidden state vector $\mathbf{x}_i \in \mathbb{R}^{n_x}$, output vector $\mathbf{y}_i \in \mathbb{R}^{n_y}$, the augmented state vector $\mathbf{s}_i \overset{\Delta}{=} [\mathbf{x}_i, \mathbf{y}_i]^T$, and the function composition operator $\circ$. For our application, the state-transition function $\mathbf{g}(\cdot,\cdot)$ describes the dynamics driven by the input speech $\mathbf{u}_i$ and the previous state (for isolated word, all speech sequences are assumed to have the same initial state). The sequence output $\mathbf{y}_i$ is related to the states and inputs by observation function $\mathbf{h}(\cdot)$.

Using a grammatical-inference formulation, the only thing we know during training are labels for the full sequences or speech utterances, i.e., the final sequence output $\mathbf{y}_f = \{\pm 1\}$ for positive or negative examples of a target class or word model. The state and transition functions can be parametrized with weight values of a fully connected recurrent network and learned using backpropagation of the label error at the end of each speech sequence. This task is an inference problem as opposed to a prediction one, i.e., a sequence-based approach vs. the conventional frame-based approach of an HMM. There is no prediction of the next frame of speech in the utterance sequence. The network either accept or reject an entire utterance at the end of each sequence. This is a more difficult problem than prediction, since we do not have complete classification knowledge of every subsequence (i.e., when prediction and inference are equivalent). On the other hand, it does not require a frame-level target or alignment, i.e., a desired signal $\mathbf{d}_i$ is not required at each time/frame index of output $\mathbf{y}_i$, only for the final index $\mathbf{y}_f$; the internal state trajectories $\mathbf{s}_i$ are also learned directly from the training sequences (given a fixed initialized state) without any observables except at the end of the sequence when $\mathbf{y}_f = \{\pm 1\}$ for $\mathbf{s}_f = [\mathbf{x}_f, \mathbf{y}_f]^T$; and, this dynamical model makes no assumption on the speech utterance duration or sequence length f, i.e., it can operate on sequences of arbitrary length.

Adaptation of parameters in the linear state model is very well understood, and the famed Kalman filter (Kalman, 1960) presents a very efficient recursive update algorithm that can be computed in real time. The problem of the linear state model is that it is not universal, i.e., it only can solve problems with small error when the desired response exists in the span of the input space (Haykin, 1998). Past work with dynamical modeling of speech shows that the linear dynamical model is not competitive with the HMM statistical model. The theory of RKHS allows classical linear method to produce general nonlinear solutions, and by operating in a new, function space, we are freed from the limitations of the original input representation/space.

To emphasize the input-agnostic property of a function-space formulation for applications using either continuous-valued input or discrete-time events, we first describe the KAARMA algorithm using a generic input sequence $\mathbf{u}_i$, then specify it for spikes in section 2.2, which basically amounts to a simple substitution on the kernel choice. Using the representer theorem, we can express the state-space model Equation (1-2) as a set of weights (functions in the input space) in the joint RKHS $\mathcal{H}_{su} \triangleq \mathcal{H}_s \otimes \mathcal{H}_u$

$$\mathbf{\Omega} \triangleq \mathbf{\Omega}_{\mathcal{H}_{su}} \triangleq \begin{bmatrix} \mathbf{g}(\cdot,\cdot) \\ \mathbf{h} \circ \mathbf{g}(\cdot,\cdot) \end{bmatrix}, \tag{3}$$

where $\otimes$ is the tensor-product operator. Finally, the kernelized state-space model becomes

$$\mathbf{s}_i = \mathbf{\Omega}^T \psi(\mathbf{s}_{i-1}, \mathbf{u}_i), \tag{4}$$

$$\mathbf{y}_i = \mathbb{I}\mathbf{s}_i, \tag{5}$$

where $\psi(\mathbf{s}_{i-1}, \mathbf{u}_i) \triangleq \varphi(\mathbf{s}_{i-1}) \otimes \phi(\mathbf{u}_i)$ is a feature in the joint RKHS and $\mathbb{I} \triangleq \begin{bmatrix} \mathbf{0} & \mathbf{I}_{n_y} \end{bmatrix}$ is a fixed selector matrix with $\mathbf{I}_{n_y}$ is an $n_y \times n_y$ identity matrix, used to extract the output components $\mathbf{y}$ from the augmented state vector $\mathbf{s}$. This is analogous to a second-order recurrent neural network defined in a function space in our previous work (Li and Príncipe, 2016).

It follows that the tensor-product kernel is defined as

$$\begin{aligned} \langle \psi(\mathbf{s},\mathbf{u}), \psi(\mathbf{s}',\mathbf{u}') \rangle_{\mathcal{H}_{su}} = \mathcal{K}_{su}(\mathbf{s},\mathbf{u},\mathbf{s}',\mathbf{u}') &= (\mathcal{K}_s \otimes \mathcal{K}_u)(\mathbf{s},\mathbf{u},\mathbf{s}',\mathbf{u}') \\ &= \mathcal{K}_s(\mathbf{s},\mathbf{s}') \cdot \mathcal{K}_u(\mathbf{u},\mathbf{u}'). \end{aligned} \tag{6}$$

This construction has several advantages over the simple concatenation of the input $\mathbf{u}$ and the state $\mathbf{s}$. First, the product of two positive-definite (PD) kernels is also a PD kernel. Second, since learning is performed in an RKHS using features, there is no constraint on the original input signal representation or the number of signals, as long as we use an appropriate reproducing kernel for each signal. Additionally, the sum or average of two PD kernels is also a PD kernel for multi-channel input. More importantly, this formulation imposes no restriction on the relationship between the signals in the original input space. This is especially useful for input signals having different representations and spatiotemporal scales. Specifically, under this framework, we can model a neurobiological system, taking continuous-amplitude local field potentials, discrete-events-in-continuous-time spike trains, and vectorized state variables as inputs.

Figure 3 shows a graphical interpretation of a dynamical system defined in a joint RKHS using a product kernel. Data instances are processed using inner products or similarity measures. The tensor-product kernel is analogous to a soft-valued logical AND operator on the joint similarity measure. To output a desired next state requires both an appropriate current input AND the right previous state. In general, the states $\mathbf{s}_i$ are assumed hidden, and during training, the desired signal does not need to be available at every time step, e.g., a deferred desired output value (± 1 sequence label vector) for $\mathbf{y}_i$ may only be observed at the final indexed step $i = f$.

The KAARMA preserves the simplicity of linear dynamical models with the universality of functional spaces, so it is an attractive candidate to substitute linear dynamical systems in computational neuroscience applications using either local field potentials or spike trains. In computational neuroscience there is a chasm between the methodologies for spike trains and continuous amplitude signals that can be easily bridged with RKHS methodologies. Indeed the same machine learning code can be utilized for both types of signals, once specific kernel are designed for each signal modality. The application for speech recognition exemplifies a statistical learning approach to work with spike trains, which improves the biorealism of the processing and lets us take advantage of the spike timing information.

The fundamental building block for designing the KAAMA for spike trains is therefore the kernel, which will be explained next.

2.2. Reproducing Kernel Hilbert Space (RKHS) for Spike Trains

We want to study how information is represented and processed as spike trains using the theory of RKHS. Since spike trains are devoided of a natural algebra, they impose many challenges to signal processing methods. We must first establish a space for computation or transformation to a space with the necessary properties. The approach explained here is to define a proper kernel function on spike trains to capture non-parametrically the instantaneous temporal structure and the variability of the spike trains of interest. Once a positive-definite kernel is defined, it maps the spike trains into a Hilbert space of functions which allows signal processing tools to be applied directly through the kernel trick, as shown in **Figure 4**.

We use the Schoenberg kernel (Park et al., 2012), a universal binless nonlinear spike train kernel, to define the joint tensor-product RKHS. This kernel is bio-inspired using conditional intensity function of a temporal point process. Among spike train

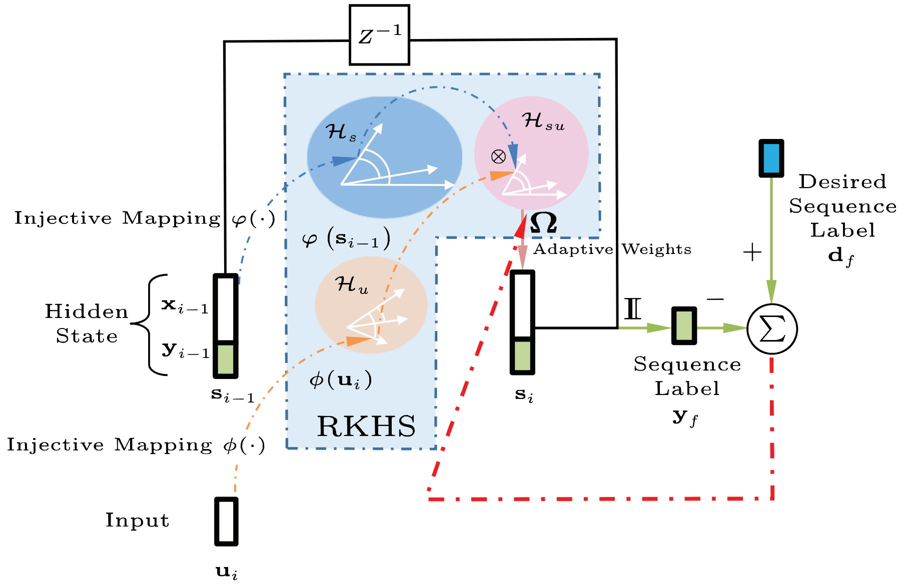
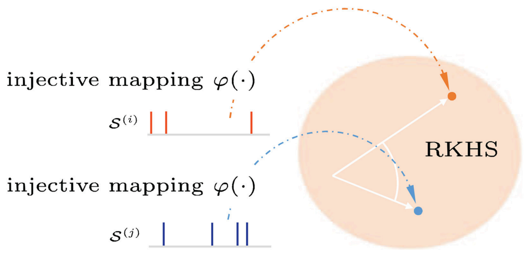


FIGURE 3 | Block diagram of the kernel adaptive ARMA (KAARMA) algorithm. The values of the adaptive weights $\boldsymbol{\Omega}$ in the feature space are learned using backpropagation and the kernel trick. In general, the states $\mathbf{s}_i$ are assumed hidden, and during training, the desired value for label y_i is only observed at the end of the sequence, i.e., at the final indexed time step $i = f$.


injective mapping $\varphi(\cdot)$
$\mathcal{S}^{(i)}$
RKHS
injective mapping $\varphi(\cdot)$
$\mathcal{S}^{(j)}$

$$\mathcal{K}_{a_\lambda}\left(\mathcal{S}^{(i)}, \mathcal{S}^{(j)}\right) = \left\langle \varphi\left(\mathcal{S}^{(i)}\right), \varphi\left(\mathcal{S}^{(j)}\right)\right\rangle$$

FIGURE 4 | Graphical interpretation of a reproducing kernel Hilbert space defined on spike trains. Spike trains with precise spike timings are mapped into an infinite-dimensional feature space (Hilbert space). Applying the kernel trick allows inner products in this space to be computed without explicit reference to the feature representation.

kernels [count and binned kernels, spikernel (Shpigelman et al., 2005), linear functional kernels (Paiva et al., 2009), and nonlinear functional kernels (Park et al., 2012)], the Schoenberg kernel has three distinct advantages: (1) provides injective mapping, (2) embeds arbitrary stochasticity of neural responses as the sample mean in the RKHS, and (3) approximates arbitrary function on spike trains as a universal kernel (Park et al., 2013).

A spike train or sequence of M ordered spike times, i.e., $\mathcal{S}^{(i)} = \{t_m \in \mathcal{T} : m = 1, \cdots, M\}$ in the interval $\mathcal{T} = [0, T]$, can be viewed as a realization of an underlying stochastic point process with conditional intensity function $\lambda(t|H_t^{(i)})$, where $t \in \mathcal{T} = [0, T]$ denotes the time coordinate, and $H_t^{(i)}$ is the history of the process up to time t. The point process is approximated as a zero-baseline-rate Hawkes process (Hawkes, 1971). Schoenberg kernel between the conditional intensity functions of two point processes (Paiva et al., 2009; Park et al., 2012; Dura-Bernal et al., 2016) is defined as

$$\mathcal{K}_{a_\lambda}(\lambda(t|H_t^{(i)}), \lambda(t|H_t^{(j)})) \triangleq \exp\left(-a_\lambda \int_\tau (\lambda(t|H_t^{(i)}) - \lambda(t|H_t^{(j)}))^2 dt\right), \tag{7}$$

where $a_\lambda > 0$ is the spike-train kernel parameter. The conditional intensity function of the self-exciting point process with zero background rate is approximated by convolving the precise spike times with a smoothing kernel $g(t)$, yielding

$$\hat{\lambda}(t) = \sum_{m=1}^{M} g(t - t_m), \{t_m \in \mathcal{T} : m = 1, \cdots, M\}. \tag{8}$$

It computes the similarity between a pair of spike trains in $\mathcal{T}$, either from a single neuron at different times or from a pair of neurons. In this application, instead of two spike trains from different frequency bands, we are interested in quantifying the time-series structure or difference in conditional intensity functions across time of the same spike channel. For computational simplicity, we use the rectangular function $g(t) = \frac{1}{\mathcal{T}}\left(U(t) - U(t - \mathcal{T})\right)$, where $U(t)$ is a Heaviside function and $\mathcal{T}$ is chosen to be much greater than the average inter-spike interval. Since we are interested in time-binned or frame-based raw spike events, $\mathcal{T}$ is effectively set to the frame duration. **Figure 5** illustrates this squared distance between the conditional intensity function estimates of two spike-train frames $\mathcal{S}^{(i)}$ and

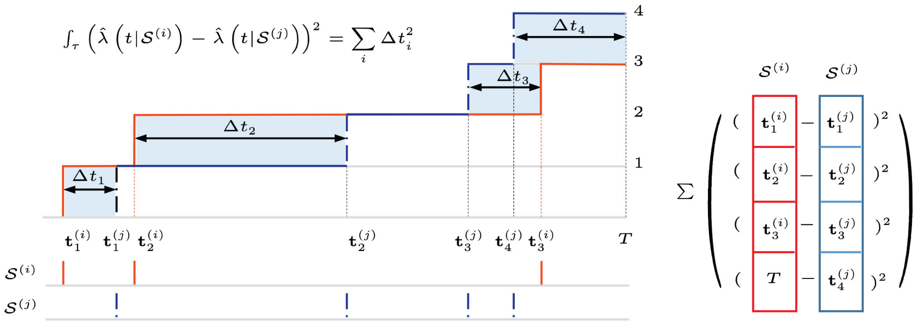

$\int_{\tau}\left(\hat{\lambda}\left(t|\mathcal{S}^{(i)}\right)-\hat{\lambda}\left(t|\mathcal{S}^{(j)}\right)\right)^2=\sum_i \Delta t_i^2$
Δt_1
Δt_2
Δt_3
Δt_4
4
3
2
1
$\mathbf{t}_1^{(i)}$
$\mathbf{t}_1^{(j)}$
$\mathbf{t}_2^{(i)}$
$\mathbf{t}_2^{(j)}$
$\mathbf{t}_3^{(j)}$
$\mathbf{t}_4^{(j)}$
$\mathbf{t}_3^{(i)}$
T
$\mathcal{S}^{(i)}$
$\mathcal{S}^{(j)}$
$\sum\left((\mathbf{t}_1^{(i)}-\mathbf{t}_1^{(j)})^2 \; (\mathbf{t}_2^{(i)}-\mathbf{t}_2^{(j)})^2 \; (\mathbf{t}_3^{(i)}-\mathbf{t}_3^{(j)})^2 \; (T-\mathbf{t}_4^{(j)})^2\right)$

FIGURE 5 | The Schoenberg spike kernel computes the similarity between a pair of spike trains. In this application, we compare the conditional intensity function estimates for spike-train frames $\mathcal{S}^{(i)}$ and $\mathcal{S}^{(j)}$ at different times in a given frequency band or channel. Using Heaviside step function for smoothing greatly simplifies the computation. We can visualize it as a sum of squared pair-wise spike-timing differences between two unit-step staircase functions (squared areas in blue) or as squared Euclidean distance on ordered sets of spike timings, with the fewer-spike set padded with frame duration time T. For multichannel spike input, the sum or average distance is used.

$\mathcal{S}^{(j)}$ at different times for a given frequency band or channel, i.e., the integral in Equation (7) using Equation (8). In this formulation, the spike-train distance only depends on the precise spike timings in ordered sets. When two spike trains are "close," more of their spike timings are synchronized, yielding a smaller pair-wise distance.

For multichannel spike input, we sum or average the spike-train distances over all channels in each time frame. Specifically, the multichannel spike trains are segmented into frames or smaller spike trains the same way as the MFCCs, with a frame duration of 25 ms and rate of 100 fps. **Figure 6** illustrates a KAARMA network working directly on spike trains.

2.3. Comparisons Between Spike-Based Kernel Approach vs. LSM

The LSM and the KAARMA are both adaptive recurrent models that operate with spike trains, but the similarity ends here.

The LSM uses a recurrent layer of spiking neurons, designed by a user, to project the input spike data into a high dimensional space, where it will be easier to find a learned projection that fulfills the data processing goal. Clearly, not all projections to high dimensional spaces will preserve the information contained in the input spike train, therefore, the designer must select a hyperparameter that achieves the prescribed separation property or SP (Maass et al., 2002). SP is quantified by a kernel-quality measure proposed in Maass et al. (2005) that is based on the rank of a matrix formed by the system states corresponding to different input signals (Bertschinger and Natschläger, 2004). Therefore, SP is signal and application dependent, which means that creating the optimal liquid is still today more of an art than a science. The advantage of the LSM is that it uses directly the instantaneous intensity function of the spike trains because it is a dynamical system.

The KAARMA handles the processing of spike trains in a very different way. First, the spike trains are projected to an infinite dimensional space of functions (RKHS) with the Schoenberg kernel using the instantaneous conditional intensity function estimated on an interval. Linear models in RKHS are universal mappers, i.e., they can approximate any input-output map. In this space, one can train a linear state model directly from data to learn the spike train structure and deliver a high quality mapping with very small model orders, using directly the input data (the representer theorem). So instead of a high dimensional and usually randomly created and fixed reservoir that an LSM uses, the KAARMA uses the functions in the Hilbert space centered by the projected input spike trains. This RKHS is based entirely on the available data samples with optimized adaptive weights. The spike kernel still operates with instantaneous information but now in the conditional intensity function of learned data, which is a suitable approximation to the intensity function, but requires the selection of a hyperparameter.

3. AUTOMATIC SPEECH RECOGNITION SYSTEM USING KAARMA

We can treat certain speech recognition tasks as grammatical inference problems and apply the KAARMA algorithm to learn temporal structures of speech features with arbitrary length, analogous to syntactic pattern recognition involving the Tomita grammars (Li and Príncipe, 2016). As a recurrent network, the KAARMA algorithm exploits the full contextual information of the entire feature sequence to create a discriminative model. It makes no assumption on the model topology of the data, and the states are learned completely from the observations.

Many spoken words share similar or identical acoustic features. Given the large variations in speech production, common trailing phoneme can be difficult for recurrent systems learning long-term dependencies, where long-drawn-out overlapping ending sequences can cause two different word models to converge. One simple way to circumvent this problem, without significant change to the experiment, is to simply reverse the temporal order of the acoustic features, such that the trailing sequences no longer overlap, and train a KAARMA classifier that recognizes this new input ordering. Digits that used to share the same trailing phoneme may end up in different ones (of course the opposite can also happen). To maximize recognition rate for

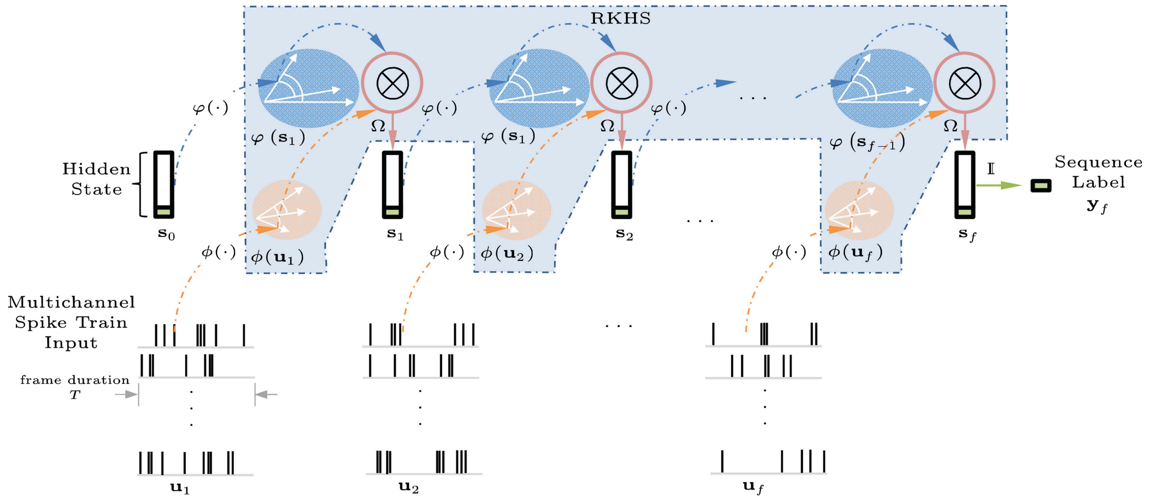


FIGURE 6 | Spike-input KAARMA network unfolded in time for *f* frames. The multichannel spike-train input frames are mapped into a joint RKHS with the current hidden state vector using a tensor-product kernel to generate the next state vector. The final state vector at frame *f* contains the prediction label for the entire sequence.

each digit, we can combine the results of two networks trained on sequences in the natural left-to-right temporal direction and the reversed right-to-left ordering, by simply multiplying their softmax scores. Flipping the sequence ordering generates a new complementary grammar that can be combined to enhance classification results. This is a feature that is entirely missing in HMMs, due to the Markov property that states are formed locally and only operate on adjacent observation vectors. States in a recurrent network, on the other hand, are memory units which encode the entire past history, starting from an initial state, and indicate a global status. To further reduce the need to learn long-term dependencies and to simplify computation, we can partition a speech feature sequence into smaller segments, without the need for complicated alignment, which we discuss in detail next.

3.1. KAARMA Chain

Here we formulate the KAARMA chain approach for isolated word recognition under a simple statistical framework. First, let us revisit the conventional HMM in **Figure 2**. In the hidden Markov model, speech signal, specifically, the sequence of acoustic feature vectors $\boldsymbol{\mathcal{U}} = \{\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_f\}$ is generated by a finite state automaton consists of L states $S = \{s_1, s_2, \cdots, s_L\}$ under a probabilistic framework. An HMM is equivalent to a stochastic regular grammar (Lari and Young, 1990). Each speech unit is associated with a specific Markov model M_i comprised of states from S according to a predefined topology. The left-to-right (Bakis) model is the most commonly used topology for speech recognition (Bakis, 1976). States are aligned from left to right to form a single Markov chain, indexed incrementally and with only self- or right-transitions allowed, i.e., $a_{i,j} = 0$, for $j < i$. Furthermore, the initial state is fixed at state s_1. Left-to-right HMMs are able to model the temporal properties of speech.

The training and recognition criteria for HMMs are based on maximizing the *a posteriori* probability $\Pr(M_i|\boldsymbol{\mathcal{U}})$ that the observation $\mathcal{U}$ has been produced by the HMM M_i. Using Bayes' rule, we can rewrite the expression as

$$\Pr(M_i|\boldsymbol{\mathcal{U}}) = \frac{\Pr(\boldsymbol{\mathcal{U}}|M_i)\Pr(M_i)}{\Pr(\boldsymbol{\mathcal{U}})}, \tag{9}$$

where $\Pr(\boldsymbol{\mathcal{U}}|M_i)$ is the maximum likelihood estimate (MLE) criterion, $\Pr(\boldsymbol{\mathcal{U}})$ is constant during recognition, and the *a priori* probability $\Pr(M_i)$ is an appropriate language model.

The BW algorithm can be used to maximize the likelihood estimate of the parameters of a HMM, given the set of observed feature vectors. Alternatively, the MLE can be replaced by the Viterbi criterion, where only the most probable state sequence of producing $\boldsymbol{\mathcal{U}}$ is considered

$$\hat{\Pr}(\boldsymbol{\mathcal{U}}|M_i) = \max_{\mathcal{S}} \Pr(\mathcal{S}, \boldsymbol{\mathcal{U}}|M_i), \tag{10}$$

and the optimal S^* is given by

$$\mathcal{S}^* = \arg\max_{\mathcal{S}} \prod_{\ell=1}^{L} \Pr(s_\ell|s_{\ell-1})\Pr(\boldsymbol{u}_\ell|s_\ell), \tag{11}$$

which can be solved using the Viterbi algorithm (Viterbi, 1967). This frame-based approach is fundamentally different from our novel sequence-based approach which requires no alignment or frame-level target for isolated word recognition.

Under a hybrid ANN-HMM paradigm, connectionist statistical methods (Franzini et al., 1990; Levin, 1990; Morgan and Bourland, 1990; Niles and Silverman, 1990; Robinson, 1994) were proposed as improvements to the standard HMM. It is well-established that the outputs of a multilayer perceptron (MLP) operating in classification mode can be interpreted as

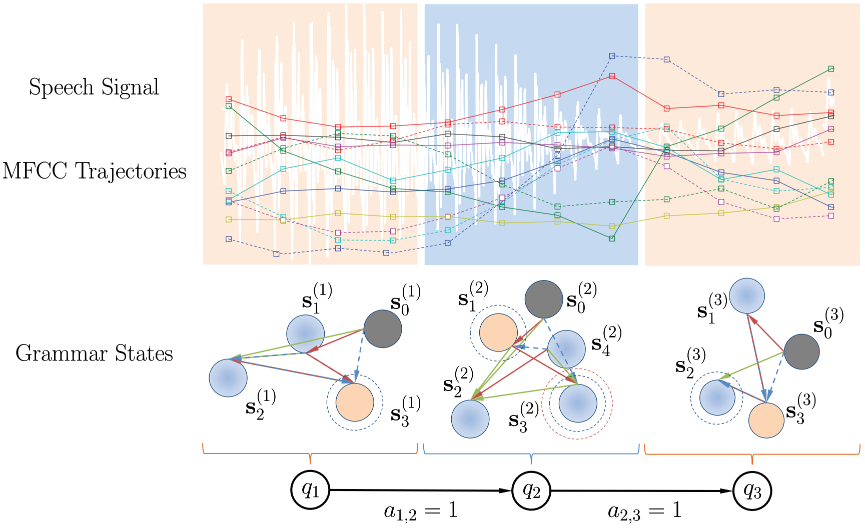


FIGURE 7 | Example of a KAARMA chain of three equal-partition grammar states.

estimates of the local *a posteriori* probabilities of output classes conditioned on the input (Bourlard and Wellekens, 1990)

$$y_j^*(\boldsymbol{u}_i) = p(s_j|\boldsymbol{u}_i), \tag{12}$$

where y_j^* is the optimal (MLP with sufficient parameters and no local minimum) classification output value for state s_j. In the hybrid approach the *a posteriori* probabilities are converted into the HMM emission probabilities $p(\boldsymbol{u}_i|s_j)$ by dividing the MLP output by the prior class probabilities. To provide context information, $2c+1$ frames were used at the input (where c is the context window parameter, with the current input frame centered in the middle) of the MLPs in Boulard and Morgan (1993), and RNNs were used in Robinson (1994).

A mixture-of-experts ESN architecture with a winner-take-all update strategy exhibited superior noise-robustness than HMM (Skowronski and Harris, 2007) for continuous-valued human factor cepstral coefficients (HFCC) (Skowronski and Harris, 2004). Multiple readout filters are grouped together to form a state (paralleling the Gaussian mixture of a Bakis HMM state), and test utterances were classified as the word model with the lowest mean-squared prediction error (MSE) along the Viterbi path for each model. Context features were used (first- and second-order temporal derivatives over ±4 frames), along with the log energy of each frame. Our approach, on the other hand, learns the contextual information directly from the input stream, without being hard-coded at each time step (a 12-dimension vector vs. the 39-dimension speech feature of the ESN), and the internal states are integrated under a unifying framework. The KAARMA recognition results are also directly obtained, without the need for Viterbi computation. Furthermore, while the ESN matched the baseline HMM performance for noise-free conditions, we will show that automatically learned recurrency can outperform HMM using the same inputs, for a computationally simpler implementation.

3.1.1. Grammar States

Instead of using universal approximators as local state emission probability estimators in the HMM framework, we can solve the statistical recognition criterion directly using the KAARMA algorithm. Recall that the MAP is defined as

$$M^* = \arg\max_M \Pr(M|\boldsymbol{\mathcal{U}}), \tag{13}$$

where M is the inference model, which is equivalent to maximizing the *a posterior* state sequence or most probable state sequence for each model.

Let us define the states in a KAARMA chain as context-free grammars, denoted by $\mathcal{Q} = \{q_1, q_2, \cdots, q_L\}$. This distinction is made to not confuse a grammar state q_i with the KAARMA internal hidden-state variables $\mathbf{s}_i$ (grammar state q is a discrete set and network hidden state $\mathbf{s}$ is a vector). Each grammar state q_i has its own set of unique internal hidden-states $\mathbf{s}^{(i)}$ that transition according to the rules learned directly from data, i.e., $q_i = \{\mathbf{s}_0^{(i)}, \mathbf{s}_1^{(i)}, \cdots, \mathbf{s}_{n_i-1}^{(i)}\}$. Under this formulation, a single KAARMA network (global grammar with $\mathcal{Q} = \{q_1\}$) trained on the entire observation trajectory $\boldsymbol{\mathcal{U}} = \{\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_f\}$ can be viewed as an HMM with only a single state, e.g.,

$$\tilde{y}_f^{(i)} = \frac{\exp(y_f^{(i)})}{\sum_{j=0}^{9} \exp(y_f^{(j)})} = \Pr(\mathcal{Q} = q = i|\boldsymbol{\mathcal{U}}), \tag{14}$$

where $y_f^{(i)}$ is the final output of a KAARMA network trained to recognize the grammar $q = i$ or classify the word "*i*." A softmax function is used to ensure that the posterior estimates are non-negative and sum to one. To improve the classification results, we can train several KAARMA networks that specialize in different ordered regions of a word in cascade, as in **Figure 7**. Since the utterances in the TI-46 digit corpus are not labeled by phoneme, without any frame-to-state alignment computation, we can simply fix the number of grammar states at L and partition naively the MFCC sequence for each isolated word into L equal segments. When necessary (e.g., total number of frames is less than L), the last MFCC vector is replicated to pad the partition. Each ordered segment is treated as a different grammar state, but given the same class label, and trained using a separate KAARMA network to learn its classification grammar, as shown in **Figure 7** (where $L = 3$).

Next, we fix the transition probability for grammar states q_i to q_j in a KAARMA chain at $a_{i,j} = 1$, for $j = i + 1$, and 0 otherwise. This is a major difference between a standard HMM and a KAARMA chain. The states in an HMM do not cope well with non-stationarity, thus during each Viterbi pass, frame-to-state alignment is performed such that each frame falls into the most likely quasi-stationary region or state in the temporal sequence, and the state transition probabilities are re-estimated. KAARMA and similar recurrent networks, on the other hand, are able to handle non-stationarities by leveraging their internal hidden states $\mathbf{s}_i$. One way to visualize these internal hidden states $\mathbf{s}_i$ in a grammar state q_i is to view the KAARMA chain as a nested HMM. But unlike the restricted structure of a traditional left-to-right model, the hidden state $\mathbf{s}_i$ in each grammar state are free to form transitions that best fit the available data, i.e., an ergodic model, as shown in **Figure 7**. Finally, in the KAARMA chain formulation, the recognized word is given by the following MAP criterion

$$M^* = \arg\max_M \prod_{i=1}^{L} \Pr(q_i = M|\boldsymbol{u}_{(f\cdot(i-1)/L)+1}^{f\cdot i/L}). \tag{15}$$

As discussed earlier, we can further improve the recognition rate by training a second KAARMA network for each grammar state, using the reversed-order feature sequences and multiplying the two softmax scores to derive a bi-directional probability score. By working on smaller segments of a speech signal, not only do we improve the training speed and reduce the need for the KAARMA algorithm to learn long-term dependencies, but also the latency needed for processing sequences of reversed order is shortened.

For real-valued speech features such as MFCCs, we can simply use a Gaussian kernel to apply the KAARMA algorithm for ASR. Next, we describe the appropriate steps for applying the KAARMA chain paradigm to a biologically-inspired ASR system. For each speech signal, biologically-plausible features are generated in the form of spike trains to mimic the front-end filtering performed by the human auditory system.

3.2. Spike-Based Speech Representation

Performing adaptive filtering in the RKHS has many advantages. One main merit being that the KAARMA model works with functions in the RKHS transformed by kernels and changing the kernel does not impact the KAARMA algorithm. Therefore, it is agnostic to the type of input and can be applied to any spatiotemporal signal, such as speech, by designing an appropriate kernel. By having separate formulations of the exogenous input vectors $\mathbf{u}$ and the internal state vectors $\mathbf{s}$, the KAARMA algorithm imposes no restriction on the relationship between the two signals in the original input space. We are free to choose the input representations independently as long as the appropriate reproducing kernels are selected. This enables us to work directly with non-numeric bio-inspired data such as spike trains, without modification of the underlying learning algorithm. The theory of RKHS allows signals of heterogeneous types to be operated under a unifying framework in a joint feature space, constructed using either direct sum or tensor-product reproducing kernels.

For our experiments, we combined a gammatone filterbank with a bank of spiking neuron models. First, a gammatone filterbank (Patterson et al., 1987) is applied to each acoustic signal. This formulation is motivated by the mechanical to electrical transduction in the cochlea (Meddis, 1986). Different regions of the basilar membrane vibrate to particular sound frequencies, in response to fluid flow in the cochlea. Sensory hair cells in the organ of Corti then convert the mechanical response to electrical signals which travel along the auditory nerve to the brain for processing. The gammatone filterbank simulates the mechanical response of the cochlea in which the output of each filter models the frequency response of the basilar membrane at a particular location, as shown in **Figure 8**. Its impulse response is defined in the time domain as

$$g(t) = a_g t^{n-1} e^{-2\pi bt} \cos(2\pi f_c t + \phi), \tag{16}$$

where f_c is the center frequency (in Hz), ϕ is the phase of the carrier (in radians), a_g is the amplitude, n is the filter order, b is the filter bandwidth (in Hz), and t indicates time (in s). The output of each gammatone filter is converted into spike trains using LIF neurons with spike-rate adaptation (SRA) and refractory current (Gerstner and Kistler, 2002), as shown in **Figure 9**.

The LIF neuron captures the basic spiking mechanism of nerve cells and is one of the simplest and most widely used model for spike processing in computational neuroscience. In this biological neuron model, the membrane capacitor C_m is charged by incoming current I until its potential V exceeds a certain threshold V_{th}, at which time it fires an action potential or spike, discharges, and resets the potential to a level V_{reset}. There are many variants of the model, based on various levels of realism, the one that we will use for this paper is determined by the following resistor-capacitor (RC) equation of the leaky integrator:

$$\tau_m \frac{dV}{dt} = (E_{rest} - V) + R_m I - E_{sra}, \tag{17}$$

where $\tau_m = R_m C_m$ is the membrane time constant, E_{rest} is the resting potential, R_m is the membrane resistance, I is the total current flowing into the cell, and instead of a fixed absolute refractory period, a reversal potential for SRA is used and defined as

$$E_{sra} \triangleq (V - E_k) R_m (g_{sra} + g_{ref}), \tag{18}$$

where E_k is the potassium reversal potential, g_{sra} and g_{ref} are the SRA and refractory conductances with time derivatives of

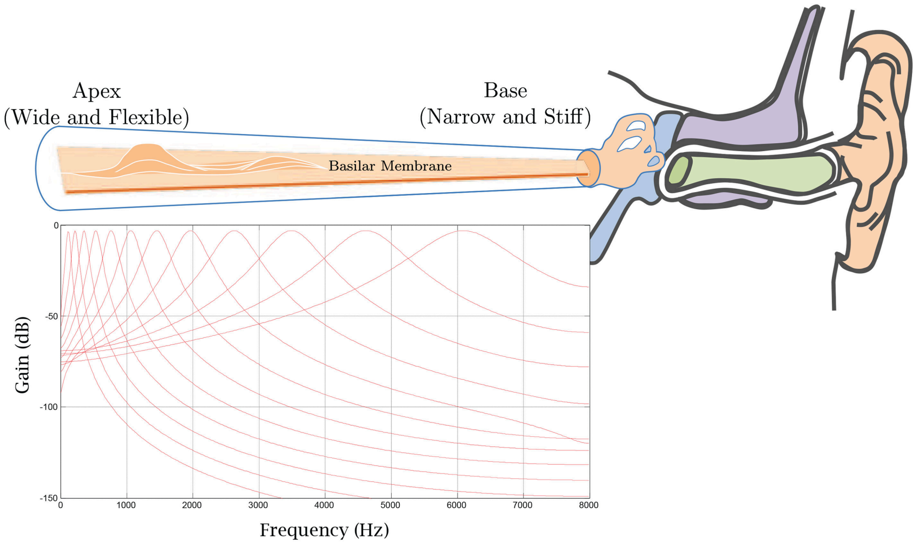


FIGURE 8 | A gammatone filterbank mimics the mechanical response of the cochlea in which the output of each filter models the frequency response of the basilar membrane at a particular location.

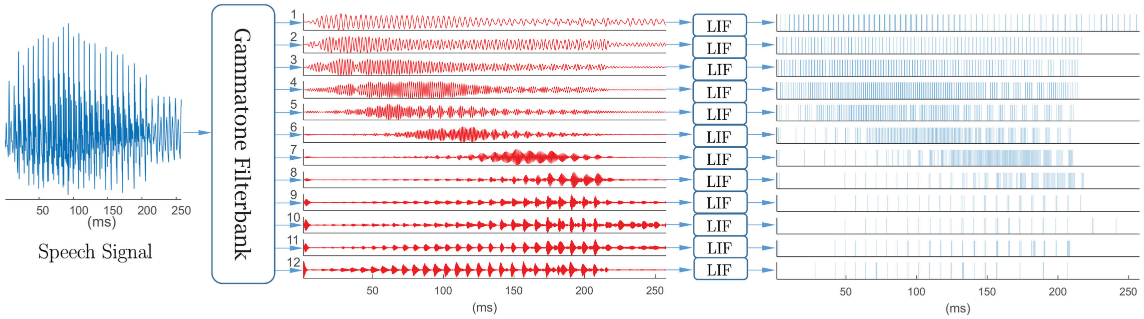


FIGURE 9 | Spike-based front-end for keyword spotting system. Speech signal first passes through a 12-channel output gammatone filterback, with center frequencies equally spaced between 50 Hz and 8 kHz on the ERB-rate scale, then converted into spike trains using leaky integrate-and-fire neurons. The mean spike count per frame (25 ms) ranged from 0.42 to 25.49 and varied across digits and channels.

$\dot{g}_{sra} = -g_{sra}/\tau_{sra}$ and $\dot{g}_{ref} = -g_{ref}/\tau_{ref}$, respectively. When membrane potential exceeds the spiking threshold or $V > V_{th}$, SRA and refractory conductances increment by Δ_{sra} and Δ_{ref} respectively, i.e., the two conductances increase at each spike and decrease exponentially between spikes. Initially, at $t = 0$, we set $V = E_{rest}$.

4. RESULTS

As a proof of concept, we used the TI-46 corpus of isolated digits to benchmark the KAARMA-based decoders in this paper. This corpus of speech consists of utterances from 16 English speakers (eight males and eight females) each speaking the digits “zero" through “nine" 26 times. Specifically, 25 out of the 26 utterances were used in the subsequent multispeaker experiments (i.e., our dataset comprises 4,000 of the 4,160 possible utterances). These utterances were further partitioned randomly into a training set (2,700 utterances with an equal number of male/female utterances and digits: 135 utterances per gender, per digit) and a testing set (1,300 utterances with an equal number of male/female utterances and digits: 65 utterances per gender, per digit). Furthermore, to reduce the number of non-speech data points used in the computation and to help align each utterance, speech signals were normalized with respect to their maximum absolute amplitudes, then automatically truncated into the smallest contiguous windows containing all non-silent regions, using a simple threshold-based endpoints detection algorithm.

Next, each truncated utterance was analyzed on 25 ms speech frames at 100 fps. For MFCC front-end, each frame was Hamming windowed, filtered by a first-order pre-emphasis filter ($\alpha = 0.95$). The magnitude spectrum from the discrete Fourier transform (DFT) was computed and scaled by a Mel-scale triangular filter bank. The output energy was then log-compressed and transformed via the DCT to cepstral coefficients.

Thirteen MFCCs were computed per frame, with only the last 12 used as features. In order to highlight the performance difference between context/grammar-based solution delivered by the KAARMA algorithm and results derived from a conventional Markov model, neither the log Parseval energy of each frame nor the time derivatives, i.e., delta and delta-delta coefficients (Furui, 1986), were used as a feature. HMM will benefit from these dynamic spectral features (Skowronski and Harris, 2007). However, our primary goal is to evaluate the performance using a bio-inspired end-to-end spike-based keyword spotting system, without hand-designed feature extraction algorithm past spike generation. The MFCC-HMM design parameters were selected to establish a more comparable baseline without significant increase to complexity.

The performances are summarized in **Table 1**. The KAARMA solution outperformed the HMM in both the training and testing sets. A big advantage of the KAARMA framework is that it can operate on a single frame at a time, but exploits the full context of an entire input sequence. As a recurrent network, it has an inherent deep structure in time. Furthermore, partitioning each sequence into smaller grammar states improves KAARMA performance and computational efficiency. On the other hand, in general, the amount of data needed to learn an HMM increases quadratically with the number of states.

For a comparable processing with the 12 MFCC coefficients used above, to generate the spike trains, a 12-filter gammatone filterbank with center frequencies equally spaced between 50 Hz and 8 kHz on the equivalent rectangular bandwidth (ERB)-rate scale was applied to each acoustic signal. Then, the maximum absolute amplitudes of the 12-channel output were normalized to 4 μA and converted into spike trains using LIF neurons defined by Equation (17). A single neuron is used per channel, for a total of 12 input neurons in this experimental setup. The parameters were membrane resistance $R_m = 10\,\mathrm{M}\Omega$, time constant $\tau_m = 10\,\mathrm{ms}$, spike threshold $V_{th} = -55\,\mathrm{mV}$, spike delta $V_{spike} = 500\,\mathrm{mV}$, reversal potential for SRA $E_K = -200\,\mathrm{mV}$, reset potential $V_{reset} = -80\,\mathrm{mV}$, SRA time constant $\tau_{sra} = 200\,\mathrm{ms}$, increase in SRA per spike $\Delta_{sra} = 5\,\mathrm{nS}$, time for refractory conductance to decay $\tau_{ref} = 2\,\mathrm{ms}$, and increase in refractory conductance per spike $\Delta_{ref} = 200\,\mathrm{nS}$. Again, the motivation here is that for a human-engineered speech feature such as MFCC, we can expect reliable performance with only 12 coefficients or inputs. Difference here is that instead of working with waveforms, we encode the information in a sequence of events over time, and not in the amplitude of the signal as is common in ASR. Increasing the number of input channels should improve the recognition accuracy, but as a proof-of-concept, we wanted to evaluate the baseline performance using only 12 channels of spike input.

We directly applied the spike trains in each time frame (temporal coding) as features in our isolated word recognition task. To reduce the bias from data imbalance using the one-vs.-all approach, the positive class (10% of the data for each word model) was replicated three times in the training set with random placement. A five-network KAARMA chain was used to model each word and trained for a single epoch only. To reduce over-fitting, the parameters were not fully optimized over their respective ranges. The results are presented in **Table 1**.

TABLE 1 | Comparisons of KAARMA chain classification accuracies with those of HMMs using an equivalent number of states and a mixture of eight Gaussians per state.

Input type			Training	Testing
5-State HMM				
MFCC			98.74%	98.00 %
Spike train	Rate		93.74%	93.23 %
5-Network KAARMA Chain				
MFCC	Sequence ordering:	Left-to-Right	99.33%	98.62
		Bi-Directional	***99.78***%	***99.08***%
Spike train	Rate	Left-to-Right	99.04%	91.85 %
		Bi-Directional	**99.56**%	94.54 %
	Temporal	Left-to-Right	96.70%	93.54 %
	(Spike kernel)	Bi-Directional	98.56%	**95.23** %

Only 12 MFCC coefficients were used, without log energy and time derivatives. Similarly, only 12 channels of spike trains were used. Bold values indicate the best performance.

Since HMM does not provide native support for spike trains, the spike count in each frame was used to compute the firing rate and formed a continuous-valued 12-D feature vector across all channels. We also show the five-network KAARMA chain recognition performances using spike-count or rate coding (hidden states $\mathbf{s} \in \mathbb{R}^3$, kernel parameters $a_s = a_u = 5$, learning rate $\eta = 0.1$, quantization threshold $\varepsilon = 0.55$) and temporal coding (hidden states $\mathbf{s} \in \mathbb{R}^3$, spike-train kernel parameter $a_\lambda = 1$, hidden-state kernel parameter $a_s = 4$, learning rate $\eta = 0.1$, quantization threshold $\varepsilon = 0.25$) in **Table 1**.

For rate vectors, a five grammar state KAARMA classifier outperformed similar HMM architecture (five-state with a mixture of eight Gaussians) significantly in the training set, but suffered from overfitting to a greater degree in the testing set. Using temporal coding yields worse performance on the training set, but is better on the test set. This suggests that KAARMA generalizes better using temporal coding of spike trains than rate coding. The information capacity of temporal coding is significantly greater than that of the spike-count rate and is limited only by the temporal resolution of the code. Therefore, the mismatch between model complexity and the task is reduced (spike timing provides additional temporal information over spike count), and the network is less prone to overfitting. On the other hand, spike-count rate is less sensitive to session variability and akin to the spectral power. This is evident from the performances shown in **Table 1**: left-to-right KAARMA networks can be easily trained to recognize the training set using rate coding (99.04%) vs. temporal coding (96.70%), but the better performance on the test set is given by temporal coding (93.54 vs. 91.85%).

Compared to the left-to-right KAARMA chain test-set performance using MFCCs (98.62%) and that of the HMM (98.00%), in **Table 1**, we see a drop in accuracies using spike-based front-ends. This is a testament to the popularity of MFCC as the *de facto* speech feature, but also to the fact that the focus of this paper is not to optimize the feature representation, i.e., feature engineering, but rather to demonstrate, as a proof

TABLE 2 | Comparisons of spike-input KAARMA chain with state-of-the-art SNN and sparse representation on TI46 multispeaker spoken digits.

	Speakers	Samples	Spike train	Train	Accuracy (%)
		(Train/Test)	input channels	epochs	
Spike-train KAARMA	16	4,000 $\left(\frac{2}{3}/\frac{1}{3}\right)$	12	1	95.23†
Digital LSM (Zhang et al., 2015)	16	1,590 $\left(\frac{4}{5}/\frac{1}{5}\right)$	77	500	92.30
SWAT SNN (Wade et al., 2010)	8	400 $\left(\frac{4}{5}/\frac{1}{5}\right)$	180 input neurons	250	*95.25*
LSM (Verstraeten et al., 2005)	5	500 $\left(\frac{3}{5}/\frac{2}{5}\right)$	39	–	*95.5*

†Spike-KAARMA achieved over 95% recognition accuracy using the largest subset with the fewest number of input channels and training epochs.

of concept, that a simple spike-based coding scheme achieves competitive result over other ASR systems using spikes.

Furthermore, reversing the input sequence ordering yields a complementary grammar that can be learned using a new set of KAARMA chains, and the two classification outputs can be combined (as discussed in section 3) to enhance recognition accuracy. The results from this formulation are labeled bi-directional in contrast to the natural left-to-right convention. The bi-directional KAARMA spike-based performances are also summarized in **Table 1**. The best spike test-set performance was given by bi-directional KAARMA chains operating directly on the spike trains (temporal coding) with a recognition accuracy of 95.23% with only one epoch of training.

As noted in a recent publication on LSM-based ASR (Zhang et al., 2015), a systematic comparison with other spike-based methods is difficult. There has been limited research in spike train representations for keyword spotting and speech recognition performances depend largely on specific experimental setups, which often vary greatly and are not fully reported. Most of the recent spike-based ASR systems in the literature utilize a variant of the liquid state machine (Maass et al., 2002). It is interesting to mention that speech was used in this landmark paper as an example of application of LSM, but unfortunately no validation of the method was reported. For a very small subset of the TI-46 corpus of ten different utterances of digits "zero" to "nine" (60% for training and 40% for testing), spoken by five different female speakers, the best LSM achieved a recognition accuracy of 95.5% (Verstraeten et al., 2005). Expanding on the five-speaker result, the state-of-the-art bioinspired performance on a larger subset of the TI-46 digit corpus is reported using a digital LSM (Zhang et al., 2015). For this multispeaker spoken digit task with 1590 speech samples (using five-fold cross validation: 80% used for training and remaining 20% for testing) and training epoch of 500, the final classification rate for the 77-channel spike-input digital LSM is 92.3%. For a smaller subset using a synaptic weight association training (SWAT) SNN, an accuracy of 95.25% was reported (Wade et al., 2010). Our proposed spike-based word spotting system achieved an accuracy of 95.23% for the largest subset with 4,000 samples (67% for training and 33% for testing) and all 16 speakers (eight male, eighht female), using a single training epoch (where only the desired class or 10% of the training data is replicated three times). The results are summarized in **Table 2**. Again, since the experimental setups are different, the performance comparisons are indicative and not directly quantitative. Nonetheless, spike-input KAARMA achieved over 95% recognition accuracy using the largest subset of the TI46 corpus with the fewest number of spike-train input channels (12) and training epochs (1).

Furthermore, we note that producing a constant output for time-varying liquid state is a major challenge for LSM, since the memory-less readout has to transform the transient and non-stationary states of the liquid filter into the output without any stable states or attractors to rely on Maass et al. (2002). For the KAARMA formulation using spike-based signals, once the stable dynamics are learned, we can even extract a finite state machine or deterministic finite automata (DFA) from the binary time sequences, where all the information of the input is contained in its temporal evolution, i.e., the inter-spike intervals of individual spike trains, as illustrated in our previous work (Li and Príncipe, 2016).

To further improve the classification accuracies in the current work under clean conditions, we can expand the original feature space by increasing the number of filtered outputs with a larger Gammatone filterbank and corresponding number of LIF neurons. For optimal application-specific results, feature engineering is required to design a set of novel spike-domain attributes.

4.1. Computational Complexity Analysis

For sequence learning (training) of length n using KAARMA, where the weight update frequency is only once per sequence, the memory and computation complexities are $\mathcal{O}(n)$ and $\mathcal{O}(n^2)$, respectively, the same as the simplest online kernel adaptive filter, i.e., the KLMS (Li, 2015). For testing, the memory and computation complexities are $\mathcal{O}(n)$, which can be easily implemented using parallel processing in hardware. To further reduce the computational complexity, we use the quantization technique to curb the linear growth of the network by discarding redundant data points and merging the updating coefficients with their nearest neighbors', resulting in a significantly more compact network with size $m \ll n$. The model complexity of KAARMA and other kernel or SVM methods are automatically set by the support vectors, in contrast to neural network based solutions like the SNN. The average number of support or centers of a KAARMA network is 1880.5, compared to the 5,040 neurons in the hidden layer of the SWAT SNN (Wade et al., 2010) and the 135 reservoir neurons in a multilayer 3D grid with thousands of synaptic connections randomly allocated (83 input neurons and 26 readout neurons) of the digital LSM (Zhang et al., 2015). Similarly, we only need to tune a few parameters, compared to the neuron modeling and learning,

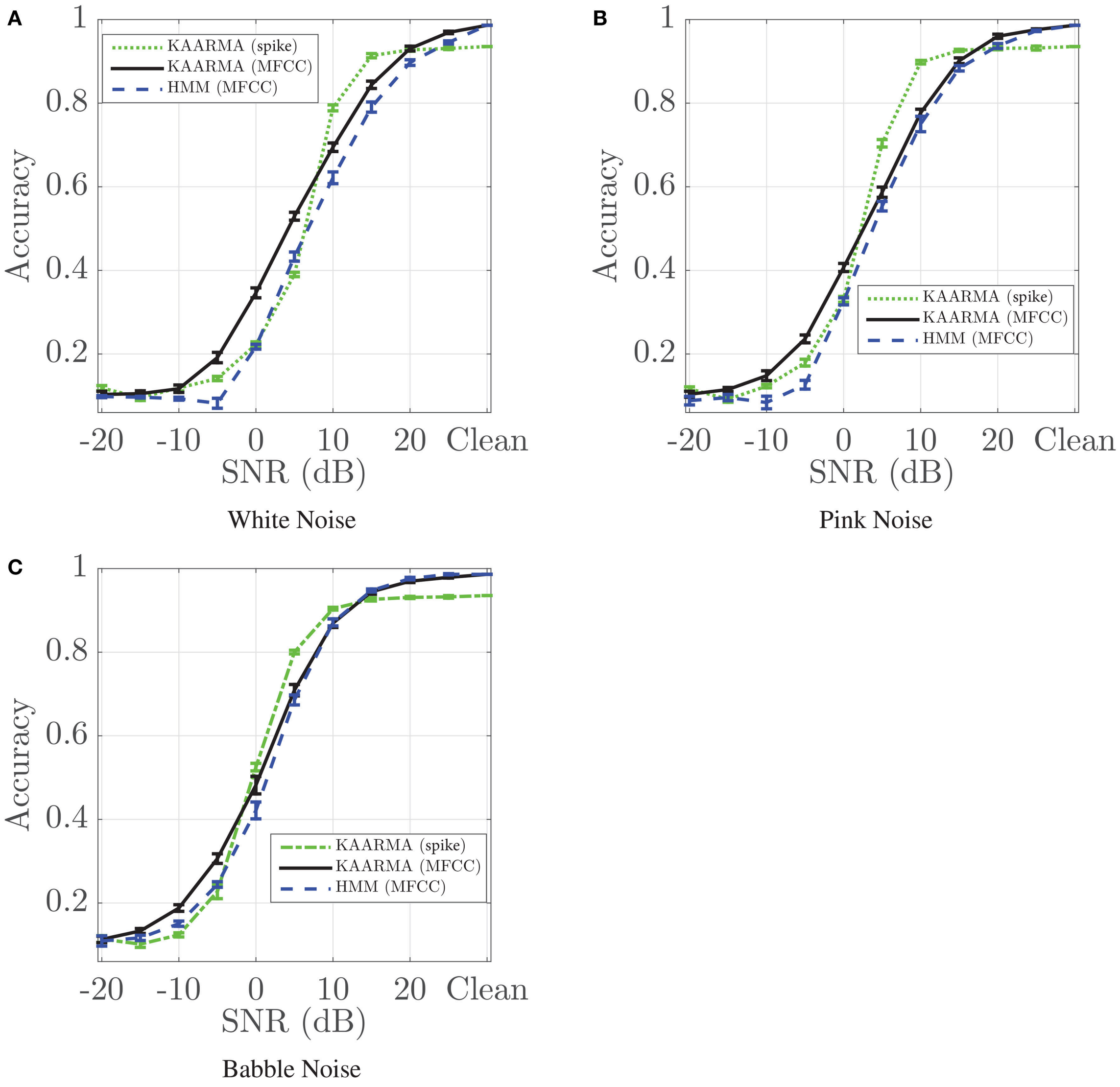


FIGURE 10 | Recognition accuracies for five-network KAARMA chain classifier using spike-train front-end compared with five-network KAARMA chain classifier and HMM using MFCC as a function of SNR. Results (mean ± 1 standard deviation) are averaged over 10 trials with different additive noise. Three types of noise sources are evaluated: **(A)** White noise degrades the performance of Mel-cepstra-based recognition systems most significantly; **(B)** Pink noise is a stationary noise having equal energy per octave; **(C)** Babble noise shares statistical properties of the reference speech and corrupts the entire information bearing spectra. For each noise type, spike-KAARMA classifiers outperformed MFCC-KAARMA and HMM baseline in certain low-SNR regime.

e.g., spike timing dependent plasticity and Bienenstock-Cooper-Munro learning in Wade et al. (2010). Furthermore, the data requirement to train KAARMA is greatly reduced compared to alternative methods. As shown in **Table 2**, KAARMA uses orders of magnitude fewer training epochs to converge to a suitable solution.

4.2. Noise Robustness Analysis

We have shown that for clean data, the KAARMA chain solution outperformed the state-of-the-art spike-based ASR system. However, we also see that KAARMA chain operating on spike trains performed worse (for bi-directional sequencing: 95.23 vs. 99.08%) than its MFCC front-end counterpart, for reasons discussed in the above section. A major drawback of MFCC features is their sensitivity to additive noise. Low energy perturbations in the power spectrum are known to cause significant variations after the log compression in their computation (Paliwal, 1998). Spike trains encoded from analogy/digital speech signals using LIF neurons have inherent noise robustness due to the integration or smoothing operation in spike generation.

Here we demonstrate that despite this initial performance degradation, KAARMA chain using spike-train front-end shows superior noise robustness in certain low-SNR regime than the MFCC front-end, with three types of noise. Additive white, pink, and multi-speaker babble noise (Hirsch and Pearce, 2000) were introduced to the test utterances, then decoded using the same KAARMA chains trained on noise-free or clean data. **Figure 10** shows the classification accuracies of the five-network left-to-right KAARMA chains train using spike-train front-end (green dotted line) as a function of SNR, from −20 to 25 dB in increments of 5 dBs. Again, although the clean-data performance on spike trains is below those of the MFCC-based solutions, the noise robustness is increased with an extended flat region from peak performance, and the drop-off SNR is pushed to the left. In certain low SNR regime, spike-based KAARMA classifiers outperformed five-network KAARMA chains and five-state HMMs using MFCCs. For additive pink noise, we see that KAARMA chain using spike-train front-end outperforms HMM with MFCC for all SNRs below 20 dB. This increased noise robustness demonstrates that neural computation is not merely an artifact of biology, but rather a key to the performance robustness of the auditory system. KAARMA classifiers are able to leverage high-dimensional nonlinear representation of speech in the RKHS, which increases the likelihood of linear class separability in the infinite-dimensional space, and the contextual information provided by the recurrency of the dynamical model.

5. CONCLUSION

We present a biologically-inspired spike-based isolated-word speech recognition or keyword spotting system with superior noise robustness using the KAARMA algorithm. By leveraging the contextual information of the input spike sequence using stable states, KAARMA networks outperform state-of-the-art spike-based processing on the benchmark TI-46 digit corpus. The grammar-based deterministic KAARMA classifier models complex nonlinear dynamical systems using spike train representation and provides a viable alternative to LSMs in small-vocabulary ASR systems and similar applications. By operating in a continuous state space, it has a parsimonious architecture, using hidden states of only three dimensions. Furthermore, spike-based KAARMA classifier outperforms its MFCC counterpart and HMMs in certain low SNR regions.

So far, in this paper, we have only provided a simple spike generation mechanism without any feature engineering step. Speech signals are encoded into spike trains and applied directly to the kernelized recurrent network. In the future, we will investigate ways to optimize the spike-based feature extraction for improved ASR performance, particularly for noisy-data. Specifically, we will address issues such as the number of filters in the gammatone filter-bank and spike-based coding that provides a suitable representation of the local spectral properties in the speech signal.

In earlier works, we represented spike trains as binned binary sequences and trained KAARMA networks to learn the dynamics directly from data, and later extracted the dynamics in the forms of deterministic finite automata (DFA). Computing using DFA is much faster than traditional methods involving analog integration or kernel functions, since state transitions are done automatically based on spike arrival, i.e., a lookup table. We will encode speech spike-train dynamics into DFA in the future. Furthermore, this methodology can be applied to other analog time series, not just limited to speech, using an appropriate analog-to-spike converter. This opens the door to countless novel applications that benefit from improved noise-robustness, ultra-low power, and ultra-fast computation, especially in hardware.

AUTHOR CONTRIBUTIONS

All authors listed, have made substantial, direct and intellectual contribution to the work, and approved it for publication.

FUNDING

This work was supported by DARPA Contracts N66001-10-C-2008 and N66001-15-1-4054, and NSF grant 1723366.

ACKNOWLEDGMENTS

We would like to thank Dr. John G. Harris for his helpful discussions during the research. We are also thankful to the editor and reviewers for their valuable comments and suggestions that improved the manuscript.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fnins.2018.00194/full#supplementary-material

REFERENCES

Bakis, R. (1976). "Continuous speech word recognition via centi-second acoustic states," in *Proc. ASA Meeting* (San Washington, DC).

Baum, L. E., Petrie, T., Soules, G., and Weiss, N. (1970). A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *Ann. Math. Stat.* 41, 164–171.

Bengio, Y., Ducharme, R., Vincent, P., and Jauvin, C. (2003). A neural probabilistic language model. *J. Mach. Learn. Res.* 3, 137–1155. Aailable online at: http://www.jmlr.org/papers/v3/bengio03a.html

Bertschinger, N., and Natschläger, T. (2004). Real-time computation at the edge of chaos in recurrent neural networks. *Neural Comput.* 16, 1413–1436. doi: 10.1162/089976604323057443

Boulard, H., and Morgan, N. (1993). Continuous speech recognition by connectionist statistical methods. *IEEE Trans. Neural Netw.* 4, 893–909.

Bourlard, H., and Wellekens, C. (1990). "Links between markov models and multilayer perceptrons," in *Proceedings IEEE Transactions on Pattern Analysis and Machine Intelligence Vol. 12*, (IEEE) 1167–1178.

Davis, S., and Mermelstein, P. (1980). "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," in

Proceedings IEEE Transactions on Acoustics, Speech, and Signal Processing Vol. 28, 357–366.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *J. R. Stat. Soc.* 39, 1–38.

Dura-Bernal, S., Li, K., Neymotin, S. A., Francis, J. T., Principe, J. C., and Lytton, W. W. (2016). Restoring behavior via inverse neurocontroller in a lesioned cortical spiking model driving a virtual arm. *Front. Neurosci.* 10:28. doi: 10.3389/fnins.2016.00028

Engel, Y., Mannor, S., and Meir, R. (2004). The kernel recursive least-squares algorithm. *IEEE Trans. Signal Process.* 52, 2275–2285. doi: 10.1109/TSP.2004.830985

Franzini, M. A., Lee, K. F., and Waibel, A. (1990). "Connectionist viterbi training: a new hybrid method for continuous speech recognition," in *Proceedings of International Conference on Acoustics Speech and Signal Processing* (Albuquerque, NM), 425–428.

Frieß, T.-T., and Harrison, R. F. (1999). "A kernel based adaline," in *ESANN* (Bruges), 245–250.

Furui, S. (1986). Speaker-independent isolated word recognition using dynamic features of speech spectrum. *IEEE Trans. Acoust. Speech Signal Process.* 34, 52–59.

Gerstner, W., and Kistler, W. M. (2002). *Spiking Neuron Models: Single Neurons, Populations, Plasticity*. Cambridge, UK: Cambridge University Press.

Hawkes, A. G. (1971). Spectra of some self-exciting and mutually exciting point processes. *Biometrika* 58, 83–90.

Haykin, S. (1998). *Neural Networks: A Comprehensive Foundation, 2nd Edn*. Upper Saddle River, NJ: Prentice Hall PTR.

Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A., Jaitly, N., et al. (2012). Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. *IEEE Signal Process. Mag.* 29, 82–97. doi: 10.1109/MSP.2012.2205597

Hirsch, H. G., and Pearce, D. (2000). "The aurora experimental framework for the performance evaluation of speech recognition systems under noise conditions," in *Proc. Int. Speech Commun. Assoc. Tutorial Res. Workshop ASR2000* (Paris), 181–188.

Hopfield, J. J., and Brody, C. D. (2001). What is a moment? Transient synchrony as a collective mechanism for spatiotemporal integration. *Proc. Natl. Acad. Sci. U.S.A.* 98, 1282–1287. doi: 10.1073/pnas.98.3.1282

Jaeger, H. (2001). *The "Echo State" Approach to Analysing and Training Recurrent Neural Networks*. Gmd report 148, German Nat. Res. Cntr. Inf. Technol., Sankt Augustin.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Trans. ASME Ser D. J. Basic Eng.* 82, 35–45.

Kivinen, J., Smola, A. J., and Williamson, R. C. (2004). Online learning with kernels. *IEEE Trans. Signal Process.* 52, 2165–2176. doi: 10.1109/TSP.2004.830991

Kuhn, R., and Mori, R. D. (1990). A cache-based natural language model for speech recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* 12, 570–583. doi: 10.1109/34.56193

Lari, K., and Young, S. J. (1990). The estimation of stochastic contextfree grammars using the inside-outside algorithm. *Comput. Speech Lang.* 4, 35–56.

Levin, E. (1990). "Word recognition using hidden control neural architecture," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing* (Albuquerque, NM), 433–436.

Li, K. (2015). *Adaptive Recurrent Filtering in Reproducing Kernel Hilbert Spaces*. Ph.D. dissertation, University of Florida.

Li, K., Ma, Y., and Príncipe, J. C. (2017). "Automatic plant identification using stem automata," in *2017 IEEE International Workshop on Machine Learning for Signal Processing (MLSP)* (Roppongi).

Li, K., and Príncipe, J. C. (2016). The kernel adaptive autoregressive-moving-average algorithm. *IEEE Trans. Neural Netw. Learn. Syst.* 27, 334–346. doi: 10.1109/TNNLS.2015.2418323

Li, K., and Príncipe, J. C. (2017a). "Automatic insect recognition using optical flight dynamics modeled by kernel adaptive arma network," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (New Orleans, LA), 2726–2730.

Li, K., and Príncipe, J. C. (2017b). "Flight dynamics modeling and recognition using finite state machine for automatic insect recognition," in *2017 International Joint Conference on Neural Networks (IJCNN)* (Anchorage, AK), 3733–3740.

Liu, W., Park, I., Wang, Y., and Príncipe, J. C. (2009). Extended kernel recursive least squares algorithm. *IEEE Trans. Signal Process.* 57, 3801–3814. doi: 10.1109/TSP.2009.2022007

Liu, W., Pokharel, P., and Príncipe, J. C. (2008). The kernel least mean square algorithm. *IEEE Trans. Signal Process.* 56, 543–554. doi: 10.1109/TSP.2007.907881

Maass, W., Legenstein, R. A., and Bertschinger, N. (2005). "Methods for estimating the computational power and generalization capability of neural microcircuits," in *Advances in Neural Information Processing Systems 17*, eds, L. K. Saul, Y. Weiss, and L. Bottou (Vancouver, BC: MIT Press), 865–872.

Maass, W., Natschlager, T., and Markram, H. (2002). Real-time computing without stable states: a new framework for neural computation based on perturbations. *Neural Comput.* 14, 2531–2560. doi: 10.1162/089976602760407955

Meddis, R. (1986). Simulation of mechanical to neural transduction in the auditory receptor. *J. Acoust. Soc. Amer.* 79, 702–711.

Mikolov, T., Karafiat, M., Burget, L., Cernokcy, J., and Khudanpur, S. (2010). "Recurrent neural network based language model," in *Proceedings of INTERSPEECH* (Makuhari), 1045–1048.

Morgan, N., and Bourland, H. (1990). "Continuous speech recognition using multilayer perceptrons with hidden markov models," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing* (Albuquerque, NM), 413–416.

Niles, L. T., and Silverman, H. F. (1990). "Combining hidden markov models and neural network classifiers," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing* (Albuquerque, NM), 417–420.

Paiva, A. R. C., Park, I., and Príncipe, J. C. (2009). A reproducing kernel Hilbert space framework for spike train signal processing. *Neural Comput.* 21, 424–449. doi: 10.1162/neco.2008.09-07-614

Paliwal, K. K. (1998). "Spectral subband centriod features for speech recognition," in *Proc. IEEE ICASSP* (Seattle, WA), 617–620.

Park, I. M., Seth, S., Paiva, A. R. C., Li, L., and Principe, J. C. (2013). Kernel methods on spike train space for neuroscience: a tutorial. *IEEE Signal Process. Mag.* 30, 149–160. doi: 10.1109/MSP.2013.2251072

Park, I. M., Seth, S., Rao, M., and Príncipe, J. C. (2012). Strictly positive-definite spike train kernels for point-process divergences. *Neural Comput.* 24, 2223–2250. doi: 10.1162/NECO_a_00309

Patterson, R. D., Nimmo-Smith, I., Holdsworth, J., and Rice, P. (1987). Annex b of the SVOS final report: an efficient auditory filterbank based on the gammatone function. *Appl. Psychol.* 1–33.

Príncipe, J. C., and Chen, B. (2015). Universal approximation with convex optimization: Gimmick or reality. *IEEE Comp. Intell. Mag.* 10, 68–77. doi: 10.1109/MCI.2015.2405352

Rabiner, L. R. (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proc. IEEE* 77, 257–286.

Ralaivola, L., and d'Alche Buc, F. (2005). "Time series filtering, smoothing and learning using the kernel Kalman filter," in *IEEE International Joint Conference on Neural Networks, 2005, Vol. 3*, (Montreal, QC), 1449–1454.

Robinson, T. (1994). An application of recurrent nets to phone probability estimation. 5, 298–305.

Scholkopf, B., and Smola, A. J. (2001). *Learning with Kernels, Support Vector Machines, Regularization, Optimization and Beyond*. Cambridge, MA: MIT Press.

Shpigelman, L., Singer, Y., Paz, R., and Vaadia, E. (2005). Spikernels: predicting arm movements by embedding population spike rate patterns in inner-product spaces. *Neural Comput.* 17, 671–690. doi: 10.1162/0899766053019944

Skowronski, M. D., and Harris, J. G. (2004). Exploiting independent filter bandwidth of human factor cepstral coefficients in automatic speech recognition. *J. Acoust. Soc. Am.* 116, 1774–1780. doi: 10.1121/1.1777872

Skowronski, M. D., and Harris, J. G. (2007). Noise-robust automatic speech recognition using a predictive echo state network. *IEEE Trans*

Audio Speech Lang. Process. 15, 1724–1730. doi: 10.1109/TASL.2007.896669

Verstraeten, D., Schrauwen, B., and Campenhout, J. V. (2005). "Recognition of isolated digits using a liquid state machine," in *Proc. SPS-DARTS 2005* (Antwerp), 135–138.

Viterbi, A. J. (1967). Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. *IEEE Trans. Inform. Theory* 13, 260–269.

Wade, J. J., McDaid, L. J., Santos, J. A., and Sayers, H. M. (2010). SWAT: a spiking neural network training algorithm for classification problems. *IEEE Trans. Neural Netw.* 21, 1817–1830. doi: 10.1109/TNN.2010.2074212

Zhang, Y., Li, P., Jin, Y., and Choe, Y. (2015). A digital liquid state machine with biologically inspired learning and its application to speech recognition. *IEEE Trans. Neural Netw. Learn. Syst.* 26, 2635–2649. doi: 10.1109/TNNLS.2015.2388544

Zhu, P., Chen, B., and Príncipe, J. C. (2014). Learning nonlinear generative models of time series with a Kalman filter in RKHS. *IEEE Trans. Signal Process.* 62, 141–155. doi: 10.1109/TSP.2013.2283842

Conflict of Interest Statement: We declare a pending patent filed with the University of Florida (UF): Pulse-Based Automatic Speech Recognition, UF #15736; PCT/US2016/065344; WO2017100298A1.

ORIGINAL RESEARCH
published: 11 September 2017
doi: 10.3389/fnins.2017.00485

Cascaded Amplitude Modulations in Sound Texture Perception

Richard McWalter and Torsten Dau**

Hearing Systems Group, Technical University of Denmark, Kongens Lyngby, Denmark

Sound textures, such as crackling fire or chirping crickets, represent a broad class of sounds defined by their homogeneous temporal structure. It has been suggested that the perception of texture is mediated by time-averaged summary statistics measured from early auditory representations. In this study, we investigated the perception of sound textures that contain rhythmic structure, specifically second-order amplitude modulations that arise from the interaction of different modulation rates, previously described as "beating" in the envelope-frequency domain. We developed an auditory texture model that utilizes a cascade of modulation filterbanks that capture the structure of simple rhythmic patterns. The model was examined in a series of psychophysical listening experiments using synthetic sound textures—stimuli generated using time-averaged statistics measured from real-world textures. In a texture identification task, our results indicated that second-order amplitude modulation sensitivity enhanced recognition. Next, we examined the contribution of the second-order modulation analysis in a preference task, where the proposed auditory texture model was preferred over a range of model deviants that lacked second-order modulation rate sensitivity. Lastly, the discriminability of textures that included second-order amplitude modulations appeared to be perceived using a time-averaging process. Overall, our results demonstrate that the inclusion of second-order modulation analysis generates improvements in the perceived quality of synthetic textures compared to the first-order modulation analysis considered in previous approaches.

Keywords: sound texture, amplitude modulation, auditory model, natural sound, auditory perception

OPEN ACCESS

Edited by:
Malcolm Slaney,
Google (United States), United States

Reviewed by:
Yves Boubenec,
École Normale Supérieure, France
Nai Ding,
Zhejiang University, China

****Correspondence:***
Richard McWalter
mcwalter@mit.edu
Torsten Dau
tdau@elektro.dtu.dk

Specialty section:
This article was submitted to Auditory Cognitive Neuroscience, a section of the journal Frontiers in Neuroscience

***Received:** 29 March 2017*
***Accepted:** 15 August 2017*
***Published:** 11 September 2017*

Citation:
McWalter R and Dau T (2017) Cascaded Amplitude Modulations in Sound Texture Perception. Front. Neurosci. 11:485. doi: 10.3389/fnins.2017.00485

INTRODUCTION

Sound textures are characterized by their temporal homogeneity and may be represented with a relatively compact set of time-averaged summary statistics measured from early auditory representations (Saint-Arnaud and Popat, 1995; McDermott et al., 2013). Although, textures can be expressed in a relatively compact form, they are ubiquitous in the natural world and span a broad perceptual range (e.g., rain, fire, ocean waves, insect swarms etc.). The perceptual range has been defined by a set of *texture* statistics outlined by McDermott and Simoncelli (2011). However, it remains unclear what sound features might also be represented in the auditory system via a time-averaging mechanism. In the present study, we investigated and expanded the perceptual space of texture, particularly in the domain of amplitude modulations.

The texture synthesis system of McDermott and Simoncelli (2011) described spectral and temporal tuning properties of the early auditory system that are crucial for texture perception. Synthetic textures were generated by measuring time-averaged texture statistics at the output of several processing stages of a biologically plausible auditory model, which were subsequently used to shape a Gaussian noise seed to have matching statistics. The auditory texture model included frequency-selective auditory filters and amplitude-modulation selective filters derived from both psychophysical and physiological data (Dau et al., 1997). The authors demonstrated that when the auditory model deviated in its biological plausibility, such as applying linearly spaced auditory filters, the perceptual quality of the texture exemplars was reduced. In addition, McDermott and Simoncelli (2011) identified which texture statistics were necessary for correct identification, revealing subsets of statistics that were requisite for different sound textures. Collectively, the results suggested that textures synthesized with the complete set of texture statistics and a biologically plausible auditory model were preferred over all other identified synthesis system configurations.

The sound synthesis system proposed by McDermott and Simoncelli (2011) generated compelling exemplars for a broad range of sounds, but there were also sounds for which the auditory texture model failed to capture some of the perceptually significant features. The failures were identified by means of a realism rating performed by human listeners, who compared synthetic textures to corresponding original real-world texture recordings. The shortcomings were attributed to either the processing structure or the statistics measured from the auditory texture model. One such texture group were sounds that contained rhythmic structure (McDermott and Simoncelli, 2011).

In the present study, the auditory texture model of McDermott and Simoncelli (2011) was extended to include sensitivity to second-order amplitude modulations. Second-order amplitude modulations arise from beating in the envelope-frequency domain. Intuitively, this can be described as the interaction between two modulators acting on a carrier. At slow modulation rates, second-order amplitude modulations have the perceptual quality of simple rhythms (Lorenzi et al., 2001a). This type of amplitude modulation has been shown to be salient in numerous behavioral experiments (Lorenzi et al., 2001a,b; Ewert et al., 2002; Verhey et al., 2003; Füllgrabe et al., 2005). The perception of second-order amplitude modulation has also been modeled by applying non-linear processing and modulation-selective filtering to a signal's envelope (Ewert et al., 2002). While the role of second-order amplitude modulation in sound perception has been investigated using artificial stimuli, their significance in natural sound perception has yet to be examined.

We undertook an analysis-via-synthesis approach to examine the role of second-order amplitude modulations in sound texture perception (Portilla and Simoncelli, 2000; McDermott and Simoncelli, 2011). This entailed generating synthetic sounds from time-averaged statistics measured at different stages of our auditory texture model (**Figure 1A**). The synthetic sounds were controlled by two main factors: the structure of the auditory texture model and the statistics passed to the texture synthesis system. We first ensured that the sound texture synthesis system was able to capture the temporal structure of a second-order amplitude modulated signal (**Figure 1B**). Subsequently, we examined the significance of the auditory texture model in a series of behavioral texture identification and preference tasks. Lastly, we attempted to quantify the role of time-averaging in the perception of second-order amplitude modulation stimuli.

METHODS

Auditory Texture Model

The auditory texture model is based on a cascaded filterbank structure that separates the signal into frequency subbands (**Figure 1A**). The first stage of the model uses 34 gammatone filters, equally spaced on the equivalent rectangular bandwidth $(\mathrm{ERB})_{\mathrm{N}}$ scale from 50 Hz to ∼8 kHz (Glasberg and Moore, 1990):

$$\boldsymbol{g}\,(\boldsymbol{t}) = \boldsymbol{c}\boldsymbol{t}^{3}\boldsymbol{e}^{2\pi i\cdot f_c\cdot t}\boldsymbol{e}^{-2\pi\cdot\beta\cdot t},$$

where f_c is the gammatone center frequency, β is a bandwidth tuning parameter and c is a scale coefficient. Although gammatone filters only capture the basic frequency selectivity of the auditory system, more advanced and dynamic filterbank architectures, such as dynamic compressive gammachirp filters (Irino and Patterson, 2006), did not yield any improvement in texture synthesis as observed in pilot experiments. To allow for the reconstruction of the subbands, a paraconjugate filter, $\tilde{G}(z)$, was created for each gammatone filter, $G(z)$ (Bolcskei et al., 1998):

$$\tilde{\boldsymbol{G}}\,(\boldsymbol{z}) = \left(\frac{\boldsymbol{1}}{\boldsymbol{G}\,(\boldsymbol{z})}\right)\cdot\left(\boldsymbol{G}\,(\boldsymbol{z})\,\boldsymbol{G}\,(\boldsymbol{z})^{\boldsymbol{T}} + \boldsymbol{G}^{*}\,(\boldsymbol{z})\,\boldsymbol{G}^{*}\,(\boldsymbol{z})^{\boldsymbol{T}}\right),$$

where $G\,(z)$ is the Fourier transform of $g(t)$, and $G^{*}(z)$ is the complex conjugate of $G(z)$. Perfect reconstruction is achieved as long as:

$$\tilde{\boldsymbol{G}}\,(\boldsymbol{z})\,\boldsymbol{G}\,(\boldsymbol{z}) = \boldsymbol{1}.$$

To model fundamental properties of the peripheral auditory system, we applied compression and envelope extraction to the subband signals. The compression was used to model the non-linear behavior of the cochlea (e.g., Ruggero, 1992) and was implemented as a power-law compression with an exponent value of 0.3. As all textures were presented at a sound pressure level (SPL) of 70 dB, it was deemed not necessary to include level-dependent compression. To functionally model the transduction from the cochlear to the auditory nerve, the envelopes of the compressed subbands were extracted using the Hilbert transform and down-sampled to 400 Hz (McDermott and Simoncelli, 2011). The compressed, down-sampled envelopes roughly estimate the transduction from basilar-membrane vibrations to inner hair-cell receptor potentials.

The model then processed each cochlear channel signal by a modulation filterbank, accounting for the first-order modulation sensitivity and selectivity of the auditory system. The filterbank applied to each cochlear channel comprised of 19 filters, half-octave spaced from 0.5 to 200 Hz. This type of functional modeling is consistent with previous perceptual models of modulation sensitivity (Dau et al., 1997) and shares similarities with neurophysiological findings (Miller et al., 2002; Joris et al., 2004; Malone et al., 2015). The broadly tuned modulation filters have a constant $Q = 2$ and a shape defined by a Kaiser–Bessel window. Reconstruction of the modulation

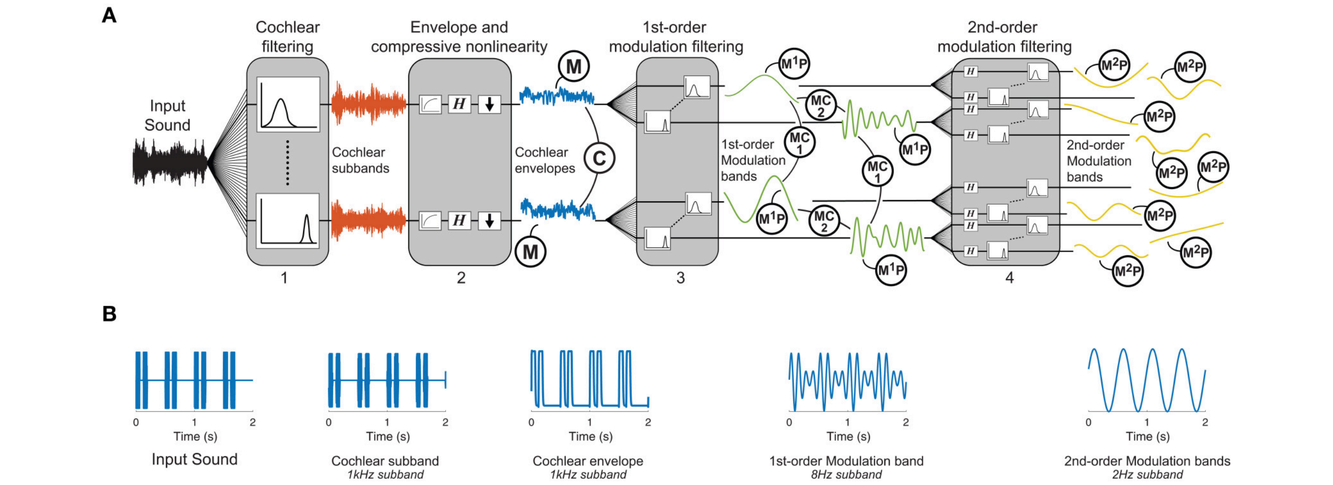

A
Cochlear filtering
Envelope and compressive nonlinearity
1st-order modulation filtering
2nd-order modulation filtering
Input Sound
Cochlear subbands
Cochlear envelopes
1st-order Modulation bands
2nd-order Modulation bands
H
M
C
M1P
MC 1
MC 2
M2P
1
2
3
4
B
0
1
2
Time (s)
Input Sound
Cochlear subband
1kHz subband
Cochlear envelope
1kHz subband
1st-order Modulation band
8Hz subband
2nd-order Modulation bands
2Hz subband

FIGURE 1 | Texture analysis model. **(A)** The functional auditory model captures the tuning properties of the peripheral and subcortical auditory system: (1) An auditory filterbank simulates the resonance frequencies of the cochlea, (2) a non-linearity captures the compression of the cochlea followed by a computation of the Hilbert envelope, functionally modeling the transduction from the mechanical vibrations on the basilar membrane to the receptor potentials in the hair cells, (3) a first-order modulation filterbank captures the selectivity of the auditory system to different envelope fluctuation rates, and (4) a second-order modulation filterbank captures the sensitivity of the auditory system to beating in the envelope frequency domain. Texture statistics include marginal moments of cochlear envelopes (M), 1st-order modulation power ($\mathrm{M}^1\mathrm{P}$), pair-wise correlations between cochlear envelopes (C), pairwise correlations between modulation subbands (MC_1), phase correlations between octave-spaced modulation bands (MC_2), and 2nd-order modulation power ($\mathrm{M}^2\mathrm{P}$). **(B)** Example second-order modulation stimulus. The far-left panel shows the input stimulus that consists of two short 62.5 ms pulses repeated every 500 ms. The example outputs are shown at each stage of the model. The output of the 1st-order modulation band is shown for the 8 Hz subband which captures the period of the short pulses. The 2nd-order modulation band is shown for the 2 Hz subband which captures the period of the repetition.

filterbank was achieved with the same method as the frequency selective gammatone filterbank.

The output of each modulation filter was subsequently processed by a second modulation filterbank, accounting for the sensitivity of the auditory system to second-order amplitude modulations. Each second-order modulation filterbank contained 17, half-octave spaced bands in the range from 0.25 to 64 Hz. The model was inspired by behavioral experiments and simulations revealing an auditory sensitivity to second-order modulations that is similar in nature to the sensitivity to first-order amplitude modulations (Lorenzi et al., 2001a,b; Ewert et al., 2002; Füllgrabe et al., 2005). The model processing layer proposed here has some shared attributes to the model presented in Ewert et al. (2002), but has the added benefit of being easily invertible. The second-order modulation filters have a constant $Q = 2$ and a Kaiser–Bessel window.

Texture Statistics

The goal of statistics selection is to find a description of sound textures that is consistent with human sensory perception (Portilla and Simoncelli, 2000). The selected statistics should be based on relatively simple operations that could plausibly occur in the neural domain. The values of the measured statistics should also vary across textures, facilitating the recognition of sound textures by the difference in the statistical representation. Lastly, there should be a perceptual salience to the textures, such that the use of their statistics contributes to the realism of the corresponding synthetic texture.

The statistics measured from the auditory model include marginal moments and pair-wise correlations (Portilla and Simoncelli, 2000; McDermott and Simoncelli, 2011). The included texture statistics are similar to those described in McDermott and Simoncelli (2011). They were computed from the envelope of the cochlea channels, including the first- and second-order modulation filters, and were measured over texture excerpts of several seconds. Examples of the statistics for three textures (insect swarm, campfire, and small stream) measured from the auditory texture model (**Figure 1A**) are shown in **Figure 2**.

The envelope statistics include the mean (μ), coefficient of variance ($\frac{\sigma^2}{\mu^2}$), skewness (η), and kurtosis (κ), and represent the first four marginal moments, defined as:

$$\begin{aligned}
\mu_n &= \overline{\vec{x}_n}, \\
\frac{\sigma_n^2}{\mu_n^2} &= \frac{\overline{\left(\vec{x}_n - \mu_n\right)^2}}{\mu_n^2}, \\
\eta_n &= \frac{\overline{\left(\vec{x}_n - \mu_n\right)^3}}{\sigma_n^3}, \\
\kappa_n &= \frac{\overline{\left(\vec{x}_n - \mu_n\right)^4}}{\sigma_n^4},
\end{aligned}$$

where n is the cochlear channel of x. Pair-wise correlations were computed as a cross-covariance with the form:

$$c_{mn} = \frac{\overline{\left(\vec{x}_m - \mu_m\right)\left(\vec{x}_n - \mu_n\right)}}{\sigma_m \sigma_n},$$

where m and n are the cochlear channel pairs. The final statistic captures envelope phase:

$$c_{jk} = \frac{\overline{\vec{d}_k^{\,*}\vec{a}_j}}{\sigma_k \sigma_j},\ d_k = \frac{a_k^2}{\|a_k\|},\ \vec{a}_k = \vec{b}_k + iH\left(\vec{b}_k\right),$$

where j and k are the modulation channel pairs of b, $\boldsymbol{H}$ is the Hilbert transform, and $*$ is the complex conjugate.

First Level Statistics

The first level of statistics were measured on the cochlear envelopes of the auditory texture model (**Figure 1**). The marginal moments (M) describe the distribution of the individual subbands (**Figure 2A**) and capture the overall level as well as the sparsity of the signal (Field, 1987). The correlation statistics (C) capture how neighboring signals co-vary. The correlation statistics are measured between the eight neighboring cochlear channels (**Figure 2B**). There are 372 statistics measured at the cochlear stage of the auditory model ($M = 128$, and $C = 236$).

Second Level Statistics

The second level statistics were measured on the first-order modulation bands (**Figure 1**) and include the coefficient of variance (M^1P, **Figure 2C**), the correlation measured across cochlear channels and first-order modulation channels (MC1, **Figure 2D**), and the correlation measured across modulation channels for the first-order modulations (MC2, **Figure 2E**). Because the outputs of the modulation filters have zero mean, the variance effectively reflects a measure of the modulation channel power. The variance was measured for cochlear channels that have a center frequency at least four times that of the modulation frequency (Dau et al., 1997). The modulation correlations measured across cochlear channels (MC1) reflect a cross-covariance measure. The correlation was measured for two neighboring cochlear channels. The modulation correlation measured across modulation rates (MC2) included phase information and was computed for octave-spaced modulation frequencies. The number of statistics considered in the modulation domain was 1,258 ($M^1P = 646$, $MC1 = 408$, and $MC2 = 204$).

Third Level Statistics

The last analysis stage was conducted on the second-order modulation envelope bands (**Figure 1**), where the modulation power was measured for each band (M^2P, **Figure 2F**). This analysis stage extends beyond the model of McDermott and Simoncelli (2011) to capture second-order modulations (Lorenzi et al., 2001b). The power was measured for first-order modulation rates that are at least twice that of the second-order modulation rate. The 2nd-order modulation power required the largest overall number of statistics ($M^2P = 3{,}400$).

Synthesis System

The synthesis of sound textures was accomplished by modifying a Gaussian noise seed to have statistics that match those measured from a real-world texture recording (Portilla and Simoncelli,

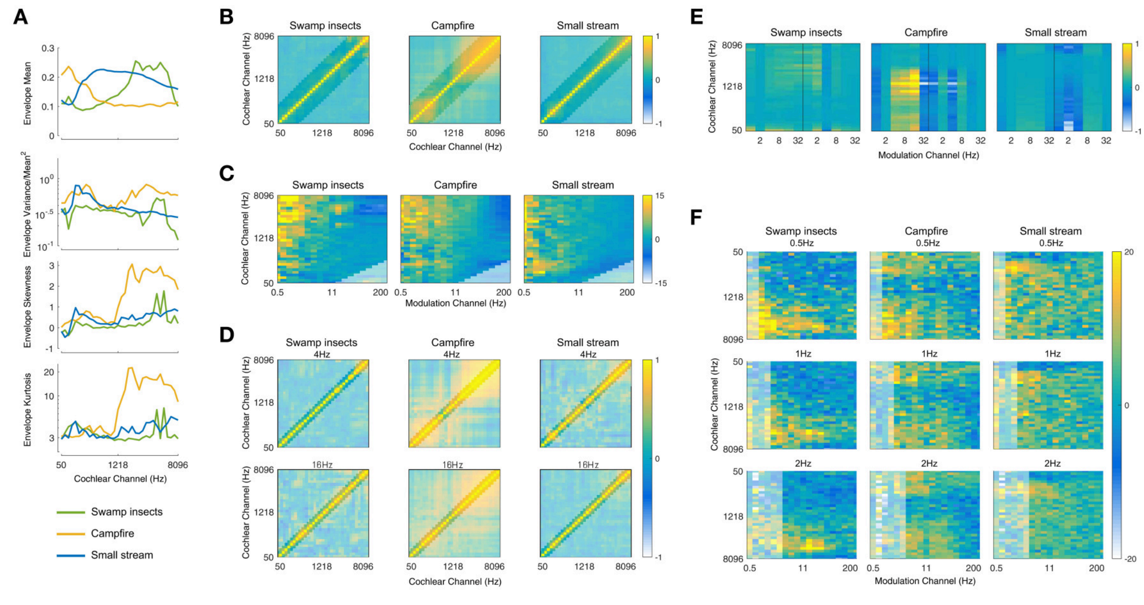


FIGURE 2 | Texture Statistics. **(A)** Cochlear envelope marginal moments (mean, coefficient of variance, skewness, kurtosis) measured from three real-world texture recordings (Swamp insects, campfire, small stream). **(B)** Cochlear envelope pair-wise correlations measured between different cochlear channels. The label of the texture analyzed is located above the subfigure (and for all subsequent subfigures). Lightened regions here and elsewhere denote texture statistics that are not imposed during the synthesis process. **(C)** Modulation band power (variance). The figure is normalized by the modulation power of Gaussian noise and shown on a log (dB) scale. **(D)** Modulation correlation measured for a particular rate across cochlear channels. The modulation rate is indicated above the subfigure. **(E)** Modulation phase correlation measured between octave-spaced modulation bands. **(F)** Second-order modulation band power (variance). The second-order modulation frequency is indicated above the individual subfigures for a selection of rates (0.5, 1, and 2 Hz). The statistics are plotted relative to Gaussian noise on a log (dB) scale.

2000; McDermott and Simoncelli, 2011). The original texture recording was decomposed using our biologically motivated auditory model where the texture statistics were measured. The statistics were then passed to the synthesis algorithm which imposed the measured statistics on the decomposed Gaussian noise signal. The modified signals were reconstructed back to a single-channel waveform. A schematic of the synthesis system can be seen in **Figure 3A**.

The imposition of texture statistics on the noise input was achieved using the LM-BFGS variant of gradient descent (limited-memory Broyden–Fletcher–Goldfarb–Shanno). The noise signal was decomposed to the second-order modulation bands, where the power statistics were imposed. The bands were then reconstructed to the first-order modulation bands, and the modulation power and correlation statistics were imposed. The modulation bands were then reconstructed to the cochlear envelopes, where the marginal moments and pair-wise correlations statistics were imposed. Lastly, the cochlear envelopes were combined with the fine-structure of the noise seed and the cochlear channels were resynthesized to the single channel waveform.

The synthesis process requires many iterations in order to attain convergence for each of the texture statistics due to the reconstruction of the subbands and tiered imposition of statistics. The reconstruction of the filterbanks modified the statistics of each subband due to the overlap in frequency of neighboring filters. The reconstruction from the cochlear envelopes to the cochlear channels was also affected by the combination of the envelope and fine structure. In addition, the texture statistics were modified at 3 layers (cochlear envelopes, 1st-order modulations, and 2nd-order modulations) of the auditory model, and the modification at each level had an impact on the other two. Due to these two factors, an iterative process for imposing texture statistics was required.

The synthesis was deemed successful if the synthetic texture statistics approached those measured from the original real-world texture recoding. The convergence was evaluated based on the signal-to-noise ratio (SNR) between the synthetic and original texture statistics (Portilla and Simoncelli, 2000). When the synthesis process reached an SNR of 30 dB or higher across the texture statistics, the process ended, generating a synthetic texture. The system also had a maximum synthesis iteration limit of 60. However, the convergence criterion was often met within 60 iterations. The cochleograms of the original and synthetic textures are shown in **Figure 3B**.

Texture Synthesis System Validation

The proposed auditory texture model and adjoining synthesis system were validated with a second-order amplitude modulated signal identified by McDermott and Simoncelli (2011). The signal was generated by applying a binary mask to a Gaussian noise carrier. The mask contained a long noise burst ($t = 0.1875\,\text{s}$ or $\frac{3}{16}$ s), followed by two short noise bursts $\left(t = 0.0625\,\text{s or } \frac{1}{16}\,\text{s}\right)$ that were repeated every 500 ms (see **Figure 4A**, upper panel). The stimulus has a second-order modulation of 2 Hz, generated by the interaction between two first-order modulations at 6 and 8 Hz.

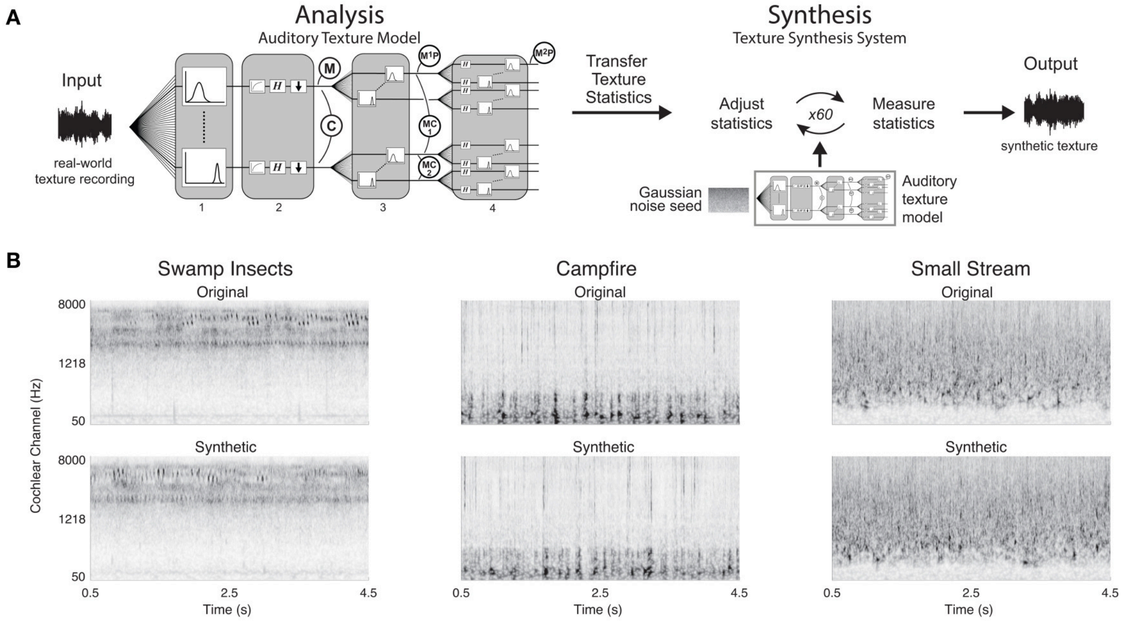


FIGURE 3 | Texture synthesis system and synthetic examples. **(A)** Texture synthesis is accomplished by measuring statistics from a real-world texture recording at different stages of the auditory texture model. The statistics are then passed to the synthesis system that adjusts the statistics of a Gaussian noise seed to match the input statistics. The iterative process outputs a synthetic texture with the same time-averaged statistics as the real-world texture recording. **(B)** Original real-world texture recordings and their synthetic counterparts. The synthetic textures were generated with a complete set of texture statistics. Example audio files corresponding to the original and synthetic spectrograms can be found in the Supplementary Material (Swamp Insects: **Audio files 1**, **2**; Campfire: **Audio files 3**, **4**; Small Stream: **Audio files 5**, **6**).

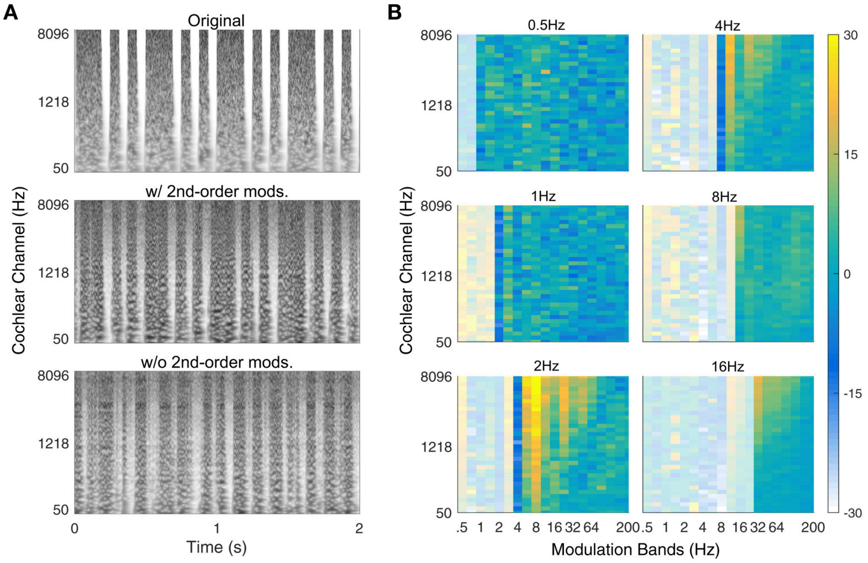


FIGURE 4 | Verification of second-order texture synthesis. **(A)** Spectrogram of example rhythmic (second-order modulated) noise bursts with 500 ms repetition pattern. The upper panel shows the original sound, the middle panel shows the synthetic version with second-order modulation texture statistics (w/ 2nd-order mods.) and the bottom panel shows the synthetic version without second-order modulation texture statistics (w/o 2nd-order mods.). **(B)** Second-order modulation power statistics. The 500 ms period is reflected in the majority of power held within the 2 Hz 2nd-order modulation band (lower-left panel). Example audio files corresponding to the spectrograms can be found in the Supplementary Material (Original: **Audio file 7**; w/ 2nd-order mods.: **Audio file 8**; w/o 2nd-order mods.: **Audio file 9**).

Psychophysical Experiments

The listeners were recruited from a university specific job posting site. The listeners completed the required consent form and were compensated with an hourly wage for their time. All experiments were approved by the Science Ethics Committee for the Capital Region of Denmark.

The listeners performed the experiment in a single-walled IAC sound isolating booth. The sounds were presented at 70 dB SPL via Sennheiser HD 650 headphones. The playback system included an RME Fireface UCX soundcard and the experiments were all created using Mathworks MATLAB and the PsychToolBox (psychtoolbox.org) software.

The synthetic textures used in experiments 1 and 2 were generated in 5-s long samples. Multiple exemplars were generated for each texture. Each exemplar was created using a different Gaussian noise seed such that no sample was identical in terms of the waveform, but had the same time-averaged texture statistics. Four-second long excerpts were taken from the middle portion of the texture samples with a tapered cosine (Tukey) window with 20-ms ramps at the onset and offset.

Experiment 1—Texture Identification

Each trial consisted of a 4-s texture synthesized from subsets of texture statistics that were cumulatively included from the cochlear envelope mean to the 2nd-order modulation power. The listeners were required to identify the sound from a list of 5 label descriptors. The experiment consisted of 59 sound textures. The textures were divided into 5 texture groups, defined by the authors: animals, environment, mechanical, human, and water sounds. The list of 4 incorrect labels for each texture was selected from different texture groups. There were 7 conditions per texture (6 synthetic and 1 original) and 413 trials per experiment. Eleven self-reported normal-hearing listeners participated in the experiment (6 female, 23.3 mean age).

Experiment 2—Modulation Processing Model Comparison

Each trial consisted of three intervals; the original real-world texture recording, a synthetic texture generated from the above-mentioned texture synthesis system (reference), and a synthetic texture generated from a modified version of the auditory model. The real-world texture was presented first. Textures generated from the reference system and a modified auditory model were then presented in intervals 2 and 3, where by the order of presentation was randomized. Each interval was 4 s long with an inter-stimulus-interval of 400 ms. The listeners were asked to select the interval that was most similar to the real-world texture recording. The same 59 textures were used in the experiment, presented in 236 trails. Eleven self-reported-normal hearing listeners participated in the experiment (7 female, 24.2 mean age).

Synthetic textures generated from a reference auditory model and four alternate auditory models were included in the experiment. The reference model is described in **Figure 1**, including texture statistics measured from the cochlear envelope, 1st- and 2nd-order modulation bands. The first alternate model removed the 2nd-order modulation bands, and was in principle similar to that of McDermott and Simoncelli (2011). The second alternate model removed the 2nd-order modulation bands and replaced the half-octave spaced 1st-order modulation filterbank by an octave-spaced variant. Octave-spaced modulation selectivity has been suggested in several models of auditory perception (Dau et al., 1997; Jorgensen and Dau, 2011). The third alternate model removed the 2nd-order modulation bands and substituted the half-octave spaced modulation filterbank with a low-pass filter of 150 Hz. The low-pass characteristic of amplitude modulation perception has been proposed, and here we used a model that preserves the sensitivity to modulation rates but lacks the selectivity of the filterbank model (Kohlrausch et al., 2000; Joris et al., 2004). The fourth alternate model also removed the 2nd-order modulation bands and substituted the half-octave spaced modulation filterbank with a low-pass filter with a cutoff frequency of 5 Hz. The sluggishness of the auditory system to amplitude modulation perception is reflected in the heightened sensitivity to slow modulation rates (Viemeister, 1979; Dau et al., 1996).

Experiment 3—Second-Order Modulation Discrimination

Each trial consisted of three 2-s intervals. The listeners performed an odd-one-out experiment, where they were instructed to identify the interval (first or last) that was different from the other two. The stimulus sets described below were evaluated in separate experiment blocks. Twelve self-reported-normal hearing listeners participated in the experiment (3 female, 23.0 mean age).

The first stimulus set was generated from second-order amplitude modulated white noise using the following equation:

$$\boldsymbol{s}\left(\boldsymbol{t}\right)=\left(\mathbf{1}+\left(\mathbf{0.5}+\sin\left(2\pi\boldsymbol{f_{m1}t}+\phi\right)\right)*\sin\left(2\pi\boldsymbol{f_{m2}t}\right)\right)*\boldsymbol{n}\left(\boldsymbol{t}\right),$$

where f_{m1} is the first modulator, t is time, ϕ is the phase of the first modulator, $\boldsymbol{f_{m2}}$ is the second modulator, and $n\left(t\right)$ is the Gaussian noise carrier. f_{m1} had a modulation frequency of 2, 4, 8, 16, 32, or 64. f_{m2} had a modulation rate of f_{m1}[0.1, 0.13, 0.17, 0.22, 0.28, 0.36, 0.46, 0.60, 0.77, *or* 1.00]. ϕ was randomized for each trial. The exemplars were 5 s in duration. Two intervals were sampled from the first 2 s, and the "odd" interval was sampled from the last 2 s. Each condition was repeated 4 times, for a total of 240 trials.

The next stimulus set used second-order amplitude modulated white noise generated from a combination of f_{m1} and f_{m2} pairs, creating a complex amplitude modulated signal. Each stimulus was created using the six f_{m1} frequencies, each paired with a corresponding f_{m2} frequency that was randomly selected from the list of 10, modulating the same white noise seed. The six second-order modulated signals were then summed to create one stimulus. The exemplars were 5 s in duration. Two intervals were sampled from the first 2 s, and the "odd" interval was sampled from the last 2 s. There were 48 stimuli presented, one per trial.

The final stimulus set was composed of sound textures generated with the complete set of texture statistics, including second-order amplitude modulation power. The 59 textures used in experiments 1 and 2 were used in this experiment. The exemplars were 5 s in duration. Two intervals were sampled from the first 2 s, and the "odd" interval was sampled from the last 2 s. There were 59 trials in total.

RESULTS

The auditory texture model proposed in the present study includes frequency-selective filtering (in the audio-frequency domain) as well as a cascade of amplitude modulation filterbanks to capture time-averaged amplitude modulations and simple rhythmic structure. The model was combined with a sound synthesis system to generate synthetic textures that were then examined in several behavioral listening experiments. The results show three main findings: (1) the model captures simple rhythmic structure by way of second-order amplitude modulation analysis, (2) the inclusion of second-order amplitude modulation analysis contributes to the recognition of the synthetic textures, and (3) second-order amplitude modulations in textures may be perceived using time-averaged summary statistics.

Synthesis Verification for Second-Order Modulations

Although, the second-order texture statistics varied across textures, it was unclear how the synthesis process would perform in creating new sound examples. To test this, we used a second-order amplitude modulation signal identified by McDermott and Simoncelli (2011) that has a salient rhythmic structure. **Figure 4A** shows the original sound (top), a synthetic version with second-order modulation analysis (middle) and a synthetic version without second-order analysis (bottom). The synthetic sound generated from texture statistics that included second-order amplitude modulation analysis captured the rhythmic pattern of the original sound, whereas the version without second-order analysis failed to capture the rhythmic structure even though the duration of the noise bursts is comparable to that in the original sound. The successful synthesis of the rhythmic sound suggests that the cascaded modulation filterbank analysis can capture rhythmic structure.

The second-order amplitude modulation statistics for the example rhythmic sound are shown in **Figure 4B**. The majority of the modulation power can be found in the 2 Hz second-order modulation channel (bottom left panel) across several first-order modulation rates. For a relatively simple rhythmic sound, there is considerable modulation power across frequencies. This is primarily due to amplitude modulation interactions between the modulation frequencies and the broadband (Gaussian) noise carrier. If a second-order amplitude modulated tone was used instead of the noise with its intrinsic modulations, the modulation power would be relegated entirely to the 2-Hz band.

Texture Perception: Identification and Preference

Our first behavioral experiment investigated the ability of listeners to identify sound textures generated from subsets of statistics. Listeners were presented with a 4 s texture and asked to identify the sound from a list of 5 text label descriptors. The textures synthesized with the cochlear envelope power resulted in low performance, but the performance increased with the inclusion of higher-order texture statistics and approached that of the original real-world texture recording when second-order amplitude modulation statistics were used [**Figure 5A**; $F_{(6,\,49)} = 123.51$, $p < 0.0001$]. The results suggest that listeners benefited from the addition of second-order amplitude modulation analysis to the auditory texture model.

Next, we were interested in how synthetic textures generated with alternate amplitude modulation processing models compared to our auditory texture models. To investigate this, we generated textures from four models that included only the first-order amplitude modulation analysis (**Figure 5B**). The results show that our auditory texture model, with second-order amplitude modulation analysis, was preferred over all other model variants (**Figure 5C**; $p < 0.01$ relative to chance). Notably, the inclusion of second-order modulation analysis yielded a modest yet significant improvement over the half-octave spaced first-order modulation, which is comparable to that developed by McDermott and Simoncelli (2011). The results from the preference experiment revealed which textures benefited most from second-order amplitude modulation analysis. **Figure 5D** shows a list of the top 8 most preferred and least preferred textures measured between the half-octave spaced filterbank and our auditory texture model. The list includes a broad range of sounds, from mechanical/machine noises to animal/insect sounds. The least preferred textures reveal sounds which may not depend greatly on amplitude modulation texture statistics (i.e., cochlear envelope marginal moments and pair-wise correlations).

Two example textures, *helicopter* and *frogs-crickets*, are shown in **Figure 6**. For each texture, the left panel shows the 2nd-order modulation texture statistics for selected bands and the right panel shows the original and synthetic texture cochleograms. Notably, the second-order amplitude modulation power differs between the two textures, suggesting that the additional analysis contributes to sound texture recognition.

Second-Order Modulation Discrimination

To examine if second-order amplitude modulations are processed by the auditory system similarly to textures, i.e., integrated over modest time windows of a few seconds, or if the auditory system has the temporal acuity to identify and discriminate second-order modulations with higher precision, a set of discrimination experiments was performed where synthetic sound textures were compared to artificial control stimuli generated from amplitude modulated Gaussian noise. Listeners performed a three-interval odd-one-out experiment, where they were asked to identify whether the first or last interval was different from the other two. The experiments covered three stimulus groups: rate-specific second-order amplitude modulations, complex second-order amplitude modulation noise from a set of modulation rates, and synthetic sound textures generated using second-order amplitude modulation statistics.

The first experiment included second-order amplitude modulations of increasing rate from 2 to 64 Hz. The results showed that, at low rates, the listeners have the ability to discriminate modulated noise exemplars (**Figure 7—**left panel). The performance decreased with increasing modulation rate and approached chance level for modulation rates above 16 Hz. For these control stimuli, the results suggest that the auditory system may have access the modulation phase for rates 16 Hz and below.

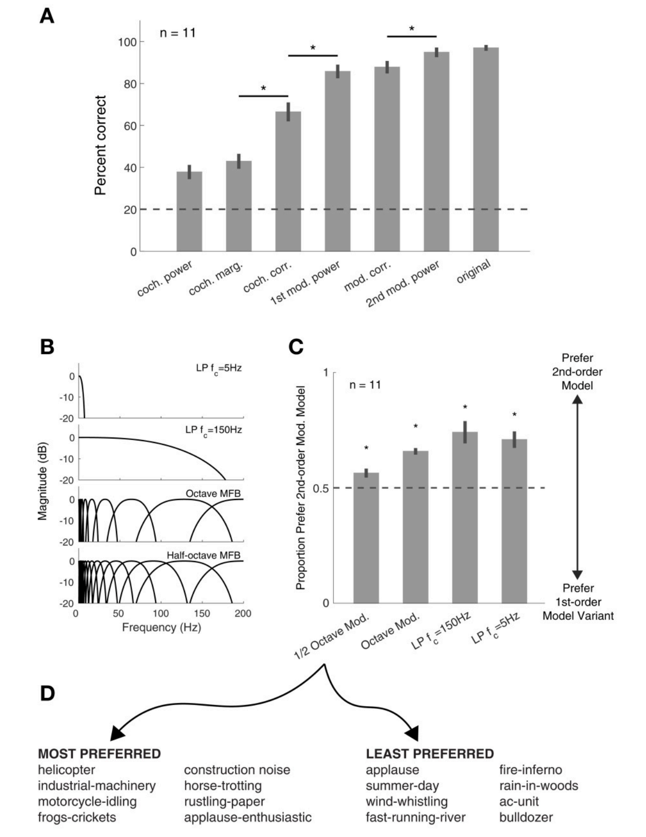


FIGURE 5 | Synthetic texture identification and preference tasks. **(A)** Identification of sound textures improves with the inclusion of more statistics. Asterisks denote significant differences between conditions, $p < 0.01$ (paired t-tests, corrected for multiple comparisons). Error bars here and elsewhere show the standard error. Dashed lines here and elsewhere show chance performance. **(B)** Modulation filter(bank) structure used in the listening experiments. For low-pass (LP) conditions, only the statistics of the signal in the passband were modified. **(C)** Sounds synthesized with the 2nd-order modulation statistics were preferred over all other auditory texture models. Asterisk denotes significance from chance ($p < 0.01$). **(D)** Eight most preferred (left) and least preferred (right) textures from experiment 2, relative to first-order modulation filterbank model (half-octave spacing).

The discriminability of the complex modulated Gaussian noise and the synthetic texture was poor (**Figure 7—**right panel) compared to the low modulation rates considered in the previous experiment. This suggests that, for texture sounds, access to the modulation phase is limited in the auditory system. Isolating the top eight most preferred textures from Experiment 2 revealed comparable performance to the complete set of textures. The performance observed for sound textures in a similar odd-one-out discrimination task was comparable to that reported in McDermott et al. (2013) for an interval duration of about 2 s. Collectively, the results suggest that textures, including those that benefit from second-order modulation analysis, may be perceived using time-average statistics, whereas the auditory system appears to retain more temporal detail for our second-order modulation control stimuli for rates below 16 Hz.

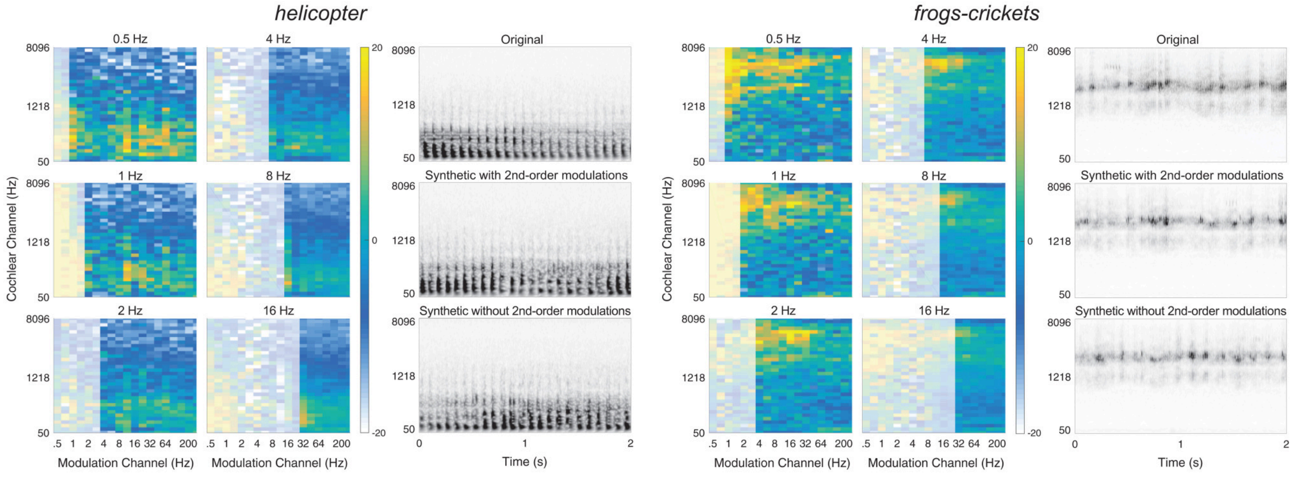


FIGURE 6 | Textures that benefit from second-order modulation statistics. Two example textures from the preferred list: Helicopter (left) and frogs-crickets (right). The left panel shows the second-order modulation statistics for six selected bands. The right panel shows the spectrogram of the original texture (top) and the synthetic texture with second-order modulation statistics (middle) and without second-order modulation statistics (bottom). Example audio files corresponding to the spectrograms of the original, synthetic with 2nd-order modulations, and without 2nd-order modulations can be found in the Supplementary Material (helicopter: **Audio files 10–12**; frogs-crickets: **Audio files 13–15**).

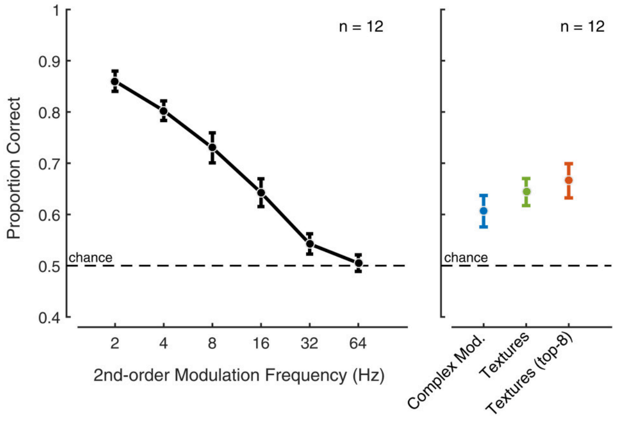


FIGURE 7 | Second-order amplitude modulation and texture exemplar discrimination. The black symbols show the response to second-order amplitude modulated Gaussian noise exemplar discrimination as a function of modulation rate. Error bars indicate the standard error. The blue symbol indicates exemplar discrimination performance for complex second-order amplitude modulated Gaussian noise. The green symbol indicates exemplar discrimination performance for synthetic sound textures that include all indicated texture statistics (including second-order amplitude modulation statistics). The red symbol indicates exemplar discrimination performance for top-8 synthetic (Experiment 2) sound textures that include all indicated texture statistics.

DISCUSSION

The perception of sound texture can be characterized by a set of time-averaged statistics measured from early auditory representations. We extended the auditory texture model of McDermott and Simoncelli (2011) to account for simple rhythmic structures in sound textures via a cascade of amplitude modulation filterbanks. The auditory texture model was coupled with a sound synthesis system to generate texture exemplars from the statistics measured at different stages of the model. The synthetic stimuli were first used in a texture identification experiment, where the listeners' ability to recognize a texture improved with the inclusion of the subgroups of statistics. We found that the performance obtained using the second-order amplitude modulation analysis approached that of the original real-world texture recordings and was higher than the performance obtained using only a first-order amplitude modulation analysis (Experiment 1). We also generated synthetic textures from alternate auditory models of amplitude modulation sensitivity. The synthetic textures were used in a preference task, where listeners' preferred sounds synthesized using second-order amplitude modulation over all other model variants (Experiment 2). Lastly, we performed an experiment focusing on second-order amplitude modulation perception in a discrimination task. The listeners' ability to discriminate second-order modulation sound exemplars decreased with increasing modulation rate, and complex second-order modulated Gaussian noise and synthetic textures appear to be perceived using a time-averaging mechanism (Experiment 3).

Amplitude Modulations in Texture Perception

The auditory texture model described by McDermott and Simoncelli (2011) included a biologically plausible first-order modulation filterbank operating on individual cochlear channel envelopes. The textures synthesized with this model produced many compelling textures, including sounds generated from machinery (e.g., helicopter, printing press) with relatively uniform short-time repetitions as well as environmental sounds (e.g., wind, ocean waves) with variable slow modulations. Our texture model built upon this work and provided further evidence for the importance of modulation selectivity in sound texture perception. For first-order modulation analysis, the results from the preference task (Experiment 2) demonstrated that using half-octave spaced modulation filterbank yields the best performance out of the model variants. The model has a slightly higher selectivity than has that reported in earlier models (Dau et al., 1997). One reason may be that the selectivity of the auditory system for natural sounds, such as textures, may be slightly different than that for artificial stimuli used to identify the auditory systems' modulation tuning curves and selectivity. Another possible explanation is that natural sounds do not conform to octave spaced modulation frequencies, and if the modulation power in a natural sound has a maximum between two modulation bands with fixed center frequencies, the synthetic sounds vary to a greater degree from the original real-world recording.

The results from the preference experiment also identified which textures were most improved (preferred) by the inclusion of the second-order modulation analysis. These textures tended to have higher first-order modulation power, but did not appear to possess obvious common feature. Some sounds, such as the helicopter, had low second-order modulation power while others, such as the frogs-crickets, had high second-order modulation power. Also, the second-order modulation power error between the first-order model and the second-order model did not tend to be higher for these textures. Intuitively, there may be aspects of first-order modulations that are captured by our model, such as mediating the modulation depth in our time-averaged measurements. However, this was difficult to reveal with our natural texture stimuli.

Model Architecture and Statistics

There might be several auditory model architectures that can successfully capture rhythmic structure in sound textures. Our proposed model, using a cascade of modulation filterbanks, seems to provide a compelling approach, as it is relatively intuitive and straight forward to implement in the already established texture analysis-synthesis framework. Another option, however, would be the "venelope" model proposed by Ewert et al. (2002) which used a side-chain analysis to measure the second-order amplitude modulations. In this model, the second-order modulations are extracted from the cochlear envelope and analyzed using a single modulation filterbank. The "venelope" model is more efficient than our cascaded model and there is some evidence to suggest that second-order modulations are processed in the auditory system using the same mechanism as the first-order modulation (Verhey et al., 2003). However, the cascaded modulation filterbank model considered in this study can capture simple rhythmic structure and provided an easier means to reconstruct the filters and thus synthesize textures.

Our approach to modeling of the auditory system, based on audio-frequency and amplitude- modulation-frequency selective filtering, is consistent with biological evidence from the mammalian auditory system (Ruggero, 1992; Joris et al., 2004; Rodríguez et al., 2010). This is found in the auditory-inspired filter structure for both cochlear channels and modulation-selective channels, which culminated in a cascade of filterbanks with intermediate envelope extraction using the Hilbert transform. A similar hierarchical processing architecture has also been well-defined by Mallat and colleagues as scattering moments (Mallat, 2012; Bruna and Mallat, 2013). The scattering moments have been shown to capture a wide range of structure in natural stimuli (Andén and Mallat, 2011, 2012, 2014), in addition to being used for sound texture synthesis (Bruna and Mallat, 2013).

A consequence of the cascaded filterbank model proposed here is that the number of statistics required to capture the auditory feature increases with each layer. This is predominantly the case for the second-order modulation analysis, where we measure 3,400 parameters, which increases the number of texture statistics by a factor of ~3 as compared to the model of McDermott and Simoncelli (2011). It may be possible to optimize the number of parameters by identifying which modulation rates are most significant for texture perception. Alternate models, such as the "venelope" model of Ewert et al. (2002), could reduce the number of parameters needed to capture the second-order amplitude modulation. Although the additional model layer increased the number of statistics, the representation is moderately compact as the statistics are computed as time-averages of the signal.

An alternate approach to representing textures via statistics, is to learn efficient representations from the stimuli themselves. This approach has been shown to be useful for identifying sparse representations of natural stimuli from hierarchical models (Karklin and Lewicki, 2005; Cadieu and Olshausen, 2009). The higher-order structure of natural sounds, such as environmental textures, has also been explored to uncover their possible neural representation (Młynarski and McDermott, 2017). These methods come with their own complications and limitations, however may be a useful avenue for identifying more efficient representations than the texture model of McDermott and Simoncelli (2011) or the one outlined in the present study.

Temporal Regularity in Texture Perception

Sounds textures have been defined as the superposition of many similar acoustic events, therefore it was not obvious *a priori* that sounds with temporal regularities would be perceived in the same way—as time-averages of sensory measurements. Temporal patterns are important for sound perception, and their contribution has been investigated in terms of auditory streaming (Bendixen et al., 2010; Andreou et al., 2011). In addition, sensitivity to temporal regularities in the auditory system has also been shown in complex listening environments (Barascud et al., 2016). Our results show that second-order modulation statistics vary across textures, and the inclusion of this second modulation analysis generated modest improvements in the perceived quality of the synthetic textures. Textures generated with second-order amplitude modulation analysis seemed to result in similar discriminability, suggesting that the features captured by the cascaded modulation filterbank model may be perceived via a similar time-averaging mechanism that has been proposed for more noise-like textures.

Relationship to Visual Texture Perception

One of the interesting ideas about texture perception is that of a unified representation across sensory modalities. Textures have been investigated in the visual system (Julesz, 1962; Portilla and Simoncelli, 2000; Freeman and Simoncelli, 2011), the somatosensory system (Connor and Johnson, 1992) and the auditory system (Saint-Arnaud and Popat, 1995; McDermott and Simoncelli, 2011). Of particular relevance to our work is how the sound texture synthesis system proposed by McDermott and Simoncelli (2011) is comparable in processing structure and analysis to that presented by Portilla and Simoncelli (2000) for visual textures. In both models, the input signal is processed by layers of linear filtering and envelope extraction, while the texture analysis statistics, which are primarily composed of marginal moments and pair-wise correlations, are also similar between the two models. Our model of cascaded filterbanks also overlaps with other models of the image texture perception (Wang et al., 2012). It therefore seems valuable to look across sensory modalities for shared perceptual spaces (Zaidi et al., 2013).

Our investigation of second-order modulation analysis in sound texture perception may also be relatable to spatial texture patterns, or maximally regular textures, in the visual system. Kohler et al. (2016) showed a neural sensitivity to image texture patterns that repeat in space. Our work is also indicative of sound texture pattern sensitivity in time. Previous work in both sound and image texture perception has also made the comparison of perceptual pooling over time and space, respectively (Balas et al., 2009; Freeman and Simoncelli, 2011; McDermott et al., 2013). Conceptually, the apparent texture time-averaging in audition draws compelling parallels to the spatial averaging observed in visual texture perception.

Implications and Perspectives

In this study, we investigated the significance of second-order amplitude modulations in natural sound texture perception. The generation of synthetic sound textures using a cascade of modulation filterbanks appears to contribute positively to the perception of texture. We also observed that the auditory system is sensitive to specific rates of second-order modulations, showing heightened acuity to isolated modulations for rates below 16 Hz. Future experiments would be useful to understand the role of temporal regularity in texture at different modulations rates and spectral frequencies. In addition, such stimuli could be useful to understand the perception of texture in complex auditory scenes, such as the perceptual segregation of speech in the presence of different types of background textures.

ETHICS STATEMENTS

This study was carried out in accordance with the recommendations of Danish Science-Ethics Committee (Den Nationale Videnskabsetiske Komité), Capital Region Committees (De Videnskabsetiske Komitéer for Region Hovedstaden) with written informed consent from all subjects. All subjects gave written informed consent in accordance with the Declaration of Helsinki.

AUTHOR CONTRIBUTIONS

RM performed the experiments and analysis. All authors designed the experiments, interpreted the results, and wrote the paper.

FUNDING

This research was supported by the Technical University of Denmark and the Oticon Centre of Excellence for Hearing and Speech Sciences (CHeSS).

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: http://journal.frontiersin.org/article/10.3389/fnins.2017.00485/full#supplementary-material

Audio file 1 | Figure 3B (Swamp Insects–Original).

Audio file 2 | Figure 3B (Swamp Insects–Synthetic).

Audio file 3 | Figure 3B (Campfire–Original).

Audio file 4 | Figure 3B (Campfire–Synthetic).

Audio file 5 | Figure 3B (Small Stream–Original).

Audio file 6 | Figure 3B (Small Stream–Synthetic).

Audio file 7 | Figure 4A (Original).

Audio file 8 | Figure 4A (w/ 2nd-order mods.).

Audio file 9 | Figure 4A (w/o 2nd-order mods.).

Audio file 10 | Figure 6 (helicopter–Original).

Audio file 11 | Figure 6 (helicopter–Synthetic with 2nd-order modulations).

Audio file 12 | Figure 6 (helicopter–Synthetic without 2nd-order modulations).

Audio file 13 | Figure 6 (frogs-crickets–Original).

Audio file 14 | Figure 6 (frogs-crickets–Synthetic with 2nd-order modulations).

Audio file 15 | Figure 6 (frogs-crickets–Synthetic without 2nd-order modulations).

REFERENCES

Andén, J., and Mallat, S. (2011). "Multiscale scattering for audio classification," in *ISMIR* (Miami, FL), 657–662.

Andén, J., and Mallat, S. (2012). "Scattering representation of modulated sounds," in *Proceedings of the 15th International Conference on Digital Audio Effects* (New York, NY).

Andén, J., and Mallat, S. (2014). Deep Scattering Spectrum. *IEEE Trans. Signal Process.* 62, 4114–4128. doi: 10.1109/TSP.2014.2326991

Andreou, L. V., Kashino, M., and Chait, M. (2011). The role of temporal regularity in auditory segregation. *Hear. Res.* 280, 228–235. doi: 10.1016/j.heares.2011.06.001

Balas, B., Nakano, L., and Rosenholtz, R. (2009). A summary-statistic representation in peripheral vision explains visual crowding. *J. Vis.* 9, 13.1–13.18. doi: 10.1167/9.12.13

Barascud, N., Pearce, M. T., Griffiths, T. D., Friston, K. J., and Chait, M. (2016). Brain responses in humans reveal ideal observer-like sensitivity to complex acoustic patterns. *Proc. Natl. Acad. Sci. U.S.A.* 113, E616–E625. doi: 10.1073/pnas.1508523113

Bendixen, A., Denham, S. L., Gyimesi, K., and Winkler, I. (2010). Regular patterns stabilize auditory streams. *J. Acoust. Soc. Am.* 128, 3658–3666. doi: 10.1121/1.3500695

Bolcskei, H., Hlawatsch, F., and Feichtinger, H. G. (1998). Frame-theoretic analysis of oversampled filter banks. *IEEE Trans. Signal Process.* 46, 3256–3268. doi: 10.1109/78.735301

Bruna, J., and Mallat, S. (2013). Invariant scattering convolution networks. *IEEE Trans. Pattern Anal. Mach. Intell.* 35, 1872–1886. doi: 10.1109/TPAMI.2012.230

Cadieu, C., and Olshausen, B. A. (2009). "Learning transformational invariants from natural movies," in *Advances in Neural Information Processing Systems* (Vancouver, BC), 209–216.

Connor, C. E., and Johnson, K. O. (1992). Neural coding of tactile texture: comparison of spatial and temporal mechanisms for roughness perception. *J. Neurosci.* 12, 3414–3426.

Dau, T., Kollmeier, B., and Kohlrausch, A. (1997). Modeling auditory processing of amplitude modulation 1. Detection and masking with narrow-band carriers. *J. Acoust. Soc. Am.* 102, 2892–2905. doi: 10.1121/1.418727

Dau, T., Püschel, D., and Kohlrausch, A. (1996). A quantitative model of the "effective" signal processing in the auditory system. I. Model structure. *J. Acoust. Soc. Am.* 99, 3615–3622. doi: 10.1121/1.414959

Ewert, S. D., Verhey, J. L., and Dau, T. (2002). Spectro-temporal processing in the envelope-frequency domain. *J. Acoust. Soc. Am.* 112, 2921–2931. doi: 10.1121/1.1515735

Field, D. J. (1987). Relations between the statistics of natural images and the response properties of cortical cells. *J. Opt. Soc. Am. A* 4, 2379–2394. doi: 10.1364/JOSAA.4.002379

Freeman, J., and Simoncelli, E. P. (2011). Metamers of the ventral stream. *Nat. Neurosci.* 14, 1195–1201. doi: 10.1038/nn.2889

Füllgrabe, C., Moore, B. C. J., Demany, L., Ewert, S. D., Sheft, S., and Lorenzi, C. (2005). Modulation masking produced by second-order modulators. *J. Acoust. Soc. Am.* 117, 2158–2168. doi: 10.1121/1.1861892

Glasberg, B. R., and Moore, B. C. J. (1990). Derivation of auditory filter shapes from notched-noise data. *Hear. Res.* 47, 103–138. doi: 10.1016/0378-5955(90)90170-T

Irino, T., and Patterson, R. D. (2006). A dynamic compressive gammachirp auditory filterbank. *IEEE Trans. Audio Speech Lang. Process.* 14, 2222–2232. doi: 10.1109/TASL.2006.874669

Jorgensen, S., and Dau, T. (2011). Predicting speech intelligibility based on the signal-to-noise envelope power ratio after modulation-frequency selective processing. *J. Acoust. Soc. Am.* 130, 1475–1487. doi: 10.1121/1.3621502

Joris, P. X., Schreiner, C. E., and Rees, A. (2004). Neural processing of amplitude-modulated sounds. *Physiol. Rev.* 84, 541–577. doi: 10.1152/physrev.00029.2003

Julesz, B. (1962). Visual pattern discrimination. *IRE Trans. Information Theor.* 8, 84–92. doi: 10.1109/TIT.1962.1057698

Karklin, Y., and Lewicki, M. S. (2005). A hierarchical Bayesian model for learning nonlinear statistical regularities in nonstationary natural signals. *Neural Comput.* 17, 397–423. doi: 10.1162/0899766053011474

Kohler, P. J., Clarke, A., Yakovleva, A., Liu, Y., and Norcia, A. M. (2016). Representation of maximally regular textures in human visual cortex. *J. Neurosci.* 36, 714–729. doi: 10.1523/JNEUROSCI.2962-15.2016

Kohlrausch, A., Fassel, R., and Dau, T. (2000). The influence of carrier level and frequency on modulation and beat-detection thresholds for sinusoidal carriers. *J. Acoust. Soc. Am.* 108, 723–734. doi: 10.1121/1.429605

Lorenzi, C., Simpson, M. I., Millman, R. E., Griffiths, T. D., Woods, W. P., Rees, A., et al. (2001a). Second-order modulation detection thresholds for pure-tone and narrow-band noise carriers. *J. Acoust. Soc. Am.* 110, 2470–2478. doi: 10.1121/1.1406160

Lorenzi, C., Soares, C., and Vonner, T. (2001b). Second-order temporal modulation transfer functions. *J. Acoust. Soc. Am.* 110, 1030–1038. doi: 10.1121/1.1383295

Mallat, S. (2012). Group Invariant Scattering. *Commun. Pure Appl. Math.* 65, 1331–1398. doi: 10.1002/cpa.21413

Malone, B. J., Beitel, R. E., Vollmer, M., Heiser, M. A., and Schreiner, C. E. (2015). Modulation-frequency-specific adaptation in awake auditory cortex. *J. Neurosci.* 35, 5904–5916. doi: 10.1523/JNEUROSCI.4833-14.2015

McDermott, J. H., and Simoncelli, E. P. (2011). Sound texture perception via statistics of the auditory periphery: evidence from sound synthesis. *Neuron* 71, 926–940. doi: 10.1016/j.neuron.2011.06.032

McDermott, J. H., Schemitsch, M., and Simoncelli, E. P. (2013). Summary statistics in auditory perception. *Nat. Neurosci.* 16, 493–498. doi: 10.1038/nn.3347

Miller, L. M., Escabi, M. A., Read, H. L., and Schreiner, C. E. (2002). Spectrotemporal receptive fields in the lemniscal auditory thalamus and cortex. *J. Neurophysiol.* 87, 516–527. doi: 10.1152/jn.00395.2001

Młynarski, W., and McDermott, J. H. (2017). Learning mid-level auditory codes from natural sound statistics. *arXiv preprint arXiv:1701.07138.*

Portilla, J., and Simoncelli, E. P. (2000). A parametric texture model based on joint statistics of complex wavelet coefficients. *Int. J. Comput. Vis.* 40, 49–70. doi: 10.1023/A:1026553619983

Rodríguez, F. A., Chen, C., Read, H. L., and Escabí, M. A. (2010). Neural modulation tuning characteristics scale to efficiently encode natural sound statistics. *J. Neurosci.* 30, 15969–15980. doi: 10.1523/JNEUROSCI.0966-10.2010

Ruggero, M. A. (1992). Responses to sound of the basilar membrane of the mammalian cochlea. *Curr. Opin. Neurobiol.* 2, 449–456. doi: 10.1016/0959-4388(92)90179-O

Saint-Arnaud, N., Popat, K. (1995). "Analysis and synthesis of sound textures," in *Computational Auditory Scene Analysis*, eds D. F. Rosenthal and H. G. Okuno (L. Erlbaum Associates Inc.) ,293–308.

Verhey, J. L., Ewert, S. D., and Dau, T. (2003). Modulation masking produced by complex tone modulators. *J. Acoust. Soc. Am.* 114(4 Pt 1), 2135–2146. doi: 10.1121/1.1612489

Viemeister, N. F. (1979). Temporal modulation transfer functions based upon modulation thresholds. *J. Acoust. Soc. Am.* 66, 1364–1380. doi: 10.1121/1.383531

Wang, H. X., Heeger, D. J., and Landy, M. S. (2012). Responses to second-order texture modulations undergo surround suppression. *Vision Res.* 62, 192–200. doi: 10.1016/j.visres.2012.03.008

Zaidi, Q., Victor, J., McDermott, J., Geffen, M., Bensmaia, S., and Cleland, T. A. (2013). Perceptual spaces: mathematical structures to neural mechanisms. *J. Neurosci.* 33, 17597–17602. doi: 10.1523/JNEUROSCI.3343-13.2013

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Neuroscience

ORIGINAL RESEARCH
published: 29 May 2018
doi: 10.3389/fnins.2018.00349

Preferred Tempo and Low-Audio-Frequency Bias Emerge From Simulated Sub-cortical Processing of Sounds With a Musical Beat

Nathaniel J. Zuk[1]*, *Laurel H. Carney*[1,2] *and Edmund C. Lalor*[1,2,3,4]

[1] Department of Biomedical Engineering, University of Rochester, Rochester, NY, United States, [2] Department of Neuroscience, University of Rochester Medical Center, Rochester, NY, United States, [3] Del Monte Institute for Neuroscience, University of Rochester Medical Center, Rochester, NY, United States, [4] Trinity Centre for Bioengineering, Trinity College Dublin, Dublin, Ireland

OPEN ACCESS

Edited by:
Mounya Elhilali,
Johns Hopkins University,
United States

Reviewed by:
Michael Schwartze,
Maastricht University, Netherlands
Hugo Merchant,
Universidad Nacional Autónoma de México, Mexico
Aniruddh Patel,
Tufts University, United States

Jonathan Cannon contributed to the review of Aniruddh Patel

****Correspondence:***
Nathaniel J. Zuk
nzuk@ur.rochester.edu

Specialty section:
This article was submitted to Neuromorphic Engineering, a section of the journal Frontiers in Neuroscience

Received: *13 January 2018*
Accepted: *07 May 2018*
Published: *29 May 2018*

Citation:
Zuk NJ, Carney LH and Lalor EC (2018) Preferred Tempo and Low-Audio-Frequency Bias Emerge From Simulated Sub-cortical Processing of Sounds With a Musical Beat. Front. Neurosci. 12:349. doi: 10.3389/fnins.2018.00349

Prior research has shown that musical beats are salient at the level of the cortex in humans. Yet below the cortex there is considerable sub-cortical processing that could influence beat perception. Some biases, such as a tempo preference and an audio frequency bias for beat timing, could result from sub-cortical processing. Here, we used models of the auditory-nerve and midbrain-level amplitude modulation filtering to simulate sub-cortical neural activity to various beat-inducing stimuli, and we used the simulated activity to determine the tempo or beat frequency of the music. First, irrespective of the stimulus being presented, the preferred tempo was around 100 beats per minute, which is within the range of tempi where tempo discrimination and tapping accuracy are optimal. Second, sub-cortical processing predicted a stronger influence of lower audio frequencies on beat perception. However, the tempo identification algorithm that was optimized for simple stimuli often failed for recordings of music. For music, the most highly synchronized model activity occurred at a multiple of the beat frequency. Using bottom-up processes alone is insufficient to produce beat-locked activity. Instead, a learned and possibly top-down mechanism that scales the synchronization frequency to derive the beat frequency greatly improves the performance of tempo identification.

Keywords: auditory, rhythm, tempo induction, musical beat, biomimetic model

INTRODUCTION

When we spontaneously tap our feet to music, we are "feeling the beat." A musical beat is frequently defined by the effect it has on motor entrainment (Patel, 2010; London, 2012), and it is often identified as the fundamental level in the metrical hierarchy for keeping time (Lerdahl and Jackendoff, 1983). Many cultures have music with a beat, and the presence of beat-based music is highly related to communal dance (Savage et al., 2015). Clearly, perceiving the beat is key to the perception of music.

In many genres of music, musical beats often, but not always, occur at isochronous intervals (London, 2012). Previous models have simulated the perception of isochronous beats using an internal clock (Povel and Essens, 1985), pattern matching (Rosenthal, 1992; Parncutt, 1994), an internal resonator (van Noorden and Moelants, 1999), or a bank of neural oscillators (Large et al., 2015). These models often compute the beat frequency of discrete pulses, although a few have used annotated performances as input (ex. Rosenthal, 1992) or "onset signals" computed from cochlear-like filtering of audio signals (Scheirer, 1998; Large, 2000). Using electroencephalography and magnetoencephalography, it has been shown that cortical activity time-locks to the perceived beat for simplistic stimuli (Snyder and Large, 2005; Iversen et al., 2009; Nozaradan et al., 2011, 2012; Fujioka et al., 2012, 2015; Tierney and Kraus, 2015; Tal et al., 2017; but see Henry et al., 2017). Yet multiple stages of processing occur prior to cortical processing, each of which could affect the placement of musical beats.

Even for basic acoustic events, human subjects are biased to tapping to beats at inter-onset intervals of 500 to 700 ms (Parncutt, 1994), equivalent to a tempo range of 85 to 120 BPM. This range encompasses the "indifference interval" (Fraisse, 1963; London, 2012) for which subjects tap naturally at a regular rhythm (Semjen et al., 1998), discriminate tempi best (Drake and Botte, 1993), and can best replicate the duration of the interval (Stevens, 1886; Woodrow, 1934) (for review see Fraisse, 1963; Patel, 2010; London, 2012). This range also overlaps the range of tempi for a large proportion of dance music, which centers on 450 to 600 ms for intervals between beats, or equivalently 100 to 133 BPM (van Noorden and Moelants, 1999). However, an explanation for this optimal range of tempi is unclear. Motor entrainment plays a role in this bias since subjects tap naturally within this range, but it does not completely explain the optimization observed in studies that do not involve motor entrainment. Modulation tuning in the sub-cortical central nervous system would affect the synchronization strength of neural activity to isochronous acoustic events, which in turn could influence the preferred tempo.

Additionally, there is some evidence that our perception of musical beats is biased to certain ranges of audio frequencies. Listeners' ratings of "groove" in music, a subjective quality related to how much people want to move to the music, is correlated with the fluctuation in energy in low frequency (<200 Hz) and mid-frequency (400–1600 Hz) bands (Stupacher et al., 2016). Subjects also identify beats in piano ragtime music better when the left hand (lower frequency) is played alone than when the right hand (higher frequency) is played alone, although this could be due to the regularity of the left hand for this type of music (Snyder and Krumhansl, 2001). A low-frequency bias for beat timing could result from cochlear processing, where low frequencies cause a greater spread of excitation than higher frequencies (Hove et al., 2014), but these effects need to be disambiguated from cochlear delays that can produce similar biasing effects for simultaneous events (Wojtczak et al., 2017). For repeating "frozen" noise, where the noise signal was identical on each repetition, subjects focus on mid-frequency perturbations in the signal, between 300 and 2,000 Hz, when tapping along with the repetition (Kaernbach, 1993). Overall, while there does appear to be a frequency bias for time locking beats in music and repeating sounds, the exact frequency range of the bias, and the influence of subcortical processing on the bias, is still unclear.

Separately, several groups have developed "tempo-induction" algorithms that identify the tempo of musical recordings (for review see Gouyon et al., 2006; McKinney et al., 2007). These algorithms typically consist of three stages: identify onsets in the music, determine the pattern and speed of those onsets, and determine the tempo based on several representations of these factors (ex. Elowsson and Friberg, 2015). While some of these algorithms use processes that are similar to the auditory system (ex. Scheirer, 1998), none have been built on biomimetic models of auditory processing that simulate the neural activity produced by stages of auditory processing below the cortex. This processing is important because beat perception is based on the neural activity projected to the cortex. Both physiological modulation tuning and the inherent randomness of neural signals present in realistic auditory processing could affect beat perception in real music.

Here, we developed a model that determines the tempo of recordings of music based on the simulated neural activity of the auditory nerve and amplitude modulation tuning in the brainstem and midbrain. We hypothesized that physiologically plausible synaptic processing, which results in amplitude modulation tuning in the midbrain, can impose a preferred tempo near 100 BPM (London, 2012). We also hypothesized that innate processing in the auditory nerve can explain our low-audio-frequency bias for timing musical beats. Lastly, we quantify the strength of neural synchronization to musical beats in musical recordings and assess different ways in which the beat frequency may be inferred based on sub-cortical processing.

MATERIALS AND METHODS

Modeling

Sub-cortical neural activity was simulated using a cascade of two biomimetic models for different stages of auditory processing. The sound input was converted to time-varying firing rates using a model of auditory-nerve (AN) fibers (Zilany et al., 2014) (**Figure 1**). Each AN fiber was tuned to a particular characteristic frequency (CF). The bandwidths of the model AN fibers matched human cochlear tuning (Shera et al., 2002). High-spontaneous-rate AN fibers were simulated with CFs from 125 to 8 kHz spaced every 0.05 octaves (121 fibers total). This model includes cochlear compression and firing rate adaptation (Zhang et al., 2001; Zilany et al., 2009). Our focus was on high-spontaneous-rate AN fibers because of their predominance in the auditory nerve (Liberman, 1978). Additionally, high-spontaneous-rate fibers alone can encode speech across a wide range of sound levels and in noisy environments (Carney et al., 2015), suggesting that they might also be especially important for encoding acoustic events relevant for musical beat perception.

The time-varying AN fiber firing rate was filtered using a model of synaptic processing in the ventral cochlear nucleus (VCN) and the inferior colliculus (IC) (Nelson and Carney, 2004; Carney et al., 2015). The model produces bandpass modulation sensitivity via two-stage same-frequency inhibition and excitation (SFIE), where the time constants, delays, and

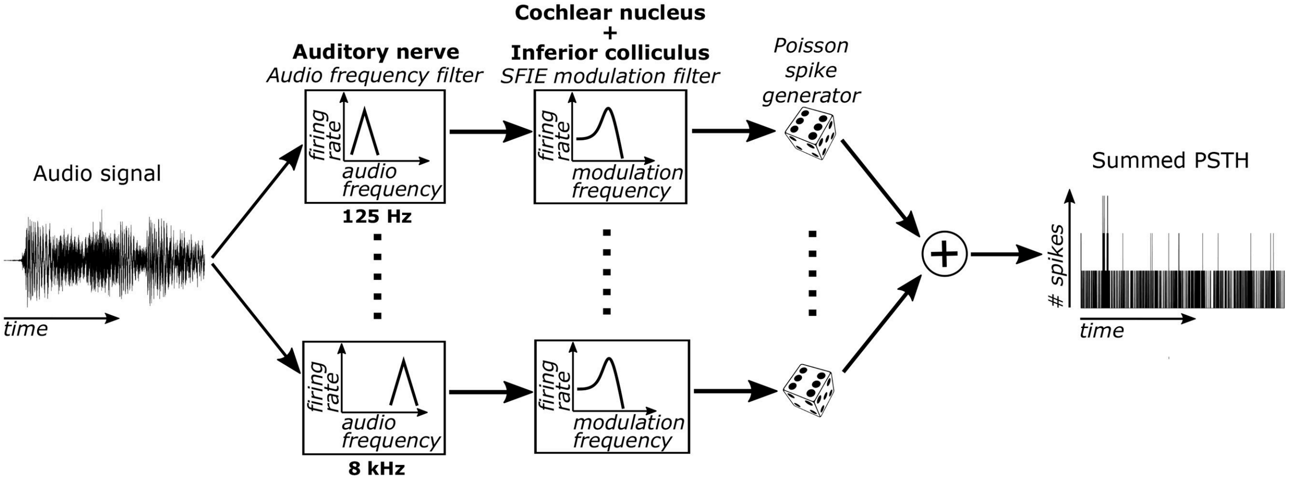


FIGURE 1 | The model used to simulate sub-cortical neural activity consisted of three stages. First, the sound was filtered through 121 model AN fibers, each of which include bandpass filtering from the basilar membrane, compression due to the outer hair cells, and firing rate adaptation. Second, the output firing rates of these AN fibers were filtered using an SFIE model that simulated processing in the VCN and IC. Lastly, neural activity was simulated for each CF using the output time varying firing rate of the second stage. The simulated activity was then summed across CFs to get the summed PSTH.

strengths of the inputs affect the neuron's best modulation frequency (**Figure 2**). The SFIE model also accentuates onset responses in the firing rate function that are akin to neural responses in the inferior colliculus or the medial geniculate body of the thalamus (Rouiller et al., 1981; Krishna and Semple, 2000; Bartlett and Wang, 2007; Nelson and Carney, 2007). We varied the SFIE model parameters (**Table 1**) to examine their effects on the strength of synchronization to a range of tempi.

For each of the 121 CFs, we randomly generated spike trains in response to each stimulus, assuming that the spike times obey an inhomogeneous Poisson process (Brown et al., 2002) with a time-varying rate parameter determined by the output of the SFIE stage. The spike trains across CF were then summed to form a post-stimulus time histogram (PSTH) for each response to a stimulus.

We hypothesized that the beat frequency of the stimulus could be determined based on the phase-locking of the PSTH to the beat frequency. The PSTH was first filtered using a Gaussian-shaped temporal smoothing window. The shape of the window was based on prior results showing a Gaussian-like variation in performance for detecting events that deviate from isochronous intervals (Repp and Penel, 2002). Periodicities in the PSTH were then identified by taking the Fourier transform of the PSTH and normalizing by the average value of the PSTH (or the magnitude of the Fourier component at 0 Hz) (**Figure 3A**). This value is computationally identical to the "vector strength" which quantifies the synchronization strength of neural activity to a particular frequency (Goldberg and Brown, 1969). The model's "synchronization tempo" was the tempo where the vector strength was maximal.

In the Fourier domain the temporal smoothing window imposed a low-pass filter on the vector strength and thus suppressed the vector strength of fast tempi (**Figure 3B**). Several studies have demonstrated that the upper limit of the human perception of isochrony occurs at inter-onset intervals around 100 ms (for review see Repp, 2005; London, 2012). To enforce this upper limit, the standard deviation (σ) of the temporal window was empirically set to 40 ms because it was the minimum σ such that the vector strength for isochronous clicks at 600 BPM (inter-onset interval of 100 ms) was no larger than the vector strength for 100% jittered clicks at the same average rate (Supplementary Figure 1). The temporal window width of 40 ms was used for all SFIE models examined.

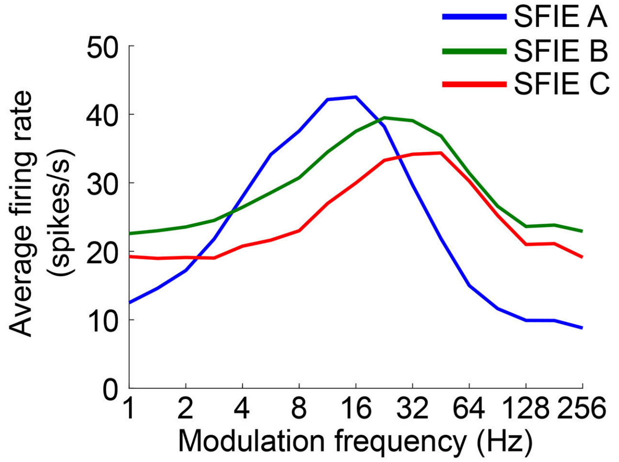


FIGURE 2 | The rate modulation transfer functions for the three SFIE models we examined. The functions were computed by averaging the firing rate of the output of the SFIE model using a single input AN fiber (CF = 800 Hz) for 4 s of sinusoidally amplitude modulated broadband noise repeated 20 times. The parameters for each of the SFIE models can be found in **Table 1**.

Throughout, all stimuli were set to 70 dB SPL and were up-sampled to a 100 kHz sampling rate, which was required for the

AN fiber model. For stimuli that started or ended with a non-zero signal (for example, amplitude modulated noise), 15 ms raised-sine ramps were applied to the start and end of the stimulus.

Stimuli and Hypotheses for Preferred Tempo Analyses

Stimuli were 10 s long and consisted of either 1 ms clicks (0.5 condensation followed by 0.5 ms rarefaction), sinusoidally amplitude modulated (SAM) broadband noise (0–50 kHz), square wave (SW) modulated broadband noise with a duty cycle of 50%, and raised-sine 100-ms-long tone pips with carrier frequencies of 250 Hz, 1 kHz, or 4 kHz. The tempo was varied from 30 BPM to 600 BPM in 30 BPM steps, and each stimulus was presented 10 times. The phase of the stimulus modulation was randomized for each presentation. The preferred tempo was determined for each type of stimulus using quadratic interpolation. To evaluate the effects of the SFIE model on this result, the analysis was repeated for each type of SFIE unit and also for the summed activity of the AN fibers alone.

TABLE 1 | Parameters used for each two-stage SFIE model (see Nelson and Carney, 2004; Carney et al., 2015).

	SFIE model parameters				
	τ_{exc} (ms)	τ_{inh} (ms)	S_{inh}	d_{inh} (ms)	A
VCN stage	0.5	2	0.6	1	1.5
IC A	5	10	1.1	2	6
IC B	2	6	1.1	2	2
IC C	1	3	1.5	2	2

The parameters in the ventral cochlear nucleus (VCN) stage, the first stage of the model, were always used. The parameters for the second stage of the model, inferior colliculus (IC) model, was varied.

Several studies have demonstrated that humans' ability to perceive and reproduce regular events is optimized for inter-onset intervals around 600 ms, corresponding to a tempo of 100 BPM (London, 2012). We hypothesized that the modulation filtering of the SFIE model and the temporal smoothing window could produce a vector strength maximum around 100 BPM. Additionally, Henry et al. (2017) showed that the strength of perceived musical beats is independent of the envelope of the stimulus. Based on this, we expected the tempo exhibiting the maximum vector strength (the "preferred tempo") to remain the same irrespective of the stimulus being used.

Assessing a Frequency Bias for Tempo Induction

To identify a frequency bias in tempo induction that could result from subcortical processing, we presented the model with stimuli consisting of two stimulus trains of 100 ms raised-sine tone pips presented at two different tempi (from the range 60 to 180 BPM) and two different frequencies (from the range 125 to 8,000 Hz) (an example stimulus can be found in **Figure 6A**). The tempi, frequencies, and phases of the two tones were randomly selected to generate 1000 different stimuli, and each stimulus was presented once. The frequencies of the two

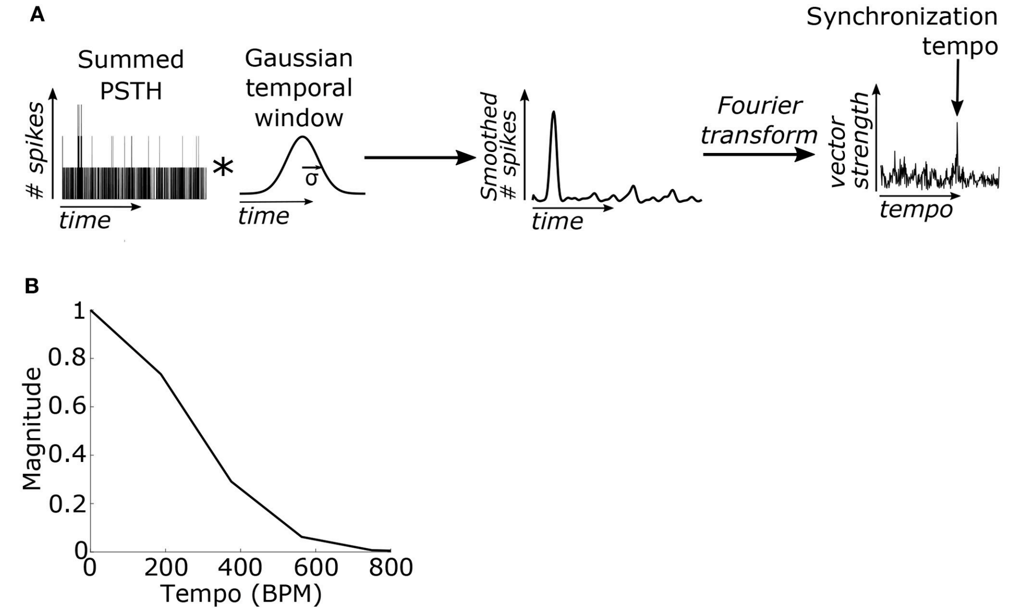


FIGURE 3 | (A) The summed PSTH was convolved (represented by an asterisk) with a Gaussian-shaped temporal smoothing window with a standard deviation of 40 ms (see Materials and Methods). Then the Fourier transform of the smoothed PSTH was used to compute the vector strength of the neural activity, which quantifies the strength of synchronization, at each tempo. The "synchronization tempo" using this method was equal to the tempo with the peak vector strength between 30 and 600 BPM. **(B)** The Fourier transform of the temporal smoothing window. The temporal smoothing window smooths the PSTH and suppresses the vector strengths at high tempi.

tone pips were spaced at least one octave apart to reduce AN adaptation effects (Zilany et al., 2009) that could produce cross-frequency forward masking. For each stimulus, we computed the normalized synchronization tempo (NST):

$$NST = \frac{T_{sync} - T_L}{T_H - T_L}$$

where T_L and T_H are the tempi for the low carrier frequency and high carrier frequency pulse trains, respectively, and T_{sync} is the synchronization tempo.

We expected the synchronization tempo to be close to the tempo of either the tone pips with the low-frequency carrier or the high-frequency carrier for most of the stimuli, resulting in an NST near either zero or one, respectively. Of those stimuli, we next examined how the other factors, the tempi of the two tone pips and their carrier frequencies, affected the NST. A logistic generalized linear model was fit to the NST values that were within ±0.08 of either zero or one (807/1000 trials) using fitglm in Matlab:

$$P\left(NST = 0|\mu\right) = \frac{e^{\mu}}{e^{\mu} + 1}$$

where:

$$\mu = \beta_0 + \beta_{f_L}\ (\frac{f_L}{125}) + \beta_{f_H} \log_2\ (\frac{f_H}{125}) + \beta_{T_L} T_L + \beta_{T_H} T_H$$

where f_L and f_H are the carrier frequencies of the low and high frequency tone pips respectively, and the beta values quantify the linear dependence between each parameter and the probability that the NST equals one. If the NST was independent of the stimulus parameters, then the model should not do significantly better than a constant model ($\mu = \beta_0$). The significance of this difference was assessed using a likelihood ratio test. The significance of the individual coefficients in the model was also assessed using a likelihood ratio test comparing the full model to a reduced model with each component removed individually.

Tempo Induction of Real Music

Lastly, we examined how well this model could correctly identify tempi for two datasets of music: a "Ballroom" dataset of 685 clips of ballroom dance music (after removing exact and recording replicates, see Sturm, 2014), and a "Songs" dataset of 465 clips of music from a wide variety of genres and cultures, including some dance music (Gouyon et al., 2006). These datasets are standards for assessing the performance of tempo-induction and beat-detection algorithms (Gouyon et al., 2006; McKinney et al., 2007). We determined the synchronization tempo based on the tempo between 30 and 600 BPM with the maximum vector strength. Throughout, the synchronization tempo was identified as correct if it fell within ±8% of the ground truth tempo (standard for the MIREX tempo induction competition, see McKinney et al., 2007).

Computing the Tempo Using a Classifier

Often, the peak vector strength occurred at a multiple of the ground truth tempo rather than at the actual ground truth tempo. One possibility is that we "feel the beat" for every 2–4 events depending upon the speed of the music (Parncutt, 1994; London, 2012). Additionally, we may be using the pattern of events in the music, or the "rhythm", to determine the beat frequency, since beat perception is affected by rhythm (Povel and Essens, 1985; Parncutt, 1994). To understand the importance of speed and rhythm on tempo induction, we used regularized multi-class linear discriminant analysis (mcLDA) (fitcdiscr.m in Matlab, other classification algorithms did not perform as well) to develop two different classifiers that identify the "scaling factor" equal to the ratio of the synchronization tempo to the ground truth tempo, either 1, 2, 3, or 4. The first classifier used the synchronization tempo alone to classify the scaling factor; faster synchronization tempi were more likely to have higher scaling factors. For the second classifier, we reasoned that, if the model neurons were synchronizing to events in the music, then the rhythm of the music could be quantified by the number of times certain intervals appear between simulated spikes. The within- and across-CF interspike interval (ISI) histogram for the summed neural activity was computed using the autocorrelation of the summed PSTH, and the "ISI ratio" for a particular interval was computed by summing the ISIs within a 20 ms rectangular window surrounding the interval and dividing by the total number of ISIs between 0.1 and 30 s. The ISI ratio was computed for ISIs at the following multiples of the event period: 1/16, 1/12, 1/9, 1/6, 1/4, 1/3, 1/2, 2/3, 3/4, and 1. All stimuli from both datasets were included in this analysis, and the ratios were rounded to closest integer between 1 and 4. This classification procedure was repeated for 1000 random re-samplings of the stimuli, selecting 75% of the stimuli for training and 25% for testing. We determined whether the second classifier performed significantly better than the first by testing the null hypothesis that the distribution of differences in performance between the two classifiers for the 1000 re-samplings was no greater than 0.

RESULTS

Dependence of Model Vector Strength on Stimulus Tempo

Firstly, we examined if the vector strength of the model PSTH was maximal over a specific range of tempi. We hypothesized that sub-cortical processing could contribute to this biasing, which has been observed around 100 BPM. The vector strength as a function of tempo was computed using three different midbrain models (**Table 1**) that were tuned to different best modulation frequencies (**Figure 2**). For comparison, the vector strength was also computed based on the unfiltered summed AN fiber output.

While the temporal smoothing window suppressed vector strengths at high tempi (**Figure 3B**), there was also a reduction in vector strengths at low tempi due to an intrinsic property of the auditory nerve model. **Figure 4** shows examples of the summed firing rate across CF for different SFIE models, which was the input to the Poisson spike generator (**Figure 1**). For click trains at 30 BPM (**Figure 4A**), SFIE model A generated the largest firing rates in response to a click, but it also produced the highest spontaneous rate, resulting in the lowest vector strength of the three midbrain models. For SAM noise (**Figure 4B**), the firing

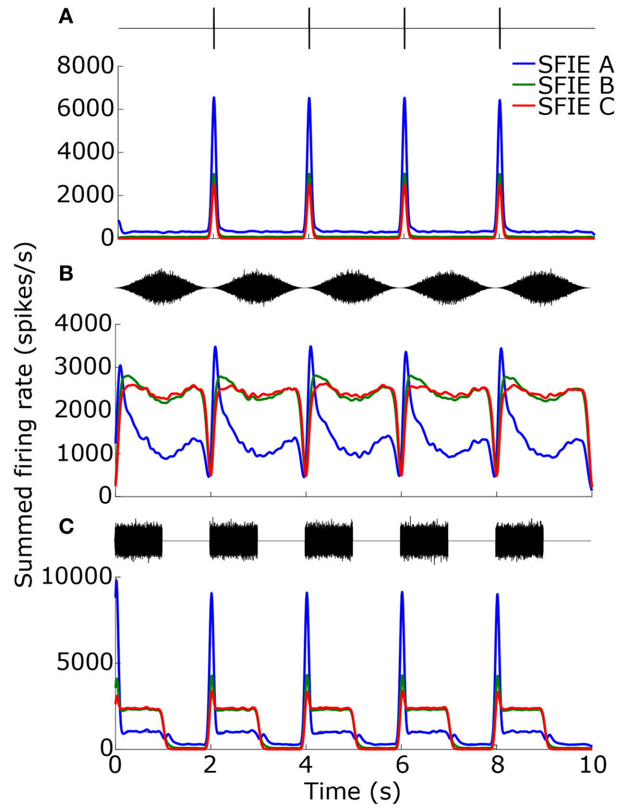


FIGURE 4 | Firing rates for the different SFIE models in response to 1 ms clicks **(A)**, SAM broadband noise **(B)**, and SW noise **(C)** at 30 BPM (SFIE A: blue, SFIE B: green, SFIE C: red). The corresponding stimulus is shown above each plot of the firing rate. All stimuli were presented at 70 dB SPL. The firing rates were summed across CF and averaged across 10 repetitions of each stimulus with different noise tokens. Spontaneous firing during silences **(A,C)** and saturating firing rates during continuous noises **(A,B)** contributed to a falloff in vector strength at lower tempi (see **Figure 5**).

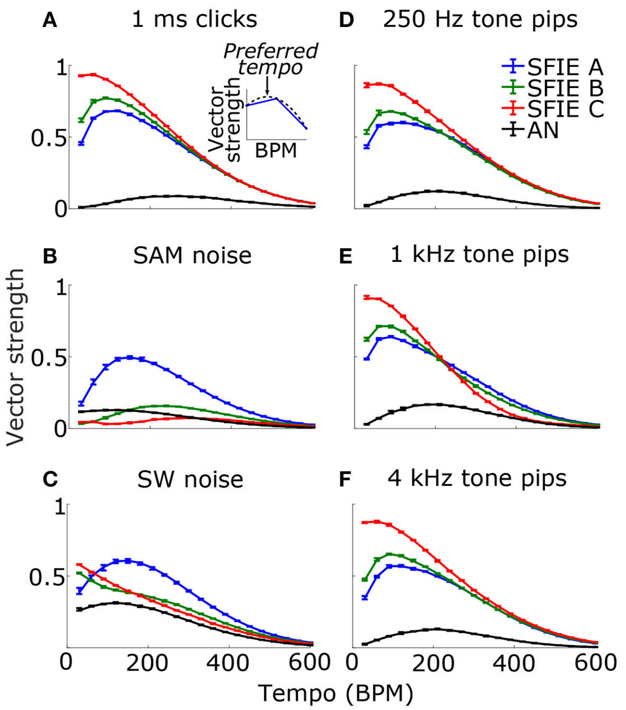


FIGURE 5 | Vector strength as a function of tempo in response to 1 ms clicks **(A)**, SAM broadband noise **(B)**, SW broadband noise **(C)**, and tone pips with carrier frequencies of 250 Hz **(D)**, 1 kHz **(E)**, and 4 kHz **(F)**. The vector strengths for the different SFIE models are color coded identically to **Figure 4**. Error bars designate interquartile ranges for 10 repetitions of each stimulus. The vector strength using the AN fiber activity alone, without an SFIE stage, is also shown in black. SFIE model A consistently produced peak vector strengths within the range of tempi typically associated with the "indifference interval" (around 100 BPM) and overlapping the range of tempi for dance music (van Noorden and Moelants, 1999). The preferred tempos were determined by quadratic interpolation. The black dashed line in the inset in **(A)** shows the quadratic fit to the points surrounding the maximum vector strength for SFIE A. The preferred tempo is equal to the peak of the quadratic fit. Preferred tempos and peak vector strengths are quantified in **Table 2**.

rates of high-spontaneous rate AN fibers saturated at moderate sound levels, resulting in saturating SAM responses for moderate to high SPLs which reduced their synchronization strength (see also Joris et al., 2004). The saturating responses were maintained for the models with high peak modulation frequencies, SFIE B and C. In contrast, SFIE A showed a stronger onset response during the rising phase of the stimulus modulation followed by a reduction in firing during the rest of the cycle of the modulation. As a result, SFIE A had a larger vector strength than the other two models. For SW noise (**Figure 4C**), the response for model SFIE A showed both a suppression of sustained firing as well as high spontaneous firing.

Across a wide variety of stimuli (clicks, SAM noise, SW noise, tone pips), SFIE A consistently produced preferred tempi between 86 and 150 BPM (**Figure 5**, peak values summarized in **Table 2**). In contrast, peak vector strengths occurred at a much wider range of tempi for the other two SFIE models and for the AN fiber activity. Since human perception of musical beats is invariant to the envelope of the stimulus (Henry et al., 2017), these results strongly suggest that neurons with long excitatory and inhibitory synaptic time constants are important for musical beat perception and responsible for biasing the preferred tempo around 100 BPM. Such neurons would produce strong onset firing and reduced sustained firing necessary for creating salient beats. We also found empirically that vector strengths were larger for musical recordings using SFIE A than the other two models (Supplementary Figure 2). For these reasons, SFIE A was used when simulating sub-cortical neural activity in the following experiments.

Dependence of the Synchronization Tempo on Stimulus Audio Frequency

There is some evidence that human perception of musical beats may be biased to particular frequency ranges, but the strength of this effect and the underlying mechanism are unclear. We hypothesized that subcortical processing may produce a frequency bias for tempo induction. Specifically, when multiple carrier frequencies are present with temporal modulations at

TABLE 2 | Preferred tempi (upper) and peak vector strengths (lower) are shown for each stimulus and SFIE model, including the summed AN fiber output without the SFIE model applied (see **Figure 5**).

	Preferred tempo (BPM) Peak vector strength						
	Clicks	SAM noise	SW noise	Tone pips			
				250 Hz	1 kHz	4 kHz	Average ± st dev
SFIE A	111	146	142	112	86	106	117 ± 23
	0.68	0.50	0.61	0.60	0.64	0.57	0.60 ± 0.06
SFIE B	94	223	30	87	76	98	101 ± 64
	0.77	0.16	0.52	0.68	0.72	0.65	0.58 ± 0.23
SFIE C	51	299	30	56	30	52	86 ± 105
	0.94	0.07	0.58	0.87	0.91	0.88	0.71 ± 0.34
AN fibers	267	117	120	204	201	206	186 ± 58
	0.09	0.13	0.31	0.12	0.17	0.13	0.16 ± 0.08

Maxima were computed using quadratic interpolation.

distinct tempi, we expected the synchronization tempo to equal the tempo of the lowest carrier frequency.

1000 stimuli were generated, consisting of two tone pips with carrier frequencies, tempi, and phases that were selected randomly (see **Figure 6A** for example). For each stimulus, the synchronization tempo was normalized relative to the tempos of the two tone pips to get the NST (**Figure 6B**). An NST of zero means that the synchronization tempo was closer to the tempo of the tone pip with the low-frequency carrier, and an NST of one means that it was closer to the tempo for the high-frequency carrier. 80.7% of the stimuli produced NSTs that were within ±0.08 of zero or one (**Figure 7A**). There were significantly more stimuli that produced NSTs near zero than near one (Chi-squared test: $\chi^2 = 149$, $p < 0.001$). On average, synchronization tempi were biased to lower audio frequencies.

The distribution of NSTs, however, also varied with the carrier frequencies (**Figure 7B**) as well as the tempi of the tone pips (**Figure 7C**). Each showed a monotonic relationship with the proportion of NSTs equal to zero. To quantify these dependences and assess their significance, we fit a logistic generalized linear model to the individual NSTs with the low-frequency carrier (f_L), high-frequency carrier (f_H), and the tempi of those tone pips (T_L and T_H respectively) as dependent variables (see Materials and Methods). We found that the generalized linear model fit significantly better than a constant model (Likelihood ratio test: $\chi^2 = 530$, $p < 0.001$), meaning that the carrier frequencies and tempi had a significant effect on the NST relative to the average bias observed initially (**Figure 7A**). Specifically, the NST was significantly dependent on f_H ($\beta_{fH} = 1.39$, $\chi^2 = 78$, $p < 0.001$) and both tempi (T_L: $\beta_{TL} = 0.043$, $\chi^2 = 253$, $p < 0.001$; T_H: $\beta_{TH} = -0.034$, $\chi^2 = 193$, $p < 0.001$). The effect of f_L was not significant ($\beta_{fL} = -0.033$, $\chi^2 = 0.08$, $p = 0.77$).

Overall, synchronization tempi were biased to the tempo for the tone pips with the lower carrier frequency, but the biasing was weakest when the interfering modulations from the higher carrier frequency was close to the lower carrier frequency. Both low-CF and high-CF responses resulted in similar vector strengths for broadband stimuli with tone-pip-like modulations, suggesting that the biasing observed here was due to the spread of excitation in the basilar membrane and not due to differences in the response properties of different CFs (Supplementary Figure 3). However, the tempi of the tone pips had a stronger influence on the synchronization tempo than the carrier frequency, and the synchronization tempo was more likely to equal the fastest tempo. This was contrary to our earlier finding that the vector strength was maximized around 100 BPM for salient, isochronous stimuli. When multiple competing modulations are present in complex stimuli, the faster modulations dominate in the summed synchronized activity, primarily because faster modulations produce more events and are more likely to mask slower modulations (Supplementary Figure 4).

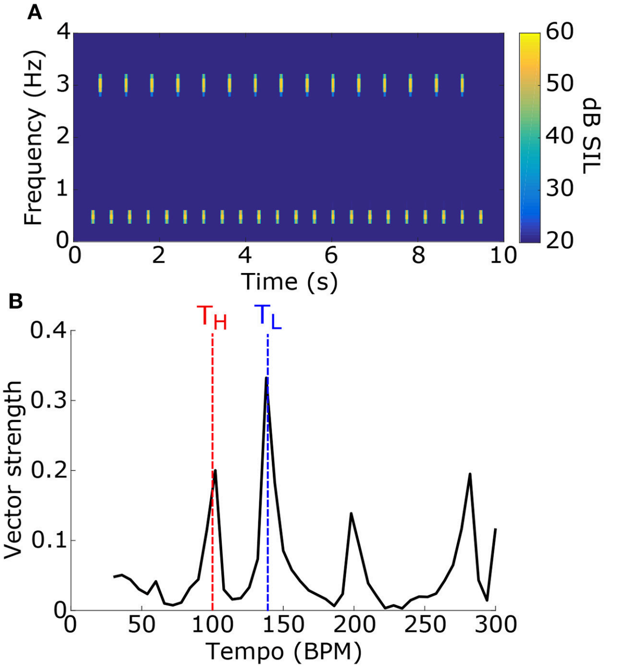


FIGURE 6 | (A) To test for a frequency bias in tempo induction, stimuli consisted of two sets of tone pips at two different carrier frequencies and different tempi. An example stimulus power spectrogram is shown (tone 1: $f_L = 500$ Hz, $T_L = 140$ BPM; tone 2: $f_H = 3$ kHz, $T_H = 100$ BPM; phase = 0 for both). **(B)** The vector strength as a function of tempo for the stimulus in **(A)** is shown. Dashed lines mark the tempi for the tone pips with the low-frequency carrier (blue) and the high-frequency carrier (red). The synchronization tempo was 138 BPM and the NST was 0.05, indicating that it is close to T_L.

Tempo Induction of Real Music

We lastly evaluated tempo-induction performance using two datasets widely used for testing tempo-induction algorithms (Gouyon et al., 2006): a "Ballroom" dataset of 685 ballroom dance music clips, and a "Songs" dataset of 465 songs from a wide variety of genres. For each stimulus the synchronization tempo was computed and compared to the ground-truth tempo for the

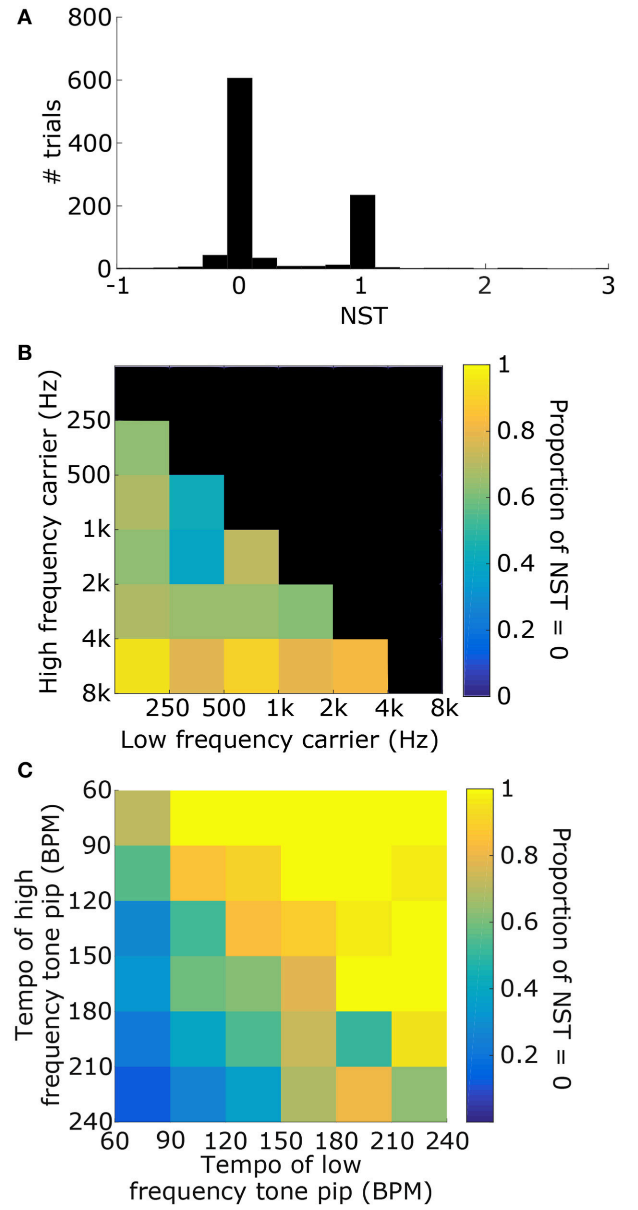


FIGURE 7 | **(A)** Distribution of the NSTs for all 1000 randomly generated stimuli consisting of two tone pips. An NST of 0 means that the synchronization tempo is equal to T_L. An NST of 1 means that the synchronization tempo is equal to T_H. On average, the synchronization tempi were closer to T_L. **(B)** Proportion of trials with NST = 0 with respect to the carrier frequencies of the stimulus. Each bin shows the marginal probability given f_L and f_H. **(C)** Proportion of trials with NST = 0 with respect to the tempi T_L and T_H, plotted similarly to **(B)**.

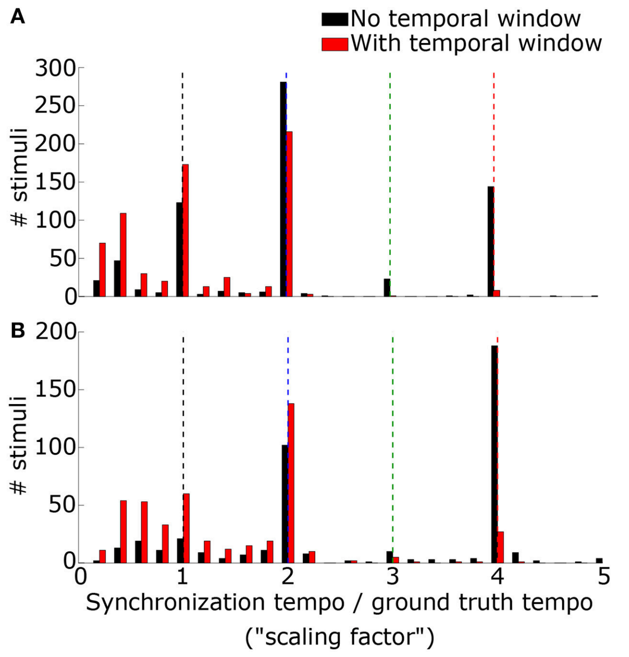


FIGURE 8 | The histogram of the ratio between the synchronization tempo and the ground truth tempo is plotted for the Ballroom dataset **(A)** and the Songs dataset **(B)** without the temporal Gaussian window applied (black) and with the temporal Gaussian window (red). Colored dashed lines mark the scaling factors of 1x (black), 2x (blue), 3x (green), and 4x (red).

recording. The synchronization tempo was equal to the ground-truth tempo for only 19.9% of the stimuli (25.0% for ballroom, 12.4% for songs) (**Figure 8**). More often, the synchronization tempo was twice the ground-truth tempo (31.7% for ballroom, 28.8% for songs, 30.5% overall).

When the PSTH was not smoothed with the temporal smoothing window, fewer synchronization tempi were equal to the ground-truth tempo (18.0% for ballroom, 3.9% for songs, 12.2% overall) (**Figure 8**). However, most of the synchronization tempi occurred at a multiple of the ground-truth tempo: 75.5% of the stimuli produced synchronization tempi at 1-4x the ground truth (81.8% for ballroom, 66.2% for songs) (**Figure 9**). This accounted for 25.1% more of the stimuli than the number that had synchronization tempi at 1-2x the ground truth after smoothing the PSTH.

Thus, while the temporal smoothing window suppresses faster synchronous activity by low-pass filtering the PSTH, it does not unearth a subharmonic peak in vector strength at the true beat frequency of the music. Instead, the model's synchronized activity at a multiple of the ground truth tempo may serve as a reference for determining the actual tempo of the music.

Scaling the Synchronization Tempo

Why is the most synchronous activity at a multiple of the ground truth tempo? One possibility is that the synchronous activity occurs at the "event frequency" of the music, a higher tempo than the beat frequency, such as the frequency of notes played by an instrument or the frequency of drum hits (London, 2012, see also Ding et al., 2017 for a similar result using the modulation spectrum). Indeed, we found that the ratio of the synchronization tempo to the actual tempo, the "scaling factor," depended upon the genre of the ballroom dance music, suggesting that the relationship between the synchronization tempo and the actual tempo may depend upon the rhythm of the music (**Figure 10A**).

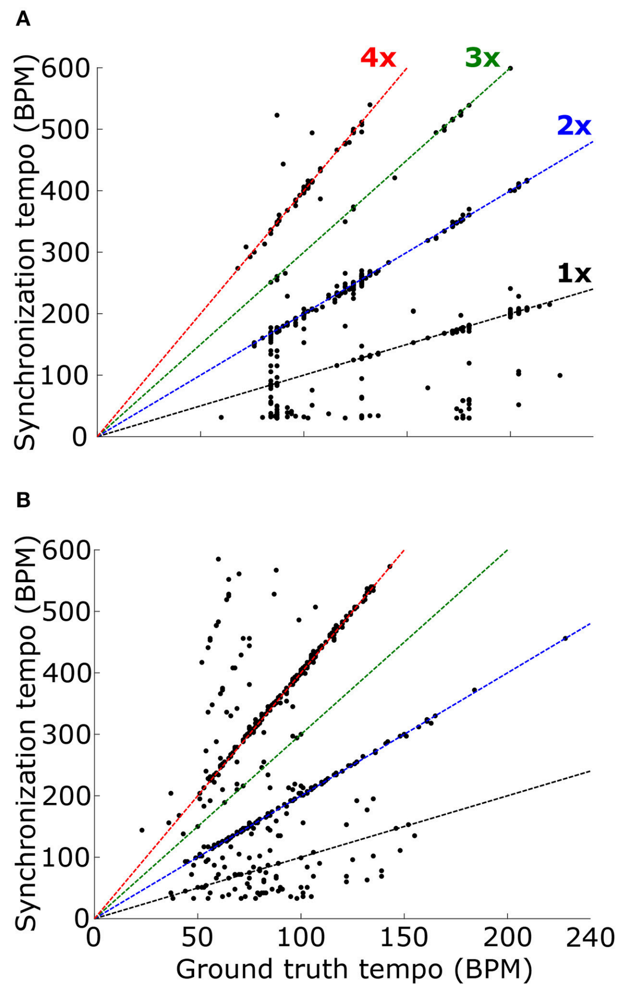


FIGURE 9 | Synchronization tempo using the vector strength of the summed PSTH without the temporal smoothing window is plotted as a function of the ground truth tempo for the Ballroom dataset **(A)** and the Songs dataset **(B)**. Dotted lines mark the slopes corresponding to scaling factors 1–4, as in **Figure 8**. For the combined datasets (1163 stimuli total), 75.5% of the synchronization tempi fell within ±8% of these four slopes.

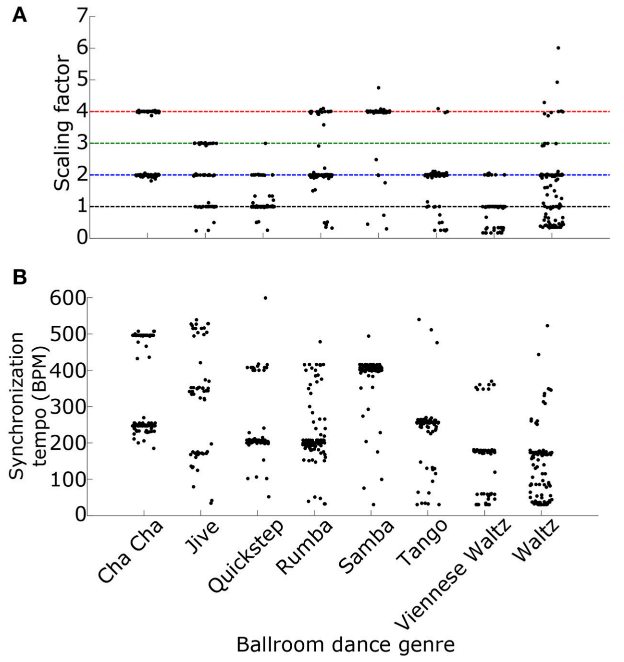


FIGURE 10 | (A) The ratios of the synchronization tempo to the ground truth tempo (the "scaling factors") and **(B)** the synchronization tempi for the 685 Ballroom stimuli are plotted as a function of the ballroom dance genre. Colored dashed lines mark the ratios 1–4, as in **Figures 8**, **9**. Synchronization tempo and the scaling factor both depend upon the genre of the ballroom dance music.

Alternatively, the relationship between the event frequency and the tempo could depend upon the speed of events. As the speed of the events increases, the event frequency would need to be divided by a larger scaling factor in order to get the correct tempo. Because different ballroom dance genres can be qualitatively characterized by different speeds (for example: tango is slower than samba), the event frequencies may also be dependent upon genre. Indeed, we found that the synchronization tempo was dependent upon the genre of the music (**Figure 10B**). Whether the scaling factor is dependent upon the speed or the rhythm of the events, it should be possible to simply divide the synchronization tempo by a scaling factor in order to get the actual beat frequency of the music.

To determine the scaling factor for each stimulus we used mcLDA to design two classifiers (see Materials and Methods). The first classifier used only the synchronization tempo, which captures the speed of the music (**Figure 11A**). The second classifier also contained ISI ratios at fractions of the synchronization tempo to capture information about the rhythm of the music that was present in the synchronized activity (**Figure 11B**). We combined the Ballroom and Songs datasets and randomly selected 75% of the stimuli for training the classifiers and 25% for testing, with 1000 re-samplings of training and testing trials.

Using the synchronization tempo alone, the scaling factor was classified correctly 72.3 ± 2.3% (mean ± standard deviation averaged across all re-samplings) of the time during testing. By dividing the synchronization tempo by the classified scaling factor, tempo-induction performance improved to 55.6 ± 2.5%. The classes were centered on synchronization tempos of 114 ± 2 BPM, 223 ± 2 BPM, 359 ± 13 BPM, and 397 ± 2 BPM for scaling factors 1–4, respectively. As expected, the class for the 1x scaling factor was centered on the 450–600 ms interonset interval range described for other music corpora from a previous study (van Noorden and Moelants, 1999) and the centers for the 2x and 4x scaling factor distributions were roughly twice and four times this range of intervals. The 3x scaling factor was never classified correctly and was often confused with the 2x and 4x classes (**Figure 12A**).

When rhythm information was included, the scaling factor was classified correctly for 76.4 ± 2.2% of the testing stimuli,

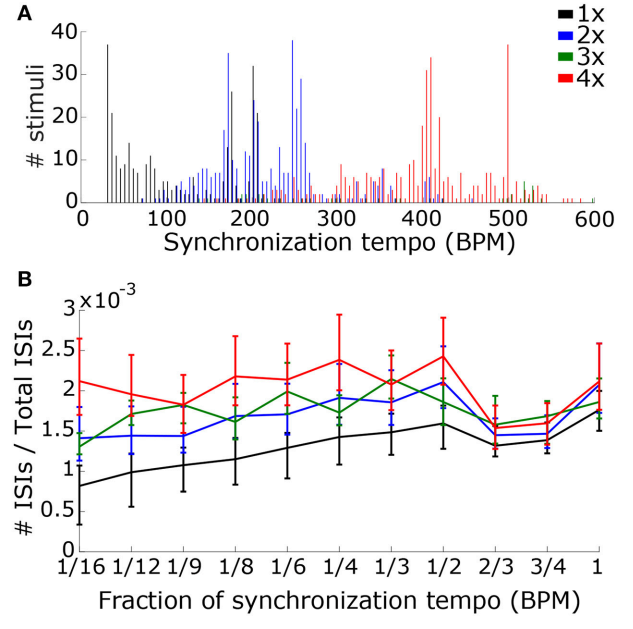


FIGURE 11 | In order to determine the scaling factor, we created two classifiers that used speed and rhythm information in the summed PSTH. **(A)** The first classifier used the speed alone, quantified by the synchronization tempo. A distribution of synchronization tempi for each scaling factor is shown. **(B)** The second classifier used both speed and rhythm. Rhythm was quantified by the ISI ratios (the number of ISIs at a particular interval divided by the total number of ISIs) at intervals corresponding to fractions of the synchronization tempo. The median and interquartile range of the ISI ratios for each fraction is shown for each scaling factor.

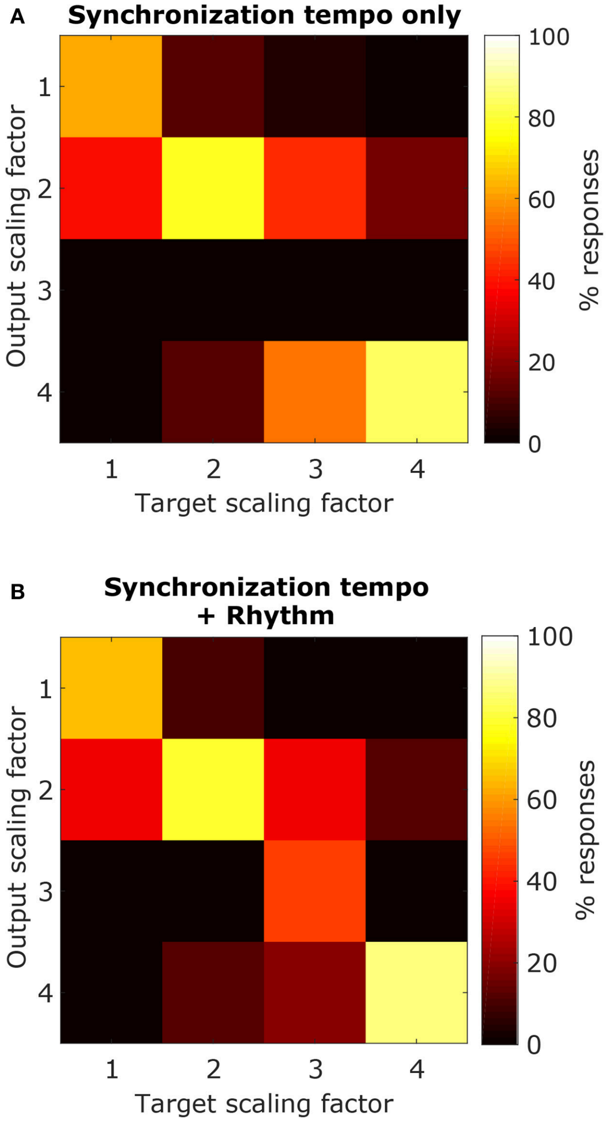


FIGURE 12 | Confusion matrices for classifying each scaling factor with a classifier that just used the synchronization tempo **(A)** or a classifier that included the ISI ratios **(B)**. By including rhythm, there was an improvement in the classification of the 3x scaling factor but little improvement for classifying the other scaling factors.

and tempo-induction performance improved to 60.3 ± 2.6% (61.9 ± 3.3% for ballroom, 58.0 ± 4.1% for songs). The difference in performance between the two classifiers was only moderately significant ($p = 0.016$ for classification, $p = 0.002$ for tempo induction). The primary reason for the improvement in performance was due to an improvement in classification accuracy for the 3x scaling factor (**Figure 12B**). Thus, the perceived beat frequency may depend primarily on the speed of events, with a smaller contribution of rhythm.

DISCUSSION

In this study, we used models of the AN (Zilany et al., 2014), brainstem, and midbrain (Nelson and Carney, 2004; Carney et al., 2015) to simulate neural activity in response to isochronous sound sequences and real music. Our goal was to quantify tempo induction performance based on the simulated sub-cortical neural activity to directly identify the mechanisms necessary to "feel the beat" in music. Furthermore, by using a biomimetic model of acoustic processing in the brainstem and midbrain, we could identify specific additional stages of processing that are necessary to find the beat frequency of music. We found that midbrain-level processing, inherent randomness in neural activity, and a smoothing temporal window together limit the strength of neural synchronization to regular acoustic events and produce a preferred tempo around 100 BPM, in agreement with prior literature. Additionally, cochlear processing generates a low-audio-frequency bias for beat perception, but the tempi of the modulations themselves have a stronger effect on the synchronization tempo than the carrier frequencies. Lastly, despite these successes with simplistic acoustic stimuli, we found that the simulated neural activity often did not synchronize to the beat frequency, but instead synchronized to a multiple of the beat frequency. By using a classifier to appropriately scale the synchronization tempo to the actual beat frequency, tempo-induction performance improved considerably.

We found that midbrain model neurons with strong onset responses produced consistent preferred tempi around 100 BPM for clicks, SAM noise, SW noise, and tone pips. The SFIE model simulates synaptic mechanisms that could give rise to amplitude modulation tuning in the midbrain (Nelson and Carney, 2004). Alternatively, onset responses can also occur from adaptation mechanisms. Rajendran et al. (2017) showed that adaptation mechanisms in the midbrain of gerbils accentuate onsets in complex rhythms that could give rise to beat perception. However, the authors did not look at various envelope shapes. The responses of our model to these rhythmic stimuli for various event durations produced consistent vector strengths at the event frequency and variable vector strengths at all other possible tempi, and often the synchronization was strongest at the event frequency (Supplementary Figure 5), in agreement with our findings for musical recordings. If the events are short enough, we expect that adaptation mechanisms will accentuate the onsets of all events and could produce an equivalent result. Additionally, subjects vary regarding when they choose to tap to these stimuli (Nozaradan et al., 2012; Rajendran et al., 2017), which also suggests that the relationship between subcortical activity and the beat frequency is not one-to-one and may involve a learned mechanism that varies across subjects.

On average, cochlear processing in the AN fiber model appeared to produce a bias to low audio frequencies because the synchronization tempo was more often equal to the tempo for the tone pips with the low-frequency carrier. This bias provides a potential neurobiological reason for why low-frequency instruments carry the beat in some music (for example see Snyder and Krumhansl, 2001). However, it is tricky to test this perceptually; multiple instruments often play simultaneously on the beat, and cochlear delays can explain biases for simultaneous events (Wojtczak et al., 2017). Our stimuli used amplitude modulations at distinct tempi and phases to reduce the effects of simultaneous events, and we quantified the bias using the synchronization strength of neural activity rather than timing to specific events. The presence of a bias may be tested perceptually using these stimuli by having subjects either subjectively identify the beat of the stimulus or tap along with the stimulus at the beat frequency that they perceive. A crowdsourcing design may be most appropriate to properly sample the parameter space of these stimuli.

We used a temporal smoothing window to limit the upper range of tempi to 600 BPM based on previous work (Repp, 2005). This limit does not necessarily correspond to a peripheral motor limit because at this event rate musically trained participants are unable to accurately tap to every fourth event in a fast, isochronous sequence of acoustic events (Repp, 2003). For isochronous, simplistic stimuli, the temporal window was critical in producing the preferred tempo around 100 BPM in our model. However, we found that sub-cortical synchronization often occurred at a multiple of the tempo in musical recordings, and ultimately, by including a classification stage, tempo-induction performance was better without the temporal window. Then when is this temporal window applied? The temporal window defines a constant tolerance for detecting irregular events, but subjects can discriminate click rates around 10 Hz with an accuracy of about 3% (Ungan and Yagcioglu, 2014) implying that it cannot correspond to a limit in acoustic processing. Additionally, it is well known that cortical neurons can synchronize to acoustic periodicities at much faster rates (Joris et al., 2004). The window more likely corresponds to predictive tolerance rather than acoustic tolerance. The exact mechanism is unclear, but it could result from motor planning mechanisms that are used for tapping to regular events (Mendoza and Merchant, 2014; Patel and Iversen, 2014; Merchant et al., 2015; Merchant and Yarrow, 2016; Nozaradan et al., 2017). Motor synchronization may also affect the processing of regular acoustic events in the brainstem and midbrain (Nozaradan et al., 2016), and the accuracy of motor synchronization appears to be correlated with the temporal consistency of brainstem-level encoding of the speech syllable /da/ (Tierney and Kraus, 2013). However, in these studies, sub-cortical activity clearly synchronizes to the acoustics at frequencies higher than 10 Hz, so it is unlikely that the temporal window is applied in the brainstem or midbrain. To explain our findings for musical recordings in particular, it is more likely that temporal limitations are applied cortically and only after the beat frequency has been determined.

Our results suggest that the beat frequency cannot be determined based on the sub-cortical neural activity alone, and a second higher-level mechanism is necessary to perceive the beat. The importance of the relationship between the heard event frequency and the perceived beat frequency has been proposed in the past (London, 2012; Ding et al., 2017). It is unclear from our work what this mechanism might be; internal neural oscillators (Large et al., 2015), motor planning mechanisms (Patel and Iversen, 2014; Merchant et al., 2015; Merchant and Yarrow, 2016), or temporal coding of sequences in the hippocampus (Geiser et al., 2014) could produce patterns of neural activity at subharmonics of the synchronization tempo. However, the process of going from the neural synchronization tempo to the actual tempo is likely to involve a dynamic, high-level system. Listeners can change where they perceive the beat for stimuli with identical rhythms (Iversen et al., 2009). One's preference for the location of the beat is based on experience, since beat perception varies with culture (Drake and El Heni, 2003) and infants prefer different beat frequencies for identical stimuli depending upon the frequency of vestibular sensation during training (Phillips-Silver and Trainor, 2005). Lastly, whereas people often agree on a particular beat for a piece of music, people may tap individually to music at different frequencies and phases relative to the expected tempo (McKinney and Moelants, 2006; Patel and Iversen, 2014). Thus, the relationship between the event frequency and the beat frequency is likely learned through experience and is not due to an innate mechanism.

The techniques used in our modeling work are similar to those used in other algorithms for tempo induction, but our model is unique in predicting the tempo of music using biomimetic models of sub-cortical auditory processing. Several tempo-induction algorithms introduce a template-matching stage that determines the proximity of the computed onset histogram for a single clip of music to a database of onset histogram templates for different rhythms (Seyerlehner et al., 2007; Holzapfel and Stylianou, 2009). Elowsson and Friberg (2015) also included the "speed" of the music, which was determined by a weighted average of the two most probable tempi for the song. In their implementation, both the rhythm information and the speed were used as inputs to a logistic classifier that ultimately determined the tempo (see also Levy, 2011 for the importance of speed judgments in tempo induction algorithms). Our classification scheme is similar. We show that a classifier based on the "speed" alone (the synchronization tempo) does well at identifying the appropriate scaling factor for determining the tempo. We also found that the pattern of interspike intervals, which was used to quantify rhythm, provides a small, albeit significant, amount of information for tempo induction. Also, in our implementation, we assumed that beats are determined based on the summed activity across CF. Similar algorithms detect onsets when the energies in multiple audio frequency bands peak simultaneously (Scheirer, 1998; Klapuri et al., 2006; Ellis, 2007). In contrast, other algorithms have used the frequency content to categorize onset events (Elowsson and Friberg, 2015; Krebs et al., 2016). It is clear that the auditory system combines frequency content into discrete events (Bregman, 1990; Darwin, 1997; Shamma et al., 2011), but where this combination occurs relative to beat processing is unclear. Nevertheless, our model might improve in performance if we introduce a stage that isolates cross-CF neural activity into discrete temporal objects and identifies the tempo based on the pattern of objects rather than on the summed neural activity alone.

Our technique inherently assumes that events equally divide beats and the rhythm that results is based on small integer ratios, which is true for the songs in the datasets we used. However, there are some songs in which the beat of the music is not isochronous, particularly when the music has a complex meter (London, 1995). Our model will identify the regular intervals of events in this case, but a more complex learning mechanism that can identify the explicit timing of non-isochronous beats would be necessary for these particular applications. More strikingly, in Malian jembe drumming, events do not occur at integer ratio subdivisions of the beat (Polak et al., 2016). Music with more complex subdivisions of the beat is particularly problematic for our model because it relies on the initial identification of an event frequency. The issue can be resolved, however, by recognizing that humans have a fairly high tolerance for deviations from synchrony when listening to regular events (Repp and Penel, 2002). The drumming is produced with consistent offsets from the isochronous subdivisions of the beat but they may still be within our perceptual tolerance to asynchrony. A similar effect is observed in classical music; performers slightly vary the timing of notes relative to the strict note durations of the piece for expressive purposes (for review see Patel, 2010). If perceptual processes and motor processes can distinctly subdivide beats, then non-musicians in Mali might subdivide isochronous intervals more evenly than jembe musicians who have experience reproducing the non-isochronous events in the music (see Jacoby and McDermott, 2017).

Our results demonstrate the importance of using real music to study beat perception. Previous studies have primarily used acoustically salient events with complex rhythms. We have shown that the speed of events is relatively more important for tempo induction than the rhythm of those events in musical recordings. We encourage other groups to study the perception of rhythm with biomimetic models of the auditory system. We also encourage others to use real music as stimuli, since musical recordings provide more realistic conditions by which we can better understand how the human brain processes music in general.

AUTHOR CONTRIBUTIONS

NZ designed, performed the study, analyzed the data, and wrote the manuscript. NZ, LC, and EL interpreted the results and approved the final version of the manuscript.

FUNDING

This work was funded by the University of Rochester Biomedical Engineering Department startup funds and by NIH R01 grant 010813.

ACKNOWLEDGMENTS

We would like to thank Jacob Berman for his help interpreting the results for the Ballroom dataset. We would also like to thank Andrew Anderson for his comments on the manuscript. All computations were performed on the BlueHive computation cluster at the University of Rochester.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fnins.2018.00349/full#supplementary-material

REFERENCES

Bartlett, E. L., and Wang, X. (2007). Neural representations of temporally modulated signals in the auditory thalamus of awake primates. *J. Neurophysiol.* 97, 1005–1017. doi: 10.1152/jn.00593.2006

Bregman, A. S. (1990). *Auditory Scene Analysis: The Perceptual Organization of Sound*. Cambridge, MA: MIT Press.

Brown, E. N., Barbieri, R., Ventura, V., Kass, R., and Frank, L. (2002). The time-rescaling theorem and its application to neural spike train data analysis. *Neural Comput.* 14, 325–346. doi: 10.1162/08997660252741149

Carney, L. H., Li, T., and McDonough, J. M. (2015). Speech coding in the brain: representation of vowel formants by midbrain neurons tuned to sound fluctuations. *eNeuro* 2:pii: ENEURO.0004-15.2015. doi: 10.1523/ENEURO.0004-15.2015

Darwin, C. J. (1997). Auditory grouping. *Trends Cogn. Sci.* 1, 327–333. doi: 10.1016/S1364-6613(97)01097-8

Ding, N., Patel, A. D., Chen, L., Butler, H., Luo, C., and Poeppel, D. (2017). Temporal modulations in speech and music. *Neurosci. Behav. Rev.* 81, 181–187. doi: 10.1016/j.neubiorev.2017.02.011

Drake, C., and Botte, M. C. (1993). Tempo sensitivity in auditory sequences: evidence for a multiple-look model. *Attent. Percept. Psychophys.* 54, 277–286. doi: 10.3758/BF03205262

Drake, C., and El Heni, J. B. (2003). Synchronizing with music: intercultural differences. *Ann. N.Y. Acad. Sci.* 999, 429–437. doi: 10.1196/annals.1284.053

Ellis, D. P. W. (2007). Beat tracking by dynamic programming. *J. New Music Res.* 36, 51–60. doi: 10.1080/09298210701653344%0A

Elowsson, A., and Friberg, A. (2015). Modeling the perception of tempo. *J. Acoust. Soc. Am.* 137, 3163–3177. doi: 10.1121/1.4919306

Fraisse, P. (1963). *The Psychology of Time.* Oxford, England: Harper & Row.

Fujioka, T., Ross, B., and Trainor, L. J. (2015). Beta-band oscillations represent auditory beat and its metrical hierarchy in perception and imagery. *J. Neurosci.* 35, 15187–15198. doi: 10.1523/JNEUROSCI.2397-15.2015

Fujioka, T., Trainor, L. J., Large, E. W., and Ross, B. (2012). Internalized timing of isochronous sounds is represented in neuromagnetic beta oscillations. *J. Neurosci.* 32, 1791–1802. doi: 10.1523/JNEUROSCI.4107-11.2012

Geiser, E., Walker, K. M., and Bendor, D. (2014). Global timing: a conceptual framework to investigate the neural basis of rhythm perception in humans and non-human species. *Front. Psychol.* 5:159. doi: 10.3389/fpsyg.2014.00159

Goldberg, J. M., and Brown, P. B. (1969). Response of binaural neurons of dog superior olivary complex to dichotic tonal stimuli: some physiological mechanisms of sound localization. *J. Neurophysiol.* 32, 613–636.

Gouyon, F., Klapuri, A., Dixon, S., Alonso, M., Tzanetakis, G., Uhle, C., et al. (2006). An experimental comparison of audio tempo induction algorithms. *IEEE Trans. Audio Speech Lang. Process.* 14, 1832–1844. doi: 10.1109/TSA.2005.858509

Henry, M. J., Herrman, B., and Grahn, J. A. (2017). What can we learn about beat perception by comparing brain signals and stimulus envelopes? *PLoS ONE*12:e0172454. doi: 10.1371/journal.pone.0172454

Holzapfel, A., and Stylianou, Y. (2009). "A scale transform based method for rhythmic similarity of music," in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing* (Taipei: IEEE), 317–320.

Hove, M. J., Marie, C., Bruce, I. C., and Trainor, L. J. (2014). Superior time perception for lower musical pitch explains why bass-ranged instruments lay down musical rhythms. *Proc. Natl. Acad. Sci. U.S.A.* 111, 10383–10388. doi: 10.1073/pnas.1402039111

Iversen, J. R., Repp, B. H., and Patel, A. D. (2009). Top-down control of rhythm perception modulates early auditory responses. *Ann. N.Y. Acad. Sci.* 1169, 58–73. doi: 10.1111/j.1749-6632.2009.04579.x

Jacoby, N., and McDermott, J. H. (2017). Integer ratio priors on musical rhythm revealed cross-culturally by iterated reproduction. *Curr. Biol.* 27, 359–370. doi: 10.1016/j.cub.2016.12.031

Joris, P. X., Schreiner, C. E., and Rees, A. (2004). Neural processing of amplitude-modulated sounds. *Physiol. Rev.* 84, 541–577. doi: 10.1152/physrev.00029.2003

Kaernbach, C. (1993). Temporal and spectral basis of the features perceived in repeated noise. *J. Acoust. Soc. Am.* 94, 91–97. doi: 10.1121/1.406946

Klapuri, A. P., Eronen, A. J., and Astola, J. T. (2006). Analysis of the meter of acoustic musical signals. *IEEE Trans. Audio Speech Lang. Process.* 14, 342–355. doi: 10.1109/TSA.2005.854090

Krebs, F., Böck, S., Dorfer, M., and Widmer, G. (2016). "Downbeat tracking using beat-synchronous features and recurrent neural networks," in *17th International Society for Music Information Retrieval Conference* (New York, NY), 129–135.

Krishna, B. S., and Semple, M. N. (2000). Auditory temporal processing: responses to sinusoidally amplitude-modulated tones in the inferior colliculus. *J Neurophysiol.* 84, 255–273. doi: 10.1152/jn.2000.84.1.255

Large, E. W. (2000). On synchronizing movements to music. *Hum. Mov. Sci.* 19, 527–566. doi: 10.1016/S0167-9457(00)00026-9

Large, E. W., Herrera, J. A., and Velasco, M. J. (2015). Neural networks for beat perception in musical rhythm. *Front. Syst. Neurosci.* 9:159. doi: 10.3389/fnsys.2015.00159

Lerdahl, F., and Jackendoff, R. (1983). *A Generative Theory of Tonal Music.* Cambridge, MA: MIT Press.

Levy, M. (2011). "Improving perceptual tempo estimation with crowd-sourced annotations," in *International Society for Music Information Retrieval Conference (ISMIR 2011)*, 317–322.

Liberman, M. C. (1978). Auditory-nerve response from cats raised in a low-noise chamber. *J. Acoust. Soc. Am.* 63, 442–455. doi: 10.1121/1.381736

London, J. (1995). Some examples of complex meters and their implications for models of metric perception. *Music Percept. Interdiscip. J.* 13, 59–77. doi: 10.2307/40285685

London, J. (2012). *Hearing in Time: Psychological Aspects of Musical Meter*. Oxford; New York: Oxford University Press.

McKinney, M. F., and Moelants, D. (2006). Ambiguity in tempo perception: what draws listeners to different metrical levels? *Music Percept. Interdiscip. J.* 24, 155–166. doi: 10.1525/mp.2006.24.2.155

McKinney, M. F., Moelants, D., Davies, M. E. P., and Klapuri, A. (2007). Evaluation of audio beat tracking and music tempo extraction algorithms. *J. New Music Res.* 36, 1–16. doi: 10.1080/09298210701653252

Mendoza, G., and Merchant, H. (2014). Motor system evolution and the emergence of high cognitive functions. *Prog. Neurobiol.* 122, 73–93. doi: 10.1016/J.PNEUROBIO.2014.09.001

Merchant, H., Grahn, J., Trainor, L., Rohrmeier, M., and Fitch, W. T. (2015). Finding the beat: a neural perspective across humans and non-human primates. *Philos. Trans. R. Soc. Lond. B. Biol. Sci.* 370:20140093. doi: 10.1098/rstb.2014.0093

Merchant, H., and Yarrow, K. (2016). How the motor system both encodes and influences our sense of time. *Curr. Opin. Behav. Sci.* 8, 22–27. doi: 10.1016/J.COBEHA.2016.01.006

Nelson, P. C., and Carney, L. H. (2004). A phenomenological model of peripheral and central neural responses to amplitude-modulated tones. *J. Acoust. Soc. Am.* 116, 2173–2186. doi: 10.1121/1.1784442

Nelson, P. C., and Carney, L. H. (2007). Neural rate and timing cues for detection and discrimination of amplitude-modulated tones in the awake rabbit inferior colliculus. *J. Neurophysiol.* 97, 522–539. doi: 10.1152/jn.00776.2006

Nozaradan, S., Peretz, I., and Mouraux, A. (2012). Selective neuronal entrainment to the beat and meter embedded in a musical rhythm. *J. Neurosci.* 32, 17572–17581. doi: 10.1523/JNEUROSCI.3203-12.2012

Nozaradan, S., Peretz, I., Missal, M., and Mouraux, A. (2011). Tagging the neuronal entrainment to beat and meter. *J. Neurosci.* 31, 10234–10240. doi: 10.1523/JNEUROSCI.0411-11.2011

Nozaradan, S., Schönwiesner, M., Caron-Desrochers, L., and Lehmann, A. (2016). Enhanced brainstem and cortical encoding of sound during synchronized movement. *Neuroimage* 142, 231–240. doi: 10.1016/J.NEUROIMAGE.2016.07.015

Nozaradan, S., Schwartze, M., Obermeier, C., and Kotz, S. A. (2017). Specific contributions of basal ganglia and cerebellum to the neural tracking of rhythm. *Cortex* 95, 156–168. doi: 10.1016/j.cortex.2017.08.015

Parncutt, R. (1994). A perceptual model of pulse salience and metrical accent in musical rhythms. *Music Percept.* 11, 409–464. doi: 10.2307/40285633

Patel, A. D. (2010). *Music, Language, and the Brain*. New York; NY: Oxford University Press.

Patel, A. D., and Iversen, J. R. (2014). The evolutionary neuroscience of musical beat perception: the action simulation for auditory prediction (ASAP) hypothesis. *Front. Syst. Neurosci.* 8:57. doi: 10.3389/fnsys.2014.00057

Phillips-Silver, J., and Trainor, L. J. (2005). Feeling the beat: movement influences infant rhythm perception. *Science* 308, 1430–1430. doi: 10.1126/science.1110922

Polak, R., Jacoby, N., and London, J. (2016). Both isochronous and non-isochronous metrical subdivision afford precise and stable ensemble entrainment: a corpus study of Malian jembe drumming. *Front. Neurosci.* 10:285. doi: 10.3389/fnins.2016.00285

Povel, D.-J., and Essens, P. (1985). Perception of temporal patterns. *Music Percept.* 2, 411–440. doi: 10.2307/40285311

Rajendran, V. G., Harper, N. S., Garcia-Lazaro, J. A., Lesica, N. A., and Schnupp, J. W. H. (2017). Midbrain adaptation may set the stage for the perception of musical beat. *Proceedings. Biol. Sci.* 284:20171455. doi: 10.1098/rspb.2017.1455

Repp, B. H. (2003). Rate limits in sensorimotor synchronization with auditory and visual sequences: the synchronization threshold and the benefits and costs of interval subdivision. *J. Mot. Behav.* 35, 355–370. doi: 10.1080/00222890309603156

Repp, B. H. (2005). Sensorimotor synchronization: a review of the tapping literature. *Psychon. Bull. Rev.* 12, 969–992. doi: 10.3758/BF03206433

Repp, B. H., and Penel, A. (2002). Auditory dominance in temporal processing: new evidence from synchronization with simultaneous visual and auditory sequences. *J. Exp. Psychol. Hum. Percept. Perform.* 28, 1085–1099. doi: 10.1037//0096-1523.28.5.108

Rosenthal, D. (1992). Emulation of human rhythm perception. *Comp. Music J.* 16, 64–76. doi: 10.2307/3680495

Rouiller, E., De Ribaupierre, Y., Toros-Morel, A., and De Ribaupierre, F. (1981). Neural coding of repetitive clicks in the medial geniculate body of cat. *Hear. Res.* 5, 81–100. doi: 10.1016/0378-5955(81)90028-9

Savage, P. E., Brown, S., Sakai, E., and Currie, T. E. (2015). Statistical universals reveal the structures and functions of human music. *Proc. Natl. Acad. Sci. U.S.A.* 112, 8987–8992. doi: 10.1073/pnas.1414495112

Scheirer, E. D. (1998). Tempo and beat analysis of acoustic musical signals. 103, 588–601. doi: 10.1121/1.421129

Semjen, A., Vorberg, D., and Schulze, H.-H. (1998). Getting synchronized with the metronome: comparisons between phase and period correction. *Psychol. Res.* 61, 44–55.

Seyerlehner, K., Widmer, G., and Schnitzer, D. (2007). "From rhythm patterns to perceived tempo," in *Austrian Computer Society (OCG)*, 519–524.

Shamma, S. A., Elhilali, M., and Micheyl, C. (2011). Temporal coherence and attention in auditory scene analysis. *Trends Neurosci.* 34, 114–123. doi: 10.1016/j.tins.2010.11.002

Shera, C. A., Guinan, J. J., and Oxenham, A. J. (2002). Revised estimates of human cochlear tuning from otoacoustic and behavioral measurements. *Proc. Natl. Acad. Sci. U.S.A.* 99, 3318–3323. doi: 10.1073/pnas.032675099

Snyder, J., and Krumhansl, C. L. (2001). Tapping to ragtime: cues to pulse finding. *Music Percept. Interdiscip. J.* 18, 455–489. doi: 10.1525/mp.2001.18.4.455

Snyder, J. S., and Large, E. W. (2005). Gamma-band activity reflects the metric structure of rhythmic tone sequences. *Cognitive Brain Res.* 24, 117–126. doi: 10.1016/j.cogbrainres.2004.12.014

Stevens, L. T. (1886). On the time-sense. *Mind* 11, 393–404. doi: 10.1093/mind/os-XI.43.393

Stupacher, J., Hove, M. J., and Janata, P. (2016). Audio features underlying perceived groove and sensorimotor synchronization in music. *Music Percept.* 33, 571–589. doi: 10.1525/mp.2016.33.5.571

Sturm, B. L. (2014). *Faults in the Ballroom dataset. Purs. Null Sp.* Available online at: http://media.aau.dk/null_space_pursuits/2014/01/ballroom-dataset.html

Tal, I., Large, E. W., Rabinovitch, E., Wei, Y., Schroeder, C. E., Poeppel, D., et al. (2017). Neural entrainment to the beat: the "missing-pulse" phenomenon. *J. Neurosci.* 37, 6331–6341. doi: 10.1523/JNEUROSCI.2500-16.2017

Tierney, A., and Kraus, N. (2013). The ability to move to a beat is linked to the consistency of neural responses to sound. *J. Neurosci.* 33, 14981–14988. doi: 10.1523/JNEUROSCI.0612-13.2013

Tierney, A., and Kraus, N. (2015). Neural entrainment to the rhythmic structure of music. *J. Cogn. Neurosci.* 27, 400–408. doi: 10.1162/jocn_a_00704

Ungan, P., and Yagcioglu, S. (2014). Significant variations in Weber fraction for changes in inter-onset interval of a click train over the range of intervals between 5 and 300 ms. *Front. Psychol.* 5:1453. doi: 10.3389/fpsyg.2014.01453

van Noorden, L., and Moelants, D. (1999). Resonance in the perception of musical pulse. *J. New Music Res.* 28, 43–66. doi: 10.1076/jnmr.28.1.43.3122

Wojtczak, M., Mehta, A. H., and Oxenham, A. J. (2017). Rhythm judgments reveal a frequency asymmetry in the perception and neural coding of sound synchrony. *Proc. Natl. Acad. Sci. U.S.A.* 114, 1201–1206. doi: 10.1073/pnas.1615669114

Woodrow, H. (1934). The temporal indifference interval determined by the method of mean error. *J. Exp. Psychol.* 17, 167–188. doi: 10.1037/h0070235

Zhang, X., Heinz, M. G., Bruce, I. C., and Carney, L. H. (2001). A phenomenological model for the responses of auditory-nerve fibers: I. nonlinear tuning with compression and suppression. *J. Acoust. Soc. Am.* 109, 648–670. doi: 10.1121/1.1336503

Zilany, M. S., Bruce, I. C., and Carney, L. H. (2014). Updated parameters and expanded simulation options for a model of the auditory periphery. *J. Acoust. Soc. Am.* 135, 283–286. doi: 10.1121/1.4837815

Zilany, M. S. A., Bruce, I. C., Nelson, P. C., and Carney, L. H. (2009). A phenomenological model of the synapse between the inner hair cell and auditory nerve: long-term adaptation with power-law dynamics. *J. Acoust. Soc. Am.* 126, 2390–2412. doi: 10.1121/1.3238250

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Neuroscience

ORIGINAL RESEARCH
published: 14 August 2018
doi: 10.3389/fnins.2018.00532

Connecting Deep Neural Networks to Physical, Perceptual, and Electrophysiological Auditory Signals

Nicholas Huang[1]*, Malcolm Slaney*[2] *and Mounya Elhilali*[1]*

[1] *Laboratory for Computational Audio Perception, Department of Electrical and Computer Engineering, Johns Hopkins University, Baltimore, MD, United States,* [2] *Machine Hearing, Google AI, Google (United States), Mountain View, CA, United States*

Deep neural networks have been recently shown to capture intricate information transformation of signals from the sensory profiles to semantic representations that facilitate recognition or discrimination of complex stimuli. In this vein, convolutional neural networks (CNNs) have been used very successfully in image and audio classification. Designed to imitate the hierarchical structure of the nervous system, CNNs reflect activation with increasing degrees of complexity that transform the incoming signal onto object-level representations. In this work, we employ a CNN trained for large-scale audio object classification to gain insights about the contribution of various audio representations that guide sound perception. The analysis contrasts activation of different layers of a CNN with acoustic features extracted directly from the scenes, perceptual salience obtained from behavioral responses of human listeners, as well as neural oscillations recorded by electroencephalography (EEG) in response to the same natural scenes. All three measures are tightly linked quantities believed to guide percepts of salience and object formation when listening to complex scenes. The results paint a picture of the intricate interplay between low-level and object-level representations in guiding auditory salience that is very much dependent on context and sound category.

Keywords: convolutional neural network, auditory salience, natural scenes, audio classification, electroencephalography, deep learning

OPEN ACCESS

Edited by:
Gert Cauwenberghs,
University of California, San Diego, United States

Reviewed by:
Edmund C. Lalor,
University of Rochester, United States
Nai Ding,
Zhejiang University, China

****Correspondence:***
Mounya Elhilali
mounya@jhu.edu

Specialty section:
This article was submitted to Neuromorphic Engineering, a section of the journal Frontiers in Neuroscience

Received: *01 April 2018*
Accepted: *16 July 2018*
Published: *14 August 2018*

Citation:
Huang N, Slaney M and Elhilali M (2018) Connecting Deep Neural Networks to Physical, Perceptual, and Electrophysiological Auditory Signals. Front. Neurosci. 12:532. doi: 10.3389/fnins.2018.00532

INTRODUCTION

Over the past few years, convolutional neural networks (CNNs) have revolutionized machine perception, particularly in the domains of image understanding, speech and audio recognition, and multimedia analytics (Krizhevsky et al., 2012; Karpathy et al., 2014; Cai and Xia, 2015; Simonyan and Zisserman, 2015; He et al., 2016; Hershey et al., 2017; Poria et al., 2017). A CNN is a form of a deep neural network (DNN) where most of the computation are done with trainable kernel that are slid over the entire input. These networks implement hierarchical architectures that mimic the biological structure of the human sensory system. They are organized in a series of processing layers that perform different transformations of the incoming signal, hence "learning" information in a distributed topology. CNNs specifically include convolutional layers which contain units that are connected only to a small region of the previous layer. By constraining the selectivity of units in these layers, nodes in the network have emergent "receptive fields," allowing them to learn from local information in the input and structure processing in a distributed way; much like neurons in the brain have receptive fields with localized connectivity organized in topographic maps that afford powerful scalability and flexibility in computing. This localized processing is often complemented with fully connected layers which integrate transformations learned across earlier layers, hence incorporating information about content and context and completing the mapping from the signal domain (e.g., pixels, acoustic waveforms) to a more semantic representation.

As with all DNNs, CNNs rely on vast amounts of data to train the large number of parameters and complex architecture of these networks. CNNs have been more widely used in a variety of computer vision tasks for which large datasets have been compiled (Goodfellow et al., 2016). In contrast, due to limited data, audio classification has only recently been able to take advantage of the remarkable learning capability of CNNs. Recent interests in audio data curation have made available a large collection of millions of YouTube videos which were used to train CNNs for audio classification with remarkable performance (Hershey et al., 2017; Jansen et al., 2017). These networks offer a powerful platform to gain better insights on the characteristics of natural soundscapes. The current study aims to use this CNN platform to elucidate the characteristics of everyday sound events that influence their acoustic properties, their salience (i.e., how well they "stand-out" for a listener), and the neural oscillation signatures that they elicit. All three measures are very closely tied together and play a crucial role in guiding our perception of sounds.

Given the parallels between the architecture of a CNN and the brain structures from lower or higher cortical areas, the current work uses the CNN as a springboard to examine the granularity of representations of acoustic scenes as reflected in their acoustic profiles, evoked neural oscillations, and crucially their underlying salience; this latter being a more abstract attribute that is largely ill-defined in terms of its neural underpinnings and perceptual correlates. Salience is a characteristic of a sensory stimulus that makes it attract our attention regardless of where our intentions are. It is what allows a phone ringing to distract us while we are intently in the midst of a conversation. As such, it is a critical component of the attentional system that draws our attention toward potentially relevant stimuli.

Studies of salience have mostly flourished in the visual literature, which benefited from a wealth of image and video datasets as well as powerful behavioral, neural, and computational tools to explore characteristics of visual salience. The study of salience in audition has been limited both by lack of data as well as limitations in existing tools that afford exploring auditory salience in a more natural and unconstrained way. A large body of work has explored aspects of auditory salience by employing artificially constructed stimuli, such as tone and noise tokens (Elhilali et al., 2009; Duangudom and Anderson, 2013). When natural sounds are used, they are often only short snippets that are either played alone or pieced together (Kayser et al., 2005; Duangudom and Anderson, 2007; Kaya and Elhilali, 2014; Tordini et al., 2015; Petsas et al., 2016). Such manipulations limit the understanding of effects of salience in a more natural setting, which must take into account contextual cues as well as complexities of listening in everyday environments.

Despite the use of constrained or artificial settings, studies of auditory salience have shed light on the role of the acoustic profile of a sound event in determining its salience. Loudness is a natural predominant feature, but is complemented by other acoustic attributes, most notably sound roughness and changes in pitch (Nostl et al., 2012; Arnal et al., 2015). Still, the relative contribution of these various cues and their linear or non-linear interactions have been reported to be very important (Kaya and Elhilali, 2014; Tordini et al., 2015) or sometimes provide little benefit (Kim et al., 2014) to determining the salience of a sound event depending on the stimulus structure, its context, and the task at hand. Unfortunately, a complete model of auditory salience that can account for these various facets of auditory salience has not yet been developed. Importantly, studies of auditory salience using very busy and unconstrained soundscapes highlight the limitations of explaining behavioral reports of salience using only basic acoustic features (Huang and Elhilali, 2017). By all accounts, auditory salience is likely a multifaceted process that not only encompasses the acoustic characteristics of the event itself, but is shaped by the preceding acoustic context, the semantic profile of the scene as well as built-in expectation both from short-term and long-term memory, much in line with processes that guide visual salience especially in natural scenes (Treue, 2003; Wolfe and Horowitz, 2004; Veale et al., 2017).

Convolutional neural networks offer a powerful platform to shed light on these various aspects of a natural soundscape and hence can provide insight into the various factors at play in auditory salience in everyday soundscapes. In the present work, we leverage access to a recently published database of natural sounds for which behavioral and neural salience measures are available (Huang and Elhilali, 2017, 2018) to ask the question: how well does activity in a large-scale DNN at various points in the network correlate with these measures? Owing to the complexity of these convolutional models, we do not expect an explicit account of exact factors or processes that determine salience. Rather, we examine the contribution of peripheral vs. deeper layers in the network to explore contributions of different factors along the continuum from simple acoustic features to more complex representations, and ultimately to semantic-level embeddings that reflect sound classes. A number of studies have argued for a direct correspondence between the hierarchy in the primate visual system and layers of deep CNNs (Kriegeskorte, 2015; Yamins and DiCarlo, 2016; Kuzovkin et al., 2017). A recent fMRI study has also shown evidence that a hierarchical structure arises in a sound classification CNN, revealing an organization analogous to that of human auditory cortex (Kell et al., 2018). In the same vein, we explore how well activations at different layers in an audio CNN explain acoustic features, behaviorally measured salience, and neural responses corresponding to a set of complex natural scenes. These signals are all related (but not limited) to salience, and as such this comparison reveals the likely contribution of early vs. higher cortical areas in guiding judgments of auditory salience.

This paper is organized as follows. First, the material and methods employed are presented. This next section describes the database used, the acoustic analysis of audio features in the dataset, and the behavioral and neural responses for this same set obtained from human subjects. The architecture of the neural network is also described as the platform that guides the analysis of other metrics. The results present the information gleaned from the CNN about its representation of acoustic, behavioral, and neural correlates of salience. Finally, the discussion section summarizes the insights gained from these results and its impact for future work to better understand auditory salience and its role in our perception of sounds.

MATERIALS AND METHODS

This next section describes the acoustic data, three types of auditory descriptors [acoustic features, a behavioral measure, and electroencephalography (EEG)], as well as three types of analyses employed in this study (CNN, surprisal, and correlation).

Stimuli

The stimuli used in the present study consist of 20 natural scenes taken from the JHU DNSS (Dichotic Natural Salience Soundscapes) Database (Huang and Elhilali, 2017). Scenes are approximately 2 min in length each and sampled at 22,050 Hz. These scenes originate from several sources, including YouTube, FreeSound, and the BBC Sound Effects Library. The scenes encompass a wide variety of settings and sound objects, as well as a range of sound densities. Stimuli are manually divided into two groups for further analysis; a "sparse" group, which includes scenes with relatively few but clearly isolated acoustic events. An example of a sparse scene includes a recording of a bowling alley in which a relatively silent background is punctuated by the sound of a bowling ball first striking the floor and then the pins. The remaining scenes are categorized as "dense" scenes. Examples of these scenes include a maternity ward, a protest on the streets, and a dog park with continuously ongoing sounds and raucous backgrounds. This comparison between sparse and dense scenes is important because salience in dense scenes is particularly difficult to explain using only acoustic features, and thus more complex information such as sound category may provide a benefit.

Acoustic Features

Each of the scenes in the JHU DNSS database is analyzed to extract an array of acoustic features, including loudness, brightness, bandwidth, spectral flatness, spectral irregularity, pitch, harmonicity, modulations in the temporal domain (rate), and modulations in the frequency domain (scale). Details of these feature calculations can be found elsewhere (Huang and Elhilali, 2017). In addition, the current study also includes an explicit measure of roughness as one of the acoustic features of interest. It is defined as the average magnitude of temporal modulations between 30 and 150 Hz, normalized by the root-mean-squared energy of the acoustic signal, following the method proposed by Arnal et al. (2015).

Behavioral Salience

The Huang and Elhilali (2017) study collected a behavioral estimate of salience in each of the scenes in the JHU DNSS dataset. Briefly, subjects listen to two scenes presented simultaneously in a dichotic fashion (one presented to each ear). Subjects are instructed to use a computer mouse to indicate which scene they are focusing on at any given time. Salience is defined as the percentage of subjects that attend to a scene when compared to all other scenes, as a function of time.

Peaks in the derivative of the salience curve for each scene define onsets of *salient events*. These are moments in which a percentage of subjects concurrently begin listening to the associated scene, regardless of the content of the opposing scene playing in their other ear. The strength of an event is defined as a linear combination of the height of the slope at that point in time and the maximum percentage of subjects simultaneously attending to the scene within a 4-s window following the event. The strongest 50% of these events are used in the event-related analysis in the current study. These events are further manually categorized into one of seven sound classes (speech, music, other vocalization, animal, device/vehicle, tapping/striking, and other). The speech, music, other vocalization, vehicle/device, and tapping/striking classes contained the most number of events and are included in the current study for further analysis. By this definition of salience, the scenes contained 47 events in the speech class, 57 events in music, 39 events in other vocalization, 44 events in vehicle/device, and 28 events in tapping. The two remaining classes consisted of too few instances, with only 11 events in the animal category and eight in a miscellaneous category.

Electroencephalography

Cortical activity while listening to the JHU DNSS stimuli is also measured using EEG, following procedures described in the study by Huang and Elhilali (2018). Briefly, EEG recordings are obtained using a Biosemi Active Two 128-electrode array, initially sampled at 2048 Hz. Each of the 20 scenes is presented to each subject one time in a random order, and listeners are asked to ignore these scenes playing in the background. Concurrently, subjects are presented with a sequence of tones and perform an amplitude modulation detection task. The neural data relevant to the modulation task is not relevant to the current study and is not presented here. It is discussed in the study by Huang and Elhilali (2018).

Electroencephalography signals are analyzed using FieldTrip (Oostenveld et al., 2011) and EEGLab (Delorme and Makeig, 2004) analysis tools. Data are demeaned and detrended, and then resampled at 256 Hz. Power line energy is removed using the Cleanline MATLAB plugin (Mullen, 2012). EEG data are then re-referenced using a common average reference, and eyeblink artifacts are removed using independent component analysis (ICA).

Following these preprocessing steps, energy at various frequency bands is isolated using a Fourier transform over sliding windows (length 1 s, step size 100 ms), and then averaged across the frequencies in a specific band. Six such frequency bands are used in the analysis to follow: Delta (1–4 Hz), Theta (4–7 Hz), Alpha (8–15 Hz), Beta (15–30 Hz), Gamma (30–50 Hz), and High Gamma (70–110 Hz). Next, band energy is z-score normalized within each channel. Band activity is analyzed both on a per-electrode basis and also by averaging activity across groups of electrodes. In addition to a grand average across all 128 electrodes, analysis is also performed by averaging activity in frontal electrodes (21 electrodes near Fz) and central electrodes (23 electrodes near Cz) as defined in Shuai and Elhilali (2014).

Deep Neural Network

A neural network is used in the current study to explore its relationship with salience judgments based on acoustic analysis, behavioral measures, and neural EEG responses (**Figure 1**). The network structure like VGG follows network E presented by Simonyan and Zisserman (2015), with modifications made by Hershey et al. (2017) and Jansen et al. (2017). Briefly, the network staggers convolutional and pooling layers. It contains four convolutional layers, each with relatively small 3 × 3 receptive fields. After each convolutional layer, a spatial pooling layer reduces the number of units by taking maximums over non-overlapping 2 × 2 windows. Next, two fully connected layers then reduce the dimensionality further before the final prediction layer. **Table 1** lists the layers of the network along with their respective dimensionalities. Due to dimensionality constraints, only the layers shown in bold are used in this analysis and reported here, without any expected loss of generality about the results.

Our CNN was trained on the audio from a 4923 class video-classification problem that eventually became the YouTube-8M challenge (Abu-El-Haija et al., 2016). This dataset includes 8 million videos totaling around 500,000 h of audio, and is available online (Abu-El-Haija, 2017). As in the study by Hershey et al. (2017), the audio from each video was divided into 960 ms frames, each mapped onto a time–frequency spectrogram (25 ms window, 10 ms step size, 64 mel-spaced frequency bins). This spectrogram served as the input to the neural network. For training purposes, ground truth labels from each video were automatically generated and every frame within that video was assigned the same set of labels. Each video could have any number of labels, with an average of around five per video, and 4923 distinct labels in total. The labels ranged from very general to very specific. The most general category labels (such as arts and entertainment, games, autos/vehicles, and sports) were applied to roughly 10–20% of the training videos. The most specific labels (such as classical ballet, rain gutter, injury, and FIFA Street) applied only to 0.0001–0.001% of the videos. The network was trained to optimize classification performance over the ground truth labels. The network's classification performance nearly matches that of the Inception DNN model, which was found to show the best results in Hershey et al. (2017), in terms of equal error rate and average precision. Details about the evaluation process can be found in Jansen et al. (2017).

Network Surprisal

We defined change in the activation patterns within a layer of the CNN as "*network surprisal*" (this definition is unrelated to other surprisal analyses that employ information theory or principles of thermodynamics to characterize system dynamics, often used in physics, chemistry, and other disciplines). It represents an estimate of variability in the response pattern across all nodes of a given layer in the network and as such quantifies how congruent or surprising activity at a given moment is relative to preceding activity (**Figure 1B**). In this study, it is computed by taking the Euclidean distance between the activity in a layer at a given time bin (labeled "Current" in red in **Figure 1B**) vs. the average activation in that layer across the previous four seconds (labeled "History" in gray in **Figure 1B**). Thus, a constant pattern of activity would result in a low level of surprisal, while a fluctuation in that pattern over multiple seconds would result in a higher level of surprisal. This measure corresponds structurally to the definition of semantic dissimilarity by Broderick et al. (2018), although it utilizes Euclidean distance as a common metric for evaluating dissimilarity in neural network activity (Krizhevsky et al., 2012; Parkhi et al., 2015). This surprisal feature tracks changes in the scene as it evolves over time by incorporating elements of the acoustic history into its calculation.

Correlation Analyses

The audio, EEG, and CNN data have all been reduced to low-dimensional features. The audio is represented by 10 different acoustic measures, while the 128 channel EEG measurements are summarized by the energy in six different frequency bands, and the multi-channel outputs from the six different layers of the CNN are summarized by the surprisal measure. We next examine correlation between these metrics and the neural network activations.

Each layer of the neural network is compared to behavioral salience, basic acoustic features, and energy in EEG frequency bands using normalized cross correlation. All signals are resampled to the same sampling rate of 10 Hz, and the first 2 s of each scene are removed to avoid the effects of the trial onset. Scenes that are longer than 120 s are shortened to that length. All signals are high-pass filtered with a cutoff frequency of 1/30 Hz to remove overall trends, and then low-pass filtered at 1/6 Hz to remove noise at higher frequencies. Both filters are fourth-order Butterworth filters. The low-pass cutoff frequency is chosen empirically to match the slow movements in the salience signal. Despite the low cutoff frequency, no observable ringing artifacts are noted. Adjusting signal duration to examine any filtering artifacts at the onset of the signal yields quantitively similar results as reported in this paper.

After these pre-processing steps, we compute the normalized cross-correlation between network surprisal and the other continuous (acoustic and neural) signals with a maximum delay time of −3 to +3 s. The normalized correlation is defined as a sliding dot-product of these two signals normalized by the product of their standard deviation (Rao Yarlagadda, 2010). The highest correlation coefficient within a ± 3 s window is selected as the correlation between network surprisal and each of the corresponding signals.

The behavioral responses reflect onsets of salient events (peaks in the slope of the salience curve) and are discrete in time. CNN surprisal activity is compared to behavioral salience in windows surrounding salient events, extending from 3 s before to 3 s after each event. These windows are used to compare correlations for subsets of events, such as for a single category of events. Quantitatively similar results are obtained when using the whole salience curve instead of windows surrounding all salient

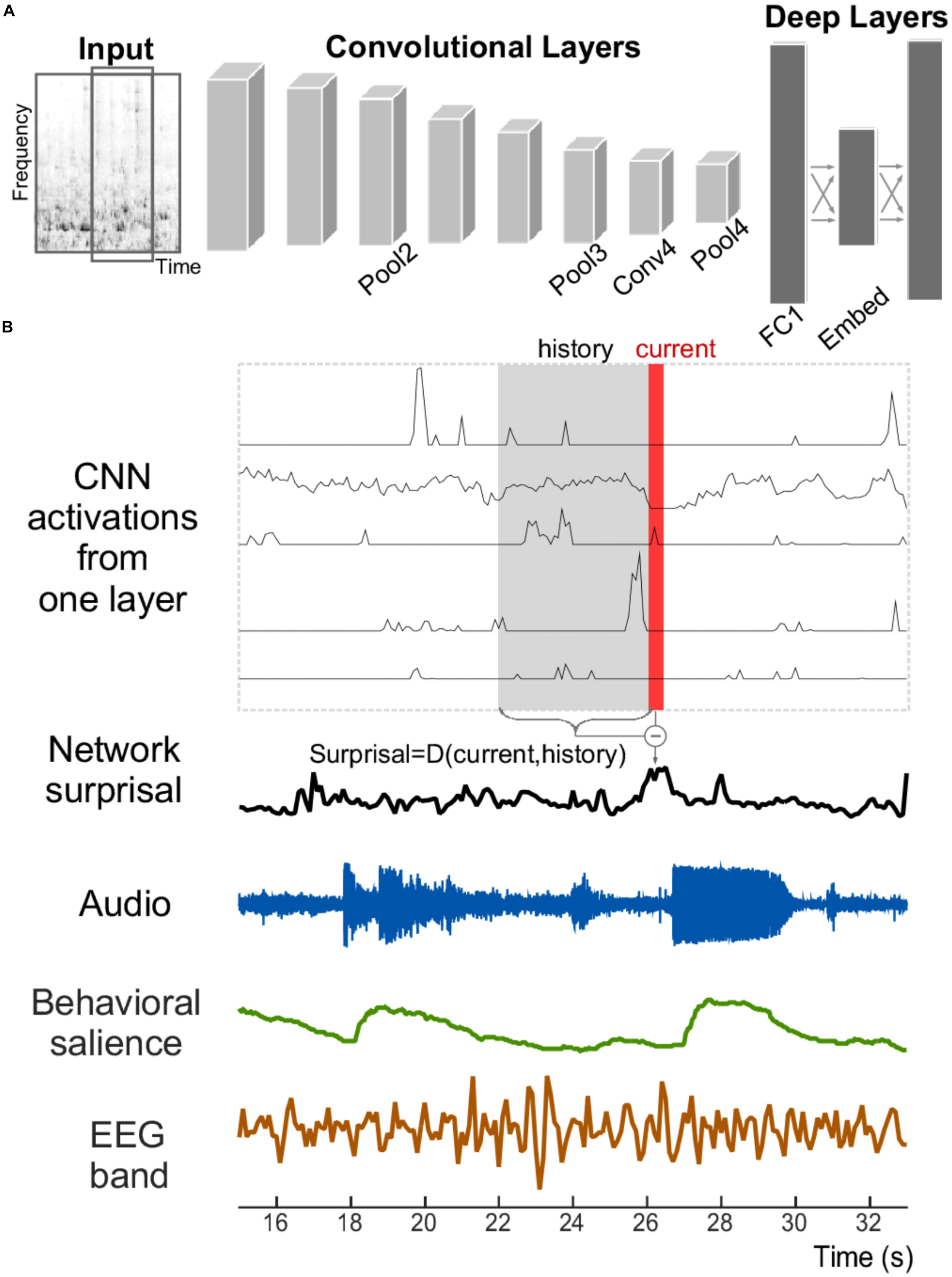


FIGURE 1 | Structure of the convolutional neural network and signals analyzed. **(A)** The convolutional neural network receives the time–frequency spectrogram of an audio signal as input. It is composed of convolutional and pooling layers in an alternating fashion, followed by fully connected layers. **(B)** An example section of an acoustic stimulus (labeled Audio); along with corresponding neural network activity from five example units within one layer of the CNN. A network surprisal measure is then computed as the Euclidian distance between the current activity of the network nodes at that layer (shown in red) against the activity in a previous window (shown in gray with label "History"). Measures of behavioral salience by human listeners (in green) and cortical activity recorded by EEG (in brown) are also analyzed.

TABLE 1 | Dimensions of the input and each layer of the neural network.

Layer type	Abbreviation	Dimensions	Total number of outputs
Input spectrogram		96 × 64	16,384
Convolutional layer	Conv1	96 × 64 × 64	393,216
Pooling layer	Pool1	48 × 32 × 64	98,304
Convolutional layer	Conv2	48 × 32 × 128	196,608
Pooling layer	**Pool2**	**24 × 16 × 128**	**49,152**
Convolutional layer	Conv3	24 × 16 × 256	98,304
Pooling layer	**Pool3**	**12 × 8 × 256**	**24,576**
Convolutional layer	**Conv4**	**12 × 8 × 512**	**49,152**
Pooling layer	**Pool4**	**6 × 4 × 512**	**12,288**
Fully connected layer	**FC1**	**4096**	**4096**
Fully connected layer	**Embed**	**128**	**128**
Output layer/predictions	Predic	4923	4923

Bold text indicates which layers are used in the analysis.

events. The correlation coefficient between behavioral salience and neural surprisal vectors is taken in these windows. For this analysis, the behavioral salience signal is delayed by a fixed time of 1.4 s. A shift is necessary to reflect the delay in motor response required from the behavioral task to report salience. Here, a shift of 1.4 s is empirically determined to correspond to the maximum cross correlation for a majority of the network layers. A fixed delay is used for this case for greater consistency when comparing across different conditions.

To complement the correlation analysis described above, we also examine the cumulative contribution of different CNN layers by assessing the cumulative variance explained by combining activation of consecutive layers. This variance is quantified using a linear regression that uses behavioral salience as the dependent variable and network surprisal from individual layers as independent variables (Weisberg, 2005). Consecutive linear regressions with each layer individually are performed starting with lower layers and continuing to higher layers of the network. After each linear regression, the cumulative variance explained is defined as 1 minus the variance of the residual divided by the variance of the original salience curve (i.e., 1 minus the fraction of variance explained). Then, the residual is used as the independent variable for regression with the next layer. To generate a baseline level of improvement by increasing the number of layers, this linear regression procedure is repeated after replacing all values in layers after the first with numbers generated randomly from a normal distribution (mean 0, variance 1).

Event Prediction

Prediction of salient events is performed by dividing the scene into overlapping time bins (2 s bin size, 0.5 s step size) and then using linear discriminant analysis (LDA; Duda et al., 2000). Each time bin is assigned a label of +1 if a salient event occurred within its respective time frame and a label of 0 otherwise. Network surprisal and the slopes of acoustic features are used to predict salient event using an LDA classifier. The slope of an acoustic feature is calculated by first taking the derivative of the signal, and then smoothing it with three iterations of an equally weighted moving average (Huang and Elhilali, 2017). This smoothing process is selected empirically to balance removal of higher frequency without discarding potential events. As with the previous event-based analysis, these signals are time-aligned by maximizing their correlation with behavioral salience. Each feature is averaged within each time bin, and LDA classification is performed using fivefold cross validation to avoid overfitting (Izenman, 2013). Finally, a threshold is applied to the LDA scores at varying levels to obtain a receiver operating characteristic (ROC) curve (Fawcett, 2006).

RESULTS

This section describes the correlation between the six different layers of the CNN vs. the 10 acoustic features, salience as measured by a behavioral task, and energy in six different frequency bands from the EEG data.

Comparison to Basic Acoustic Features

First, we examine the correspondence between activity in different neural network layers and the acoustic features extracted from each of the scenes. **Figure 2A** shows the correlation coefficient between each acoustic feature and the activity of individual CNN layers. Overall, the correlation pattern reveals stronger values in the four earliest layers (convolutional and pooling) compared the deep layers in the network (fully connected and embedding). This difference is more pronounced in features of a more spectral nature such as spectral irregularity, frequency modulation, harmonicity, and loudness, suggesting that such features may play an important role in informing the network about sound classification during the training of the network. Clearly, not all acoustic features show this strong correlation or any notable correlation. In fact, roughness and rate are basic acoustic measures that show slightly higher correlation in deeper layers relative to earlier layers. **Figure 2B** summarizes the average correlation across all basic acoustic features used in this study as a function of network layer. The trend reveals a clear drop in correlation, indicating that the activity in deeper layers is more removed from the acoustic profile of the scenes. **Figure 2B** inset depicts a statistical analysis of this drop, with slope = -0.026, $t(1198) = -5.8$, p = 7.6×10^{-9}.

Next, we examine the correspondence between activations in the CNN layers and the behavioral judgments of salience as reported by human listeners. **Figure 3A** shows the correlation between behavioral salience and network surprisal across individual layers of the network, taken in windows around salient events (events being local maxima in the derivative of salience, see section "Materials and Methods"). As noted with the basic acoustic features (**Figure 2**), correlation is higher for the earlier layers of the CNN and lower for the later layers. A statistical analysis of the change in correlation across layers reveals a significant slope of -0.041, $t(1360) = -6.8$, $p = 2.1 \times 10^{-11}$ (**Figure 3A**, inset). However, although the correlation for individual deeper network layers is relatively poor, an analysis of their complementary information suggests additional independent contributions of each layer. In fact, the cumulative variance explained as one goes deeper into

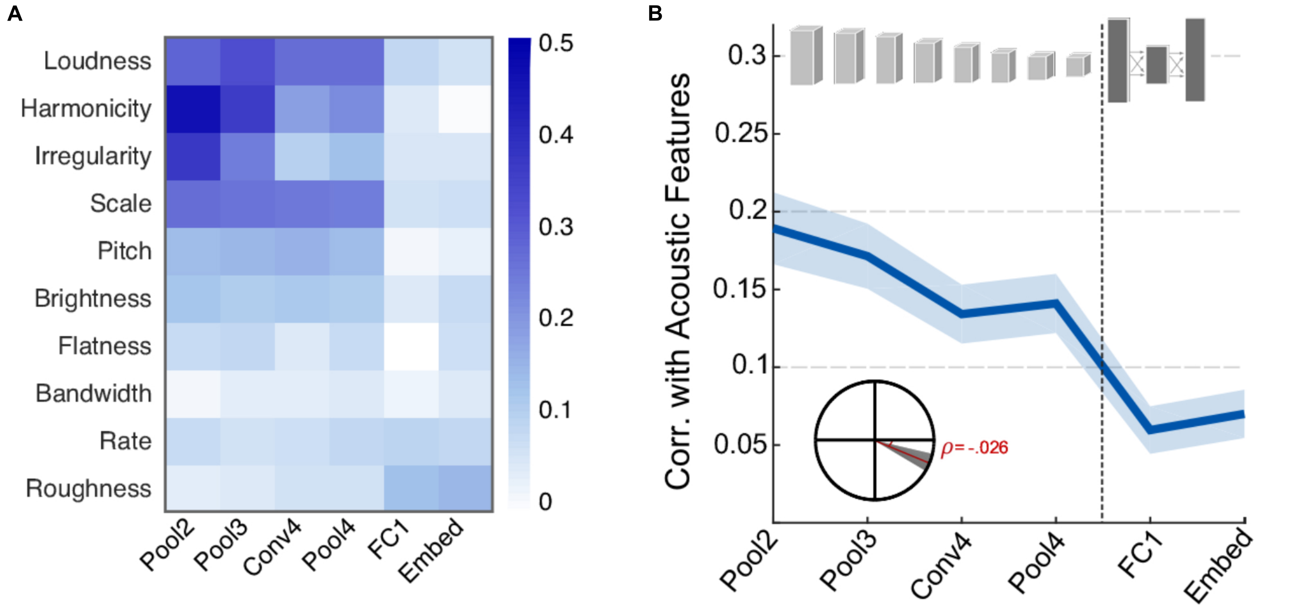


FIGURE 2 | Correlation between neural network activity and acoustic features. **(A)** Correlation coefficients between individual acoustic features and layers of the neural network. Loudness, harmonicity, irregularity, scale, and pitch are the most strongly correlated features overall. **(B)** Average correlation across acoustic features and layers of the neural network. Shaded area depicts ±1 standard error of the mean (SEM). Inset shows the slope of the trend line fitted with a linear regression. The shaded area depicts 99% confidence intervals of the slope.

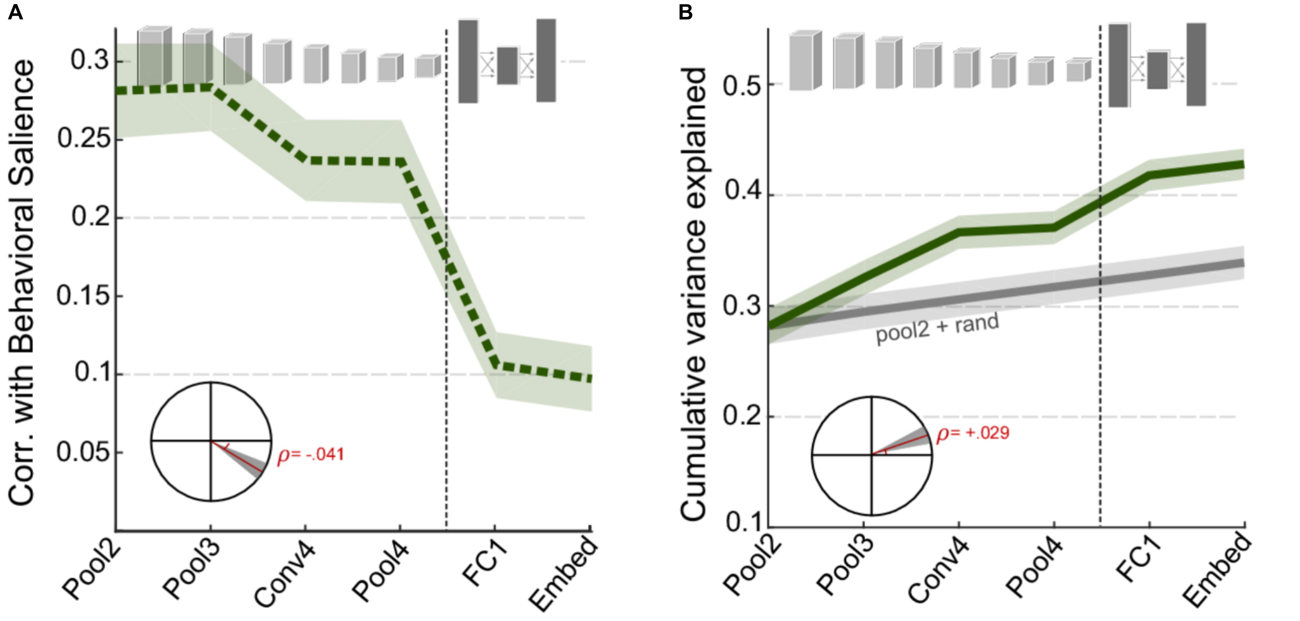


FIGURE 3 | CNN surprisal and behavioral salience. **(A)** Correlation between CNN activity and behavioral salience. **(B)** Cumulative variance explained after including successive layers of the CNN. The gray line shows a baseline level of improvement estimated by using values drawn randomly from a normal distribution for all layers beyond Pool2. For both panels, shaded areas depict ±1 SEM. Insets show the slope of the trend line fitted with a linear regression, with shaded areas depicting 99% confidence intervals of the slope.

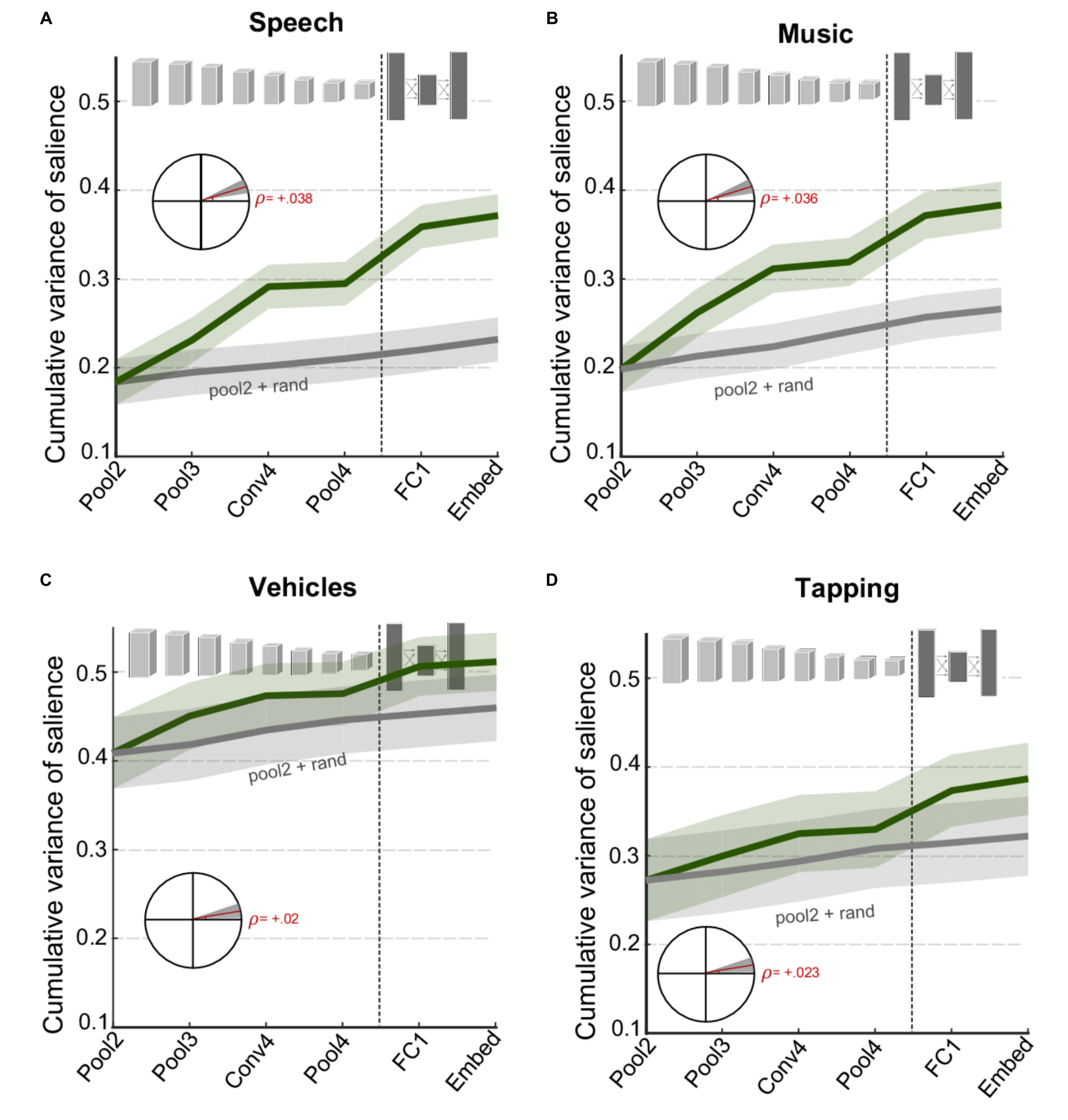


FIGURE 4 | Cumulative variance explained after including successive layers of the CNN for specific categories of events **(A)** speech events, **(B)** music events, **(C)** vehicle events, and **(D)** tapping/striking events. The gray line shows a baseline level of improvement estimated by using values drawn randomly from a normal distribution for all layers beyond Pool2. For all panels, shaded areas depict ±1 SEM. Insets show the slope of the trend line fitted with linear regression, with shaded areas depicting 99% confidence intervals of the slope.

the network shows significantly improved correlation between superficial and deep layers (**Figure 3B**), with a correlation slope of 0.029, $t(1360) = 6.6$, $p = 5 \times 10^{-11}$.

While **Figure 3** looks at complementary information of different network layers in explaining behavioral judgments of salience *on average*, one can look explicitly at specific categories of events and examine changes in information across CNN layers. **Figure 4** contrasts the cumulative variance explained for four classes of events that were identified manually in the database (see section "Materials and Methods"). The figure compares cumulative variance of behavioral salience explained by the network for speech, music, vehicle, and tapping events. The figure shows that speech and music-related events are better explained with the inclusion of deeper later layers [speech: $t(280) = 5.2$, $p = 3.2 \times 10^{-7}$; music: $t(340) = 5.7$, $p = 3.3 \times 10^{-08}$]. In contrast, events from the devices/vehicles and tapping categories are well explained by only the first few peripheral layers of the network, with little benefit provided by deeper layers [device: $t(262) = 1.8$, $p = 0.069$; tapping: $t(166) = 2.2$, $p = 0.028$]. Results for other vocalizations closely match those of the vehicle category (data not shown), $t(196) = 2.2$, $p = 0.033$. Overall, the figure highlights that contribution of different CNN layers to perceived salience of different scenes does vary drastically depending on semantic meaning and show varying degrees of complementarity between the acoustic front-end representation and the semantic deeper representations.

The ability to predict where salient events occur is shown in **Figure 5**. Each scene is separated into overlapping time bins which are labeled based on whether or not an event occurred during that time frame. LDA is then performed using either a combination of acoustics and network surprisal, or the acoustic features alone. The prediction is improved through the inclusion of information from the neural network, with an area under the ROC curve of 0.734 when using only the acoustic features compared to an area of 0.775 after incorporating network surprisal. This increase in performance indicates that changes in network activity make a contribution to the salience prediction that is not fully captured by the acoustic representation.

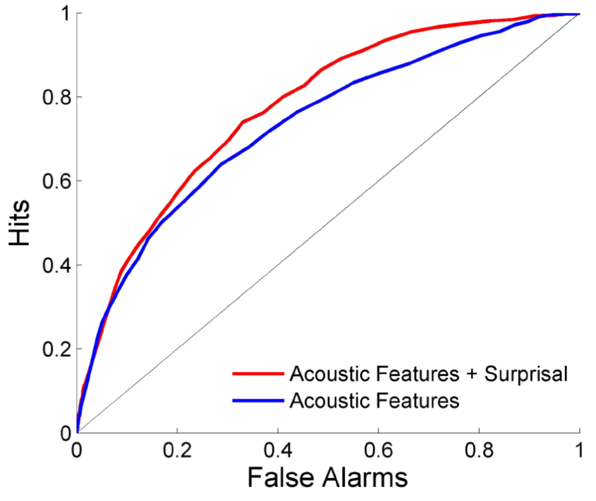


FIGURE 5 | Event prediction performance. Predictions are made using LDA on overlapping time bins across scenes. The area under the ROC curve is 0.775 with a combination of acoustic features and surprisal, while it reaches only 0.734 with acoustic features alone.

One of the key distinctions between the different event categories analyzed in **Figure 4** is not only the characteristics of the events themselves but also the context in which these events are typically present. On the one hand, speech scenes tend to have ongoing activity and dynamic backgrounds against which salient events stand out; while vehicle scenes tend to be rather sparse with few notable events standing out as salient. An analysis contrasting sparse vs. dense scenes in our entire dataset (see section "Materials and Methods") shows a compelling difference between the correlations of acoustic salience for dense scenes and for sparse scenes especially in the convolutional layers (**Figure 6A**). This difference is statistically significant when comparing the mean correlation for early vs. deep layers, $t(4) = -5.4$, $p = 0.0057$. On the other hand, the network's activation in response to acoustic profiles in the scenes do not show any distinction between sparse and dense scenes and across early and deep layers (**Figure 6B**), $t(4) = -0.24$, $p = 0.82$.

Finally, we examine the contrast between neural responses recorded using EEG and CNN activations. As shown in **Figure 7**, energy in many frequency bands of the neural signal shows stronger correlation with activity in higher levels of the CNN rather than lower layers and follows an opposite trend to that of acoustic features. **Figure 7A** shows the correlation between network activity and individual EEG frequency bands and shows a notable increase in correlation for higher frequency bands (Delta, Beta, Gamma, and High Gamma). The Theta and Alpha bands appear to follow a somewhat opposite trend, though their overall correlation values are rather small. **Figure 7B** summarizes the average correlation trend across all frequency bands, with slope = 0.015, $t(718) = 3.6$, $p = 3.2 \times 10^{-4}$). It is worth noting the average correlation between CNN activity and EEG responses is rather small overall (between 0 and 0.1) but still significantly higher than 0, $t(719) = 7.4$, $p = 4.5 \times 10^{-13}$. The increasing trend provides further support to the notion that higher frequency neural oscillations are mostly aligned with increasingly complex feature and semantic representations crucial for object recognition in higher cortical areas, and correspondingly in deeper layers of the CNN (Kuzovkin et al., 2017).

To explore the brain regions that are most closely related to the CNN activity, individual electrode activities are also correlated with surprisal. **Figure 8A** shows a small difference between neural activity in Central and Frontal areas, with the former having relatively higher correlation with early layers and the latter having higher correlation with deep layers. This trend is not statistically significant, however. **Figure 8B** shows the pattern across electrodes of these correlations values for the beta and gamma bands. Activity in the Beta band is most correlated to the convolutional layers of the CNN for central electrodes near C3 and C4, while it is most correlated to the deep layers for frontal electrodes near Fz. In contrast, Gamma band activity shows little correlation with the early layers of the CNN, but more closely matches activation in deep layers for electrodes near Cz.

DISCUSSION

Recent work on deep learning models provides evidence of strong parallels between the increasing complexity of signal representation in these artificial networks and the intricate sensory transformations in sensory biological systems that map

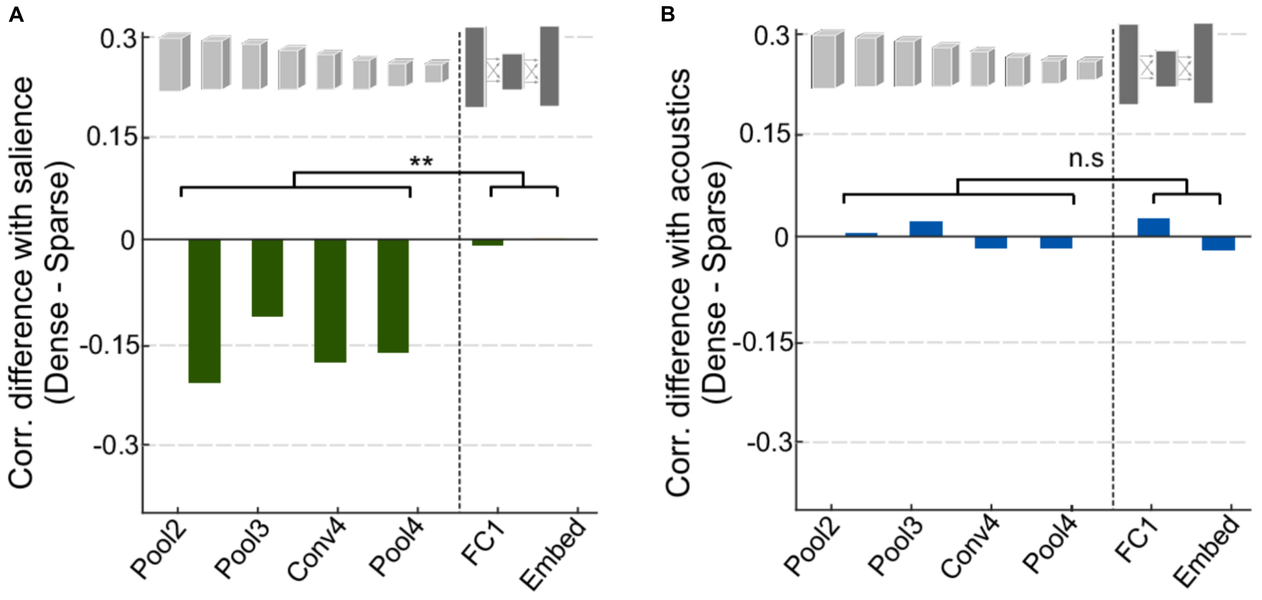


FIGURE 6 | Analysis of dense vs. sparse scenes. **(A)** Difference in correlation between salience and CNN activity for dense and sparse scenes. Negative values indicate that salience in sparse scenes was more highly correlated with CNN activity. **(B)** Difference in correlation between acoustic features and CNN activity for dense and sparse scenes.

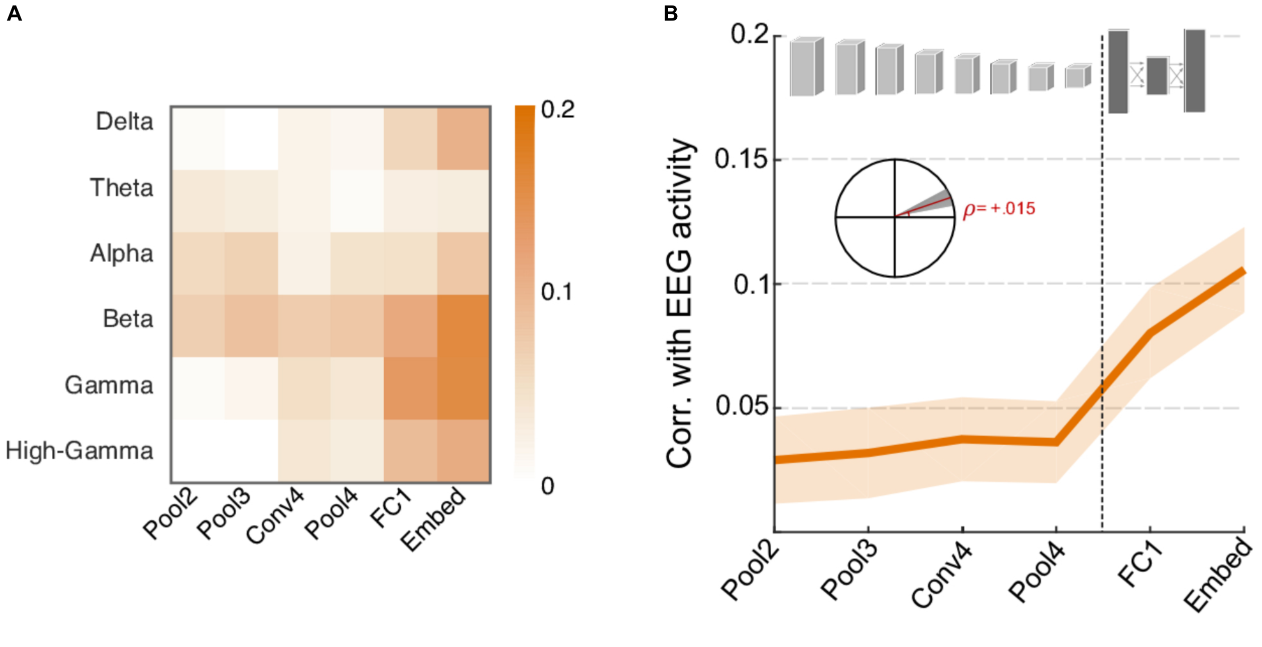


FIGURE 7 | Correlation between neural network activity and energy in EEG frequency bands. **(A)** Correlation coefficients between individual EEG frequency bands and layers of the neural network. Gamma, Beta, and High-Gamma frequency bands are the most strongly correlated bands overall. **(B)** Average correlation across EEG activity and layers of the neural network. Shaded area depicts ±1 SEM. Inset shows the slope of the trend line fitted with linear regression, with a shaded area depicting the 99% confidence interval of the slope.

incoming stimuli onto object-level representations (Yamins et al., 2014; Guclu and van Gerven, 2015; Cichy et al., 2016). The current study leverages the complex hierarchy afforded by CNNs trained on audio classification to explore parallels between network activation and auditory salience in natural sounds measured through a variety of modalities. The analysis examines the complementary contribution of various layers in a CNN architecture and draws a number of key observations from three types of signals: acoustic, behavioral, and neural profiles.

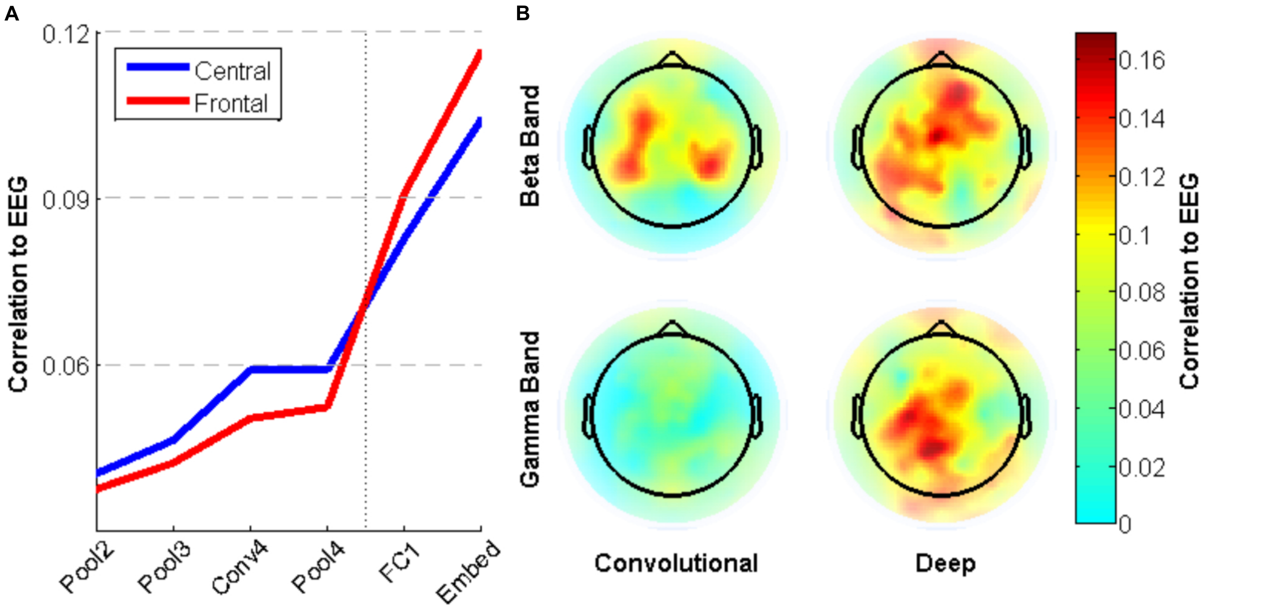


FIGURE 8 | Correlation between neural network activity and energy in EEG frequency bands for specific electrodes. **(A)** Average correlation between electrode activity across frequency bands for electrodes in central (near Cz) and frontal (near Fz) regions. **(B)** Correlation between beta/gamma band activity for individual electrodes and convolutional/deep layers of the neural network.

First, as expected, the earlier layers in the CNN network mostly reflect the acoustic characteristics of a complex soundscape. The association of acoustic features with CNN activation decreases in correlation as the signal propagates deeper into the network. The acoustic features that are most clearly reflected with higher fidelity are mostly spectral, and include harmonicity, frequency modulation, and spectral irregularity, along with loudness which directly modulates overall signal levels. It is important to remember that the CNN network used in the current work is trained for audio classification and employs a rather fine-resolution spectrogram at its input computed with 25 ms bins over frames of about 1 s. As such, it is not surprising to expect a strong correlation between spectral features in the input and early representations of the peripheral layers of the CNN network (Dai et al., 2017; Lee et al., 2017; Wang et al., 2017). Interestingly, two features that are temporal in nature, namely, rate and most prominently roughness, show a somewhat opposite trend with a mildly increased correlation with deeper CNN layers. Both these acoustic measures quantify the degree of amplitude modulations in the signal over longer time scales of tens to hundreds of milliseconds, and we can speculate that such measures would involve longer integration levels that are more emblematic of deeper layers in the network that pool across various localized receptive fields. The distributed activation of CNN layers reflecting various acoustic features supports previous accounts of hierarchical neural structures in auditory cortex that combine low-level and object-level representations extending beyond the direct physical attributes of the scenes (Formisano et al., 2008; Staeren et al., 2009). This distributed network suggests an intricate, multi-region circuitry underlying the computation of sound salience in the auditory system, much in line with reported underpinnings of visual salience circuits in the brain (Veale et al., 2017).

Second, the results show a strong correlation between peripheral layers of the CNN and behavioral reports of salience. This trend is not surprising given the important role acoustic characteristics of the signal play in determining the salience of its events (Kaya and Elhilali, 2014; Kim et al., 2014; Huang and Elhilali, 2017). This view is then complemented by the analysis of cumulative variance explained by gradually incorporating activation of deeper layers in the neural network. **Figure 3** clearly shows that information extracted in later layers of the network supplements activation in earlier layers and offers an improved account of auditory salience. This increase is maintained even at the level of the fully connected layers suggesting a complementary contribution of low-level and category-level cues in guiding auditory salience. This observation is further reinforced by focusing on salience of specific sound categories. In certain cases that are more typical of sparse settings with prominent events such as tapping or vehicle sounds, it appears that the low-level acoustic features are the main determinants of auditory salience with little contribution from semantic-level information. In contrast, events in the midst of a speech utterance or a musical performance appear to have a significant increase in variance explained by incorporating all CNN layers (**Figure 4**). The complementary nature of peripheral and object-level cues is clearly more prominent when taking into account the scene context, by contrasting denser, busy scenes with quieter environments with occasional, prominent events. Dense settings typically do not have as many conspicuous clear changes in acoustic information across time, and as a result, they seem to require more semantic-level information to complement information from acoustic features for a complete account of auditory salience.

Third, the CNN layer activation shows an opposite correlation trend with neural oscillation measured by EEG. In particular, the deeper layers of the neural network have higher correlation with activity in the higher frequency bands (beta, gamma, and high gamma bands). Synchronous activity in the Gamma band has been shown to be associated with object representation (Rodriguez et al., 1999; Bertrand and Tallon-Baudry, 2000), which would be directly related to the audio classification task. Activity in both the Gamma and Beta bands has also been linked to hearing novel stimuli (Haenschel et al., 2000). Moreover, Gamma band activity is known to be strongly modulated by attention (Tiitinen et al., 1993; Müller et al., 2000; Doesburg et al., 2008), which further reinforces the relationship between object category and salience.

In particular, the CNN activation patterns of the deep layers correlate most strongly with neural oscillations in frontal areas of the brain. This finding expands on the recent work by Kell et al. (2018), which found that activation patterns within intermediate layers of their CNN were the best at predicting activity in the auditory cortex. It stands to reason that later layers of the network would correspond more to higher level brain regions, which may play a role in attention and object recognition.

Overall, all three metrics used in the current study offer different accounts of conspicuity of sound events in natural soundscapes. By contrasting these signals against activations in a convolutional DNN trained for audio recognition, we are able to assess the intricate granularity of information that drives auditory salience in everyday soundscapes. The complexity stems from the complementary role of cues along the continuum from low-level acoustic representation to coherent object-level embeddings. Interestingly, the contribution of these different transformations does not uniformly impact auditory salience for all scenes. The results reveal that the context of the scene plays a crucial role in determining the influence of acoustics or semantics or possibly transformations in between. It is worth noting that the measure of surprisal used here is but one way to characterize surprise. Looking at changes in a representation compared to the average of the last few seconds is simple and proves to be effective. However, different ways to capture the context, perhaps including fitting the data to a multimodal Gaussian mixture model, as well as different time scales should be investigated.

Further complicating the interaction with context effects is the fact that certain acoustic features should not be construed as simple transformation of the acoustic waveform or the auditory spectrogram. For instance, a measure such as roughness appears to be less correlated with lower layers of the CNN. This difference suggests that acoustic roughness may not be as readily extracted from the signal as the other acoustic measures by the neural network, but it is nonetheless important for audio classification and correlates strongly with perception of auditory salience (Arnal et al., 2015).

One limitation of the CNN structure is that it only transmits information between layers in the forward direction, while biological neural systems incorporate both feedforward and feedback connections. Feedback connections are particularly important in studies of attention because salience (bottom-up attention) can be modified by top-down attention. This study uses behavioral and physiological data that were collected in such a way that the influence of top-down activity was limited; however, a complete description of auditory attention would need to incorporate such factors. An example of a feedback CNN that seeks to account for top-down attention can be found in Cao et al. (2015).

It is not surprising that our limited understanding of the complex interplay between acoustic profiles and semantic representations has impeded development of efficient models of auditory salience that can explain behavioral judgments, especially in natural, unconstrained soundscapes. So far, most accounts have focused on incorporating relevant acoustic cues that range in complexity from simple spectrographic representation to explicit representation of pitch, timbre, or spectro-temporal modulation (Duangudom and Anderson, 2007; Kalinli and Narayanan, 2007; Tsuchida and Cottrell, 2012; Kaya and Elhilali, 2014). However, as highlighted by the present study, it appears that a complementary role of intricate acoustic analysis (akin to that achieved from the complex architecture of convolutional layers in the current CNN) as well as auditory object representations will be necessary to not only account for contextual information about the scene but may determine the salience of a sound event depending on its category, sometimes regardless of its acoustic attributes.

ETHICS STATEMENT

This study was carried out in accordance with the recommendations of the Belmont Report and the Homewood Institutional Review Board at the Johns Hopkins University. The protocol was approved by the Homewood Institutional Review Board. All subjects gave written informed consent in accordance with the Declaration of Helsinki.

AUTHOR CONTRIBUTIONS

All authors contributed to the conception and design of the study, led by ME. NH collected behavioral and EEG data and conducted statistical analysis. MS performed the neural network computation. All authors contributed to manuscript write up and read and approved the submitted version.

FUNDING

This work was supported by the National Institutes of Health under grant R01HL133043 and the Office of Naval research under grants ONR N000141010278, N000141612045, and N000141210740.

ACKNOWLEDGMENTS

Early explorations of these ideas took place at the Neuromorphic Cognition Engineering Workshop in Telluride, Colorado.

REFERENCES

Abu-El-Haija, S. (2017). *YouTube-8M Dataset*. Available at: https://research.google.com/youtube8m/index.html [Accessed June 27, 2018].

Abu-El-Haija, S., Kothari, N., Lee, J., Natsev, P., Toderici, G., Varadarajan, B., et al. (2016). *YouTube-8M: A Large-Scale Video Classification Benchmark*. Available at: https://arxiv.org/pdf/1609.08675.pdf [Accessed March 20, 2018].

Arnal, L. H., Flinker, A., Kleinschmidt, A., Giraud, A. L., and Poeppel, D. (2015). Human screams occupy a privileged niche in the communication soundscape. *Curr. Biol.* 25, 2051–2056. doi: 10.1016/j.cub.2015.06.043

Bertrand, O., and Tallon-Baudry, C. (2000). Oscillatory gamma activity in humans: a possible role for object representation. *Int. J. Psychophysiol.* 38, 211–223. doi: 10.1016/S0167-8760(00)00166-5

Broderick, M. P., Anderson, A. J., Di Liberto, G. M., Crosse, M. J., and Lalor, E. C. (2018). Electrophysiological correlates of semantic dissimilarity reflect the comprehension of natural, narrative speech. *Curr. Biol.* 28, 803.e3–809.e3. doi: 10.1016/j.cub.2018.01.080

Cai, G., and Xia, B. (2015). "Convolutional neural networks for multimedia sentiment analysis," in *Proceedings of the Natural Language Processing and Chinese Computing. NLPCC,* Beijing, 159–167. doi: 10.1007/978-3-319-25207-0_14

Cao, C., Liu, X., Yang, Y., Yu, Y., Wang, J., and Wang, Z. (2015). "Look and think twice: capturing top-down visual attention with feedback," in *Proceedings of the IEEE International Conference Computer Vision*, Piscataway, NJ, 2956–2964.

Cichy, R. M., Khosla, A., Pantazis, D., Torralba, A., and Oliva, A. (2016). Comparison of deep neural networks to spatio-temporal cortical dynamics of human visual object recognition reveals hierarchical correspondence. *Sci. Rep.* 6:27755. doi: 10.1038/srep27755

Dai, W., Dai, C., Qu, S., Li, J., and Das, S. (2017). "Very deep convolutional neural networks for raw waveforms," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, New Orleans, LA, 421–425. doi: 10.1109/ICASSP.2017.7952190

Delorme, A., and Makeig, S. (2004). EEGLAB: an open sorce toolbox for analysis of single-trail EEG dynamics including independent component anlaysis. *J. Neurosci. Methods* 134, 9–21. doi: 10.1016/j.jneumeth.2003.10.009

Doesburg, S. M., Roggeveen, A. B., Kitajo, K., and Ward, L. M. (2008). Large-scale gamma-band phase synchronization and selective attention. *Cereb. Cortex* 18, 386–396. doi: 10.1093/cercor/bhm073

Duangudom, V., and Anderson, D. V. (2007). "Using Auditory Saliency To Understand Complex Auditory Scenes," in *Proceedings of the 15th European Signal Processing Conference (EUSIPCO 2007)*, Poznan.

Duangudom, V., and Anderson, D. V. (2013). "Identifying salient sounds using dual-task experiments," in *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, Piscataway, NJ, 1–4. doi: 10.1109/WASPAA.2013.6701865

Duda, R. O., Hart, P. E., and Stork, D. G. (2000). *Pattern Classification*. New York, NY: Wiley-Interscience.

Elhilali, M., Xiang, J., Shamma, S. A., and Simon, J. Z. (2009). Interaction between attention and bottom-up saliency mediates the representation of foreground and background in an auditory scene. *PLoS Biol.* 7:e1000129. doi: 10.1371/journal.pbio.1000129

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognit. Lett.* 27, 861–874. doi: 10.1016/j.patrec.2005.10.010

Formisano, E., Martino, F., De Bonte, M., and Goebel, R. (2008). Who is saying what? Brain-based decoding of human voice and speech. *Science* 322, 970–973. doi: 10.1126/science.1164318

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. Cambridge, MA: MIT press.

Guclu, U., and van Gerven, M. A. J. (2015). Deep neural networks reveal a gradient in the complexity of neural representations across the ventral stream. *J. Neurosci.* 35, 10005–10014. doi: 10.1523/JNEUROSCI.5023-14.2015

Haenschel, C., Baldeweg, T., Croft, R. J., Whittington, M., and Gruzelier, J. (2000). Gamma and beta frequency oscillations in response to novel auditory stimuli: a comparison of human electroencephalogram (EEG) data with in vitro models. *Proc. Natl. Acad. Sci. U.S.A.* 97, 7645–7650. doi: 10.1073/pnas.120162397

He, K., Zhang, X., Ren, S., and Sun, J. (2016). "Deep residual learning for image recognition," in *Proceedings of the 2016 IEEE Conference Computer Vision Pattern Recognition*, Las Vegas, NV, 770–778. doi: 10.1109/CVPR.2016.90

Hershey, S., Chaudhuri, S., Ellis, D. P. W., Gemmeke, J. F., Jansen, A., Moore, R. C., et al. (2017). "CNN architectures for large-scale audio classification," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, Piscataway, NJ, 131–135. doi: 10.1109/ICASSP.2017.7952132

Huang, N., and Elhilali, M. (2017). Auditory salience using natural soundscapes. *J. Acoust. Soc. Am.* 141:2163. doi: 10.1121/1.4979055

Huang, N., and Elhilali, M. (2018). Neural underpinnnings of auditory salience in natural soundscapes. *bioRxiv* [Preprint]. doi: 10.1101/376525

Izenman, A. J. (2013). "Linear discriminant analysis," in *Modern Multivariate Statistical Techniques*, eds G. Casella, S. Fienberg, and I. Olkin (Heidalberg: Springer), 237–280. doi: 10.1007/978-0-387-78189-1_8

Jansen, A., Gemmeke, J. F., Ellis, D. P. W., Liu, X., Lawrence, W., and Freedman, D. (2017). "Large-scale audio event discovery in one million YouTube videos," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, Piscataway, NJ, 786–790. doi: 10.1109/ICASSP.2017.7952263

Kalinli, O., and Narayanan, S. (2007). "A Saliency-Based Auditory Attention Model with Applications to Unsupervised Prominent Syllable Detection in Speech," *in Proceedings of the Annual Conference on International Speech Communication Association*, Los Angeles, CL, 1941–1944.

Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., and Fei-Fei, L. (2014). "Large-scale video classification with convolutional neural networks," in *Proceedings of the International. Conference on Computer Vision and Pattern Recognition*, Washington, DC. doi: 10.1109/CVPR.2014.223

Kaya, E. M., and Elhilali, M. (2014). Investigating bottom-up auditory attention. *Front. Hum. Neurosci.* 8:327. doi: 10.3389/fnhum.2014.00327

Kayser, C., Petkov, C. I., Lippert, M., and Logothetis, N. K. (2005). Mechanisms for allocating auditory attention: an auditory saliency map. *Curr. Biol.* 15, 1943–1947. doi: 10.1016/j.cub.2005.09.040

Kell, A. J. E., Yamins, D. L. K., Shook, E. N., Norman-Haignere, S. V., and McDermott, J. H. (2018). A task-optimized neural network replicates human auditory behavior, predicts brain responses, and reveals a cortical processing hierarchy. *Neuron* 98, 630.e16–644.e16. doi: 10.1016/j.neuron.2018.03.044

Kim, K., Lin, K.-H., Walther, D. B., Hasegawa-Johnson, M. A., and Huang, T. S. (2014). Automatic detection of auditory salience with optimized linear filters derived from human annotation. *Pattern Recognit. Lett.* 38, 78–85. doi: 10.1016/j.patrec.2013.11.010

Kriegeskorte, N. (2015). Deep neural networks: a new framework for modeling biological vision and brain information processing. *Annu. Rev. Vis. Sci.* 1, 417–446. doi: 10.1146/annurev-vision-082114-035447

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). "ImageNet classification with deep convolutional neural networks," in *Proceedings of the 25th International Conference on Neural Information Processing Systems*, Vol. 1, Lake Tahoe, NV, 1097–1105.

Kuzovkin, I., Vicente, R., Petton, M., Lachaux, J.-P., Baciu, M., Kahane, P., et al. (2017). Activations of Deep Convolutional Neural Network are Aligned with Gamma Band Activity of Human Visual Cortex. *bioRxiv* [Preprint]. doi: 10.1101/133694

Lee, J., Park, J., Luke, K., and Nam, K. J. (2017). *Sample-level Deep Convolutional Neural Networks for Music Auto-Tagging Using Raw Waveforms*. Available at: https://arxiv.org/pdf/1703.01789.pdf

Mullen, T. (2012). *CleanLine EEGLAB Plugin*. San Diego, CA: Neuroimaging Informatics Tools and Resources Clearinghouse.

Müller, M. M., Gruber, T., and Keil, A. (2000). Modulation of induced gamma band activity in the human EEG by attention and visual information processing. *Int. J. Psychophysiol.* 38, 283–299. doi: 10.1016/S0167-8760(00)00171-9

Nostl, A., Marsh, J. E., and Sorqvist, P. (2012). Expectations modulate the magnitude of attentional capture by auditory events. *PLoS One* 7:e48569. doi: 10.1371/journal.pone.0048569

Oostenveld, R., Fries, P., Maris, E., and Schoffelen, J. M. (2011). FieldTrip: open source software for advanced analysis of MEG, EEG, and invasive electrophysiological data. *Comput. Intell. Neurosci.* 2011:156869. doi: 10.1155/2011/156869

Parkhi, O. M., Vedaldi, A., and Zisserman, A. (2015). "Deep face recognition," in *Proceedings of the British Machine Vision Conference*, Swansea. doi: 10.5244/C.29.41

Petsas, T., Harrison, J., Kashino, M., Furukawa, S., and Chait, M. (2016). The effect of distraction on change detection in crowded acoustic scenes. *Hear. Res.* 341, 179–189. doi: 10.1016/j.heares.2016.08.015

Poria, S., Peng, H., Hussain, A., Howard, N., and Cambria, E. (2017). Ensemble application of convolutional neural networks and multiple kernel learning for multimodal sentiment analysis. *Neurocomputing* 261, 217–230. doi: 10.1016/j.neucom.2016.09.117

Rao Yarlagadda, R. K. (2010). *Analog and Digital Signals and Systems*. New York, NY: Springer. doi: 10.1007/978-1-4419-0034-0

Rodriguez, E., George, N., Lachaux, J. P., Martinerie, J., Renault, B., and Varela, F. J. (1999). Perception's shadow: long-distance synchronization of human brain activity. *Nature* 397, 430–433. doi: 10.1038/17120

Shuai, L., and Elhilali, M. (2014). Task-dependent neural representations of salient events in dynamic auditory scenes. *Front. Neurosci.* 8:203. doi: 10.3389/fnins.2014.00203

Simonyan, K., and Zisserman, A. (2015). "Very Deep Convolutional Networks for Large-Scale Image Recognition," in *Proceedings of the International Conference. Learning Representation*, Banff, AB, 1–14. doi: 10.1016/j.infsof.2008.09.005

Staeren, N., Renvall, H., Martino, F., De Goebel, R., and Formisano, E. (2009). Sound categories are represented as distributed patterns in the human auditory cortex. *Curr. Biol.* 19, 498–502. doi: 10.1016/j.cub.2009.01.066

Tiitinen, H. T., Sinkkonen, J., Reinikainen, K., Alho, K., Lavikainen, J., and Näätänen, R. (1993). Selective attention enhances the auditory 40-Hz transient response in humans. *Nature* 364, 59–60. doi: 10.1038/364059a0

Tordini, F., Bregman, A. S., and Cooperstock, J. R. (2015). "The loud bird doesn't (always) get the worm: why computational salience also needs brightness and tempo," in *Proceedings of the 21st International Conference on Auditory Display (ICAD 2015)*, Graz.

Treue, S. (2003). Visual attention: the where, what, how and why of saliency. *Curr. Opin. Neurobiol* 13, 428–432. doi: 10.1016/S0959-4388(03)00105-3

Tsuchida, T., and Cottrell, G. (2012). "Auditory saliency using natural statistics," in *Proceedings of the Social Neuroscience Meeting*, New Orleans, LA.

Veale, R., Hafed, Z. M., and Yoshida, M. (2017). How is visual salience computed in the brain? Insights from behaviour, neurobiology and modelling. *Philos. Trans. R. Soc. B Biol. Sci.* 372:20160113. doi: 10.1098/rstb.2016.0113

Wang, C.-Y., Wang, J.-C., Santoso, A., Chiang, C.-C., and Wu, C.-H. (2017). "Sound event recognition using auditory-receptive-field binary pattern and hierarchical-diving deep belief network," in *Proceedings of the IEEE/ACM Transaction Audio, Speech, Language Processes*, Piscataway, NJ, doi: 10.1109/TASLP.2017.2738443

Weisberg, S. (2005). *Applied Linear Regression*. Hoboken, NJ: Wiley-Interscience. doi: 10.1002/0471704091

Wolfe, J. M., and Horowitz, T. S. (2004). What attributes guide the deployment of visual attention and how do they do it? *Nat. Rev. Neurosci.* 5, 495–501. doi: 10.1038/nrn1411

Yamins, D. L. K., and DiCarlo, J. J. (2016). Using goal-driven deep learning models to understand sensory cortex. *Nat. Neurosci.* 19, 356–365. doi: 10.1038/nn.4244

Yamins, D. L. K., Hong, H., Cadieu, C. F., Solomon, E. A., Seibert, D., and DiCarlo, J. J. (2014). Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proc. Natl. Acad. Sci. U.S.A.* 111, 8619–8624. doi: 10.1073/pnas.1403112111

Conflict of Interest Statement: MS is employed by Google AI.

The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Neuroscience

ORIGINAL RESEARCH
published: 19 March 2019
doi: 10.3389/fnins.2019.00153

A Tutorial on Auditory Attention Identification Methods

Emina Alickovic [1,2]*, *Thomas Lunner* [1,2,3,4], *Fredrik Gustafsson* [1] *and Lennart Ljung* [1]

[1] Department of Electrical Engineering, Linkoping University, Linkoping, Sweden, [2] Eriksholm Research Centre, Oticon A/S, Snekkersten, Denmark, [3] Hearing Systems, Department of Health Technology, Technical University of Denmark, Lyngby, Denmark, [4] Swedish Institute for Disability Research, Linnaeus Centre HEAD, Linkoping University, Linkoping, Sweden

Auditory attention identification methods attempt to identify the sound source of a listener's interest by analyzing measurements of electrophysiological data. We present a tutorial on the numerous techniques that have been developed in recent decades, and we present an overview of current trends in multivariate correlation-based and model-based learning frameworks. The focus is on the use of linear relations between electrophysiological and audio data. The way in which these relations are computed differs. For example, canonical correlation analysis (CCA) finds a linear subset of electrophysiological data that best correlates to audio data and a similar subset of audio data that best correlates to electrophysiological data. Model-based (encoding and decoding) approaches focus on either of these two sets. We investigate the similarities and differences between these linear model philosophies. We focus on (1) correlation-based approaches (CCA), (2) encoding/decoding models based on dense estimation, and (3) (adaptive) encoding/decoding models based on sparse estimation. The specific focus is on sparsity-driven adaptive encoding models and comparing the methodology in state-of-the-art models found in the auditory literature. Furthermore, we outline the main signal processing pipeline for how to identify the attended sound source in a cocktail party environment from the raw electrophysiological data with all the necessary steps, complemented with the necessary MATLAB code and the relevant references for each step. Our main aim is to compare the methodology of the available methods, and provide numerical illustrations to some of them to get a feeling for their potential. A thorough performance comparison is outside the scope of this tutorial.

Keywords: cocktail-party problem, auditory attention, linear models, stimulus reconstruction, canonical correlation anaysis (CCA), decoding, encoding, sparse representation

OPEN ACCESS

Edited by:
Shih-Chii Liu,
ETH Zürich, Switzerland

Reviewed by:
Jonathan Z. Simon,
University of Maryland, College Park, United States
Alexander Bertrand,
KU Leuven, Belgium

****Correspondence:***
Emina Alickovic
emina.alickovic@liu.se

Specialty section:
This article was submitted to Neuromorphic Engineering, a section of the journal Frontiers in Neuroscience

Received: *15 January 2018*
Accepted: *11 February 2019*
Published: *19 March 2019*

Citation:
Alickovic E, Lunner T, Gustafsson F and Ljung L (2019) A Tutorial on Auditory Attention Identification Methods. Front. Neurosci. 13:153. doi: 10.3389/fnins.2019.00153

1. INTRODUCTION

The first use of the term *cocktail party* in the context of auditory scene analysis appeared in Cherry (1953), where it was used to refer to the challenge of focusing on a single sound source, often a speech stream, while suppressing other unwanted sounds in a noisy and complex background. The ability to segregate and follow a sound source of interest in a cocktail party environment is one of the hallmarks of brain functions. Although this is a highly ill-posed problem in a mathematical sense, the human brain instantly solves this problem, with a compelling ease and accuracy that is difficult to be matched by any currently available algorithm. However, recent studies have shown the potential of model-based algorithms to assist intelligent hearing aids, and the purpose of this tutorial is to provide a rather broad coverage of the mathematical tools available for solving the cocktail party problem. The algorithms are illustrated on examples from datasets previously used in several studies. The algorithms in this tutorial are relatively simple and computationally inexpensive, although further research on algorithm optimization is needed to achieve real-time performance.

Neural networks and cognitive processes assist the brain in parsing information from the environment (Bregman, 1994). These processes allow us to perform everyday tasks with remarkable ease and accuracy, for example, enjoying our time with friends in crowded places such as restaurants and cafes while being alert to salient sound events such as someone calling our name. The intrinsic complexity of the background is hidden by the brain's process of perceiving and selectively attending to any sound source: (a) competing acoustic sources (stimuli) emit acoustic signals and (b) are subsequently mixed, (c) the mixture of incoming sound streams enters the ear(s), (d) this mixture is resolved such that (e) the attended sound is perceived, and (f) the remaining, unwanted streams of sound are effectively attenuated within the human auditory cortex.

There are many studies on deciphering human auditory attention. The majority of these studies have generally focused on brain oscillations (Obleser and Weisz, 2011; Weisz et al., 2011; Henry et al., 2014) and speech entrainment (Ding and Simon, 2012a,b; Mesgarani and Chang, 2012; Pasley et al., 2012; Mirkovic et al., 2015; O'Sullivan et al., 2015, 2017; Ekin et al., 2016; Biesmans et al., 2017; Fuglsang et al., 2017; Kaya and Elhilali, 2017; Van Eyndhoven et al., 2017; Haghighi et al., 2018) in electroencephalography. Broadly speaking, the two most common approaches in the development of speech (envelope) entrainment are (1) *encoding*, i.e., estimating the neural responses from the sound features, and (2) *decoding*, i.e., estimating the sound from the neural response features. In most of these studies, the linear filters are computed using "dense" least-squares (LS) optimization tools. However, it is also possible to exploit an alternative approach based on sparse estimation. Sparse estimation has shown great potential in diverse signal processing applications (Sepulcre et al., 2013; Akram et al., 2016, 2017; Rao et al., 2016; Miran et al., 2018).

As a further alternative to encoding and decoding, *bidirectional* hybrid approaches (Dmochowski et al., 2017; de Cheveigné et al., 2018), such as *canonical correlation analysis (CCA)*, aim to combine the strengths (and weaknesses) of encoding and decoding methods. A recent work (de Cheveigné et al., 2018) supports the view that CCA-based classifier schemes may provide higher classification performance compared to encoding and decoding methods.

The applications of attention deciphering are diverse, including robotics, brain-computer interface (BCI), and hearing applications (see e.g., Li and Wu, 2009; Lunner and Gustafsson, 2013; Gao et al., 2014; Khong et al., 2014; Lunner, 2015; Tsiami et al., 2016). In fact, there is currently increased interest in auditory attention identification in, for instance, the hearing aid industry. The reason for this interest is that for a hearing-impaired listener, the ability to selectively attend to a desired speaker in a cocktail party situation is highly challenging. With an aging population with an increasing number of hearing-impaired individuals, increased understanding of the underlying mechanisms of the cocktail party problem is highly needed. Along the same lines, the hearing aid companies are also interested in applying auditory attention deciphering (AAD) techniques for cognitive control of a hearing aid and its noise-reduction algorithms (Das et al., 2017; Van Eyndhoven et al., 2017).

However, despite the increasing interest in this problem from the audiology and neuroscience research communities (Fritz et al., 2007; Mesgarani and Chang, 2012; Jääskeläinen and Ahveninen, 2014; Kaya and Elhilali, 2017), the basis for the computational models of the brain's ability to selectively attend to different sound sources remains unclear.

The primary objective of this study is to explain how to use linear models and identify a model with sufficiently high performance in terms of attention deciphering accuracy rates and computational time. Our ultimate goal is to provide an overview of the state-of-the-art for how linear models are used in the literature to *decipher human auditory attention* by exploiting the *brain activity* elicited during attentive listening to a single sound source in an acoustically complex background.

This contribution focuses on the classification of auditory attention by using multivariate linear models. Consequently, we do not cover other aspects of auditory attention and scene analysis, and to limit the scope, we do not cover (computational) auditory scene analysis (CASA) (Wang and Brown, 2006; Wang et al., 2009; Snyder et al., 2012; Gutschalk and Dykstra, 2014; Alain and Bernstein, 2015; Simon, 2017), auditory attention modeling (Kaya and Elhilali, 2017), speech masking (Scott and McGettigan, 2013; Evans et al., 2016), and sound segregation and localization (Ahveninen et al., 2014; Middlebrooks, 2017).

An important note regarding the current auditory attention identification methods is that these methods require access to the clean speech signals, which are usually not available in practice. CASA methods are then necessary to provide these. Recent attempts to perform attention deciphering without access to the individual speakers (but noisy speech mixtures instead) may provide a useful way to approach solving this problem. The study of S. Van Eyndhoven (Van Eyndhoven et al., 2017), later improved by Das Das et al. (2017), was the first that tackled this problem, based on beamforming methods. O'Sullivan later also did a similar study, using deep learning (O'Sullivan et al., 2017). After separating the individual speakers in the mixture, these

Abbreviations: AAD, auditory attention deciphering; ADMM, alternating direction method of multipliers; AIC, Akaike–s information criterion; BIC, Bayesian information criterion; CASA, computational auditory scene analysis; CCA, canonical correlation analysis; CCV, correlation coefficient value; CV, cross-validation; EEG, electroencephalography; FBS, forward-backward splitting; FIR, finite impulse response; IIR, infinite impulse response; LASSO, least absolute shrinkage and selection operator; LOOCV, leave-one-out cross-validation; LS, least squares; MEG, magnetoencephalography; MFCC, Mel-frequency cepstral coefficients; ML, machine learning; MSE, mean squared error; SIMO, single input multiple output; SISO, single input single output; SPARLS, sparse recursive least squares; SR, stimulus reconstruction; SVD, singular value decomposition; SVM, support vector machine; TLS, total least squares; TRF, temporal response function.

studies used the linear models discussed in this tutorial to identify the sound source of a listener's interest.

The outline of this contribution is as follows. To obtain accurate attention deciphering using EEG (electroencephalography) / MEG (magnetoencephalography) sensors, several important factors need to be considered. First, the algorithms that are currently used to identify the attended sound source need to be accurately described, which is the topic of section 2. Note that we must always first preprocess the data to avoid problems in the later encoding/decoding procedures, which is also a topic of section 2. Based on the analysis of the models in section 2, we can construct different models. In section 3, we discuss the datasets used in this contribution to study different auditory attention identification methods. The practical implementation of the discussed algorithms is the topic of section 4, where we provide experimental results for some different examples and datasets. We end this contribution with some concluding remarks and (potential) future improvements in section 5.

2. LINEAR MODELS FOR AUDITORY ATTENTION DECIPHERING

In this section, we explain the basics of linear modeling. Furthermore, we introduce some of the concepts from machine learning (ML) that are frequently used in the auditory attention identification literature. The last decade has witnessed a large number of impressive ML applications that involve large amounts of data, and our application of audio-EEG data is one area that has thus far remained rather unexplored. The subject of designing the linear models is introduced in section 2.1. How to select the model is a crucial part of any estimation problem. Thus, we discuss different modeling approaches in sections 2.3–2.4.

2.1. The Sound and EEG Signals

We assume that at any given point in space, a time-varying sound pressure exists that originates from n_u sound streams $p_i(t)$, $i = 1, 2, \ldots, n_u$, emitted by one or more sound sources (e.g., individual talkers and loudspeakers). The resulting sound pressure can be conceptually written as a sum

$$p(t) = \sum_{i=1}^{n_u} p_i(t). \tag{1}$$

This mixture is what the ear decodes and what can be sampled by a microphone. The latter results in a discrete time signal $p[k] = p(kT_s)$, where T_s is the sampling interval, which typically corresponds to a sampling frequency of $f_s^p = 1/T_s = 44100$ Hz.

The EEG signals are sampled by n_y EEG electrodes denoted $y_j[k]$, $j = 1, 2, \ldots, n_y$. The EEG sampling frequency f_s^y is considerably smaller than the sampling frequency of the sound f_s^p. Typical values in experiments in this field are $n_u = 2$, $n_y = \{64, 128\}$ and $f_s^y = 512$ Hz. To synchronize the data streams to the same sampling frequency, the ratio f_s^p/f_s^y defines a decimation factor that is needed to reduce the sampling rate of the sound. This downsampling needs to be done only after the envelope extraction of the individual sound sources $p_i(t)$. In the following paragraphs we will describe each of these steps in more detail.

Next, we present the basic steps that are commonly used in practice in this application:

- Extract the envelope of the audio signal, which can be performed in several ways. A complete overview of the envelope extraction methods for AAD is presented in Biesmans et al. (2017). The resulting sound signal will be denoted $u[k]$, which in the literature is supposed to be the sum $u[k] = \sum_{i=1}^{n_u} u_i[k]$ of n_u envelopes $u_i[k]$, but it should be noted that $u[k]$ will never be used in practice as the access to the individual sound streams $u_i[k]$ is needed when applying AAD techniques. Speech envelopes are spectrotemporally sparse, and therefore the equation is approximately true enough for the purposes used here.
- Downsample the EEG signal and the audio signals to the same sampling rate (e.g., to 64 Hz), which can be performed using the *nt_dsample* function from the NoiseTools toolbox (http://audition.ens.fr/adc/NoiseTools/) (Yang et al., 1992; Ru, 2001) or MATLAB built-in downsampling methods, such as *decimate* or *resample* functions.
- Bandpass filter both the EEG and the sound signals using a bandpass filter between 1 and 8 Hz, which is the frequency interval where the brain processes auditory information (Zion Golumbic et al., 2013).

The following code performs this operation, as was proposed in O'Sullivan et al. (2015):

```
p      = resample(p,44096,44100);
   % Resample to a multiple of 64 Hz
pc     = hilbert(p);
   % Transform from real to analytic signal
u      = decimate(abs(pc),44096/64);
  % Downsampling to 64 Hz, including an
    anti-alias
[b,a] = butter(3,[2 8]/64*2);
  % Bandpass filter with passband [2,8] Hz
uf     = filter(b,a,u);
  % Causal filtering to keep causality
```

Without loss of generality, we will assume that the attended sound source is $u_1[k]$, while the other sources, $u_i[k]$ for $i > 1$, represent nuisance sound sources.

2.2. Data Notation

We denote all scalars by lowercase letters, e.g., w, and all vectors and matrices by uppercase letters, e.g., W, unless stated otherwise. The (p, q) entry, $p-$th row and $q-$th column in W are expressed as $[W]_{p,q}$, $W_{p,:}$ and $W_{:,q}$, respectively, and the $p-$ th entry in vector U is expressed as U_p. The transpose of the matrix W is denoted as W^T. The functions $\|W\|_F$ (Frobenius norm) and $\|U\|_2$ (Euclidean or l_2 norm) return the matrix-valued norm and vector-valued norms, respectively, and $\|W\|_F^2 = \mathrm{trace}(\mathrm{W}^\mathrm{T}\mathrm{W})$ and $\|\mathrm{U}\|_2^2 = \mathrm{U}^\mathrm{T}\mathrm{U}$. The l_1 penalty term is defined as $\|W\|_1 = \sum_{p,q} |[W]_{p,q}|$. The letter n with an index will denote the dimension of a vector, for instance, n_y and n_u, as previously introduced.

To have a compact notation avoiding one or more indices, we will summarize the data in the data vectors U_i and Y_j, the data matrices U and Y, which are defined as follows:

$$\left[Y_j\right]_k = y_j[k], \qquad k = 1, \ldots, N, \quad j = 1, 2, \ldots, n_y, \tag{2}$$

$$\left[Y\right]_{kj} = y_j[k], \qquad k = 1, \ldots, N, \quad j = 1, 2, \ldots, n_y, \tag{3}$$

and similarly for U and U_i.

For a model that takes the latest n_a data points into account, we define the Hankel matrix

$$\begin{aligned}\left[\mathcal{H}(Y_j)\right]_{kn} &= y_j[n_a + k - n], k = 1, \ldots, N - n_a + 1, \\ n &= 1, 2, \ldots, n_a,\end{aligned} \tag{4}$$

and similarly for $\mathcal{H}(U_i)$. We will refer to the data as Y_j, U_i, Y, U.

2.3. Correlation-Based Learning

Correlation-based learning aims to find the pattern in the EEG signal that best correlates to the target sound $u_1(t)$ with less correlation to the distracting sounds $u_i(t)$, $i \neq 1$. Typical correlation-based learning approaches are:

(1) Cross-correlation:

(a) Zero-lag cross-correlation: The normalized covariance between each speech signal U_i and each EEG signal Y_j, i.e., $c_{ij} = \frac{Cov(U_i, Y_j)}{\sqrt{Var(U_i)Var(Y_j)}}$. The drawback with zero-lag cross-correlation is that it assumes that both U_i and Y_j are synchronized in time, which is hardly the case.

(b) Time-lag cross-correlation: Here one of the sequences is delayed (time-lagged) before the correlation is computed. There is here one extra degree of freedom, so one has to maximize cross-correlation with respect to this lag.

(2) Canonical Correlation Analysis (CCA).

The disadvantage of correlation-based approaches is that they compare sample by sample for the entire batch and are thus less effective if there is a dynamical relationship between U and Y, in which case only a few samples around the current time would exhibit a significant correlation. CCA corresponds to a linear model of the whole segment of speech, and the model is by construction non-causal. The segment length is an important design parameter corresponding to the model order in FIR models.

2.4. Linear Models

The linear filter formalism we use is based on the shift operator q defined by $q^{-n}x[k] = x[k-n]$ and $q^n x[k] = x[k+n]$ for all n. A causal FIR filter can then be written as

$$\begin{aligned}y_j[k] = B_i(q)u_i[k] &= (b_{i0} + b_{i1}q^{-1} + \cdots + b_{in_b}q^{-n_b})u_i[k] \\ &= b_{i0}u_i[k] + b_{i1}u_i[k-1] + \cdots + b_{in_b}u_i[k-n_b].\end{aligned} \tag{5}$$

Similarly, an IIR filter can be written as

$$\begin{aligned}&A_j(q)\, y_j[k] = B_i(q)u_i[k], \\ &\quad (1 + a_{j1}q^{-1} + \cdots + a_{jn_a}q^{-n_a})y_j[k] = y_j[k] + a_{j1} \\ &\quad y_j[k-1] + \cdots + a_{jn_a}y_j[k-n_a] = B_i(q)u_i[k], \\ &\quad\quad y_j[k] = -a_{j1}y_j[k-1] - \cdots - a_{jn_a}y_j[k-n_a] + B_i(q)u_i[k].\end{aligned} \tag{6}$$

It should also be noted that (6) does not represent the general form of $A_j(q)$, i.e., the filter $A_j(q)$ can be generalized so that positive exponents can also be used for q, as explained in the remainder of this section.

Implementation requires stability. The IIR filter specified by $A_j(q)$ can be causally stably implemented *forward* in time only if all roots to the polynomial $A_j(q)$ are *inside* the unit circle. We denote such a filter with $A^f(q)$. Conversely, a filter with all roots *outside* the unit circle can be anti-causally implemented in a stable way *backward* in time, and we denote such a filter with $A^b(q)$. Any IIR filter can be split into two parts with one causal and one anti-causal part. For more details on these issues, see basic text books in signal processing, for instance (Gustafsson et al., 2010).

Given this brief background, there are two fundamentally different ways to define a model for listening attention, forward or backward in time,

$$y_j[k] = \sum_{i=1}^{n_u} \frac{B_i^f(q)}{A_j^f(q)} u_i[k] + e_j^f[k] \tag{7}$$

$$u_i[k] = \sum_{j=1}^{n_y} \frac{A_j^b(q)}{B_i^b(q)} y_j[k] + e_i^b[k] \tag{8}$$

The first model corresponds to the forward model (using superscript f for forward), where each EEG signal is explained as a sum of filtered sound signals plus additive noise to account for measurement errors and model imperfections, while the other model corresponds to the inverse backward model (denoted with superscript b). Another note, positive exponents are used for q in backward models. It is assumed that both filters are causally stable, implying that A_j^f and B_i^b are polynomials with all roots inside the unit circle. The roots of B_j^f and A_i^b can be both inside and outside the unit circle generally. This means that inverting the forward model does not give a causally stable backward model, and is thus not in general a valid backward model. In other words, the models are not identical or related in simple terms. Also the noise realizations $e_j^f[k]$ and $e_i^b[k]$ are different and can have quite different characteristics.

Note, however, that one can mix a forward and backward model in a non-causal filter. Combining both model structures gives the linear filter

$$y_j[k] = \sum_{i=1}^{n_u} \left(\frac{B_i^f(q)}{A_j^f(q)} + \frac{B_i^b(q)}{A_j^b(q)} \right) u_i[k] + e_j[k], \tag{9}$$

and similarly for the backward model. This can be seen as a non-causal filter with poles both outside and inside the unit circle.

Given such a linear filter, one can reproduce an estimate $\hat{y}_j[k]$ of the EEG signal. For instance, the causally stable part can be implemented with

```
for j=1:ny
   yijhat[:,j]=filter(bf(j,:),af(i,:),U(:,i));
end
yihat=sum(yijhat,2);
```

Here, `af` denotes the matrix of polynomial coefficients for the polynomials $A_i^f(q)$ and so forth. A good model should provide a small estimation error $y_j[k] - \hat{y}_j[k]$. We will return to the issue of parameter estimation, or system identification (Ljung, 1998), shortly, but note that there is no good model in the traditional sense. All linear models share the property that the prediction errors are of the same order as the signal itself. In other words, the least squares loss function will be only somewhat smaller than the sum of squared measurements, which would be the least squares loss function for the trivial signal predictor $\hat{y}_j[k] = 0$ for all times k and all channels j.

The use of IIR (infinite impulse response) models is still unexplored in this area; thus, we will restrict the discussion to FIR (finite impulse responses) models, having denominators $A_j^f(q) = 1$ in (7) and $B_i^b(q) = 1$ in (8) equal to unity, in the following.

2.5. FIR Models for Encoding and Decoding

Here, we explain two modeling perspectives that are widely used in auditory research: *forward* and *inverse* (backward) modeling. Encoding and decoding are two special cases of supervised learning of forward and backward models, respectively (Haufe et al., 2014). The encoding and decoding models applied in cognitive electrophysiology are described in greater detail in Holdgraf et al. (2017). The traditional encoding approach attempts to predict neural responses (EEG) given the *sound stimulus*

$$y_j[k] = B_i^{f(q)} u_i[k] + e_j^f[k] \qquad \text{(encoding)} \tag{10}$$

Note that there is one filter $B_i^{(q)}$ for each input and output combination. Here, $\hat{y}_j[k] = B_i^{f(q)} u_i[k]$ will be referred to as a neural prediction.
In contrast, the decoding approach attempts to extract the sound from the neural responses (EEG)

$$u_i[k] = \sum_{j=1}^{n_y} A_j^b(q) y_j[k] + e_i^b[k] \qquad \text{(decoding)} \tag{11}$$

Similarly, $\hat{u}_i[k] = \sum_{j=1}^{n_y} A_j^b(q) y_j[k]$ will be referred to as a reconstructed stimulus. Note that $\hat{u}_i[k]$ usually captures the neural responses $y_j[k]$ after stimuli presentation at time step k. The stimulus reconstruction (SR) approach, which has received the greatest attention in the auditory literature, compares the reconstructed sound waveform with the actual waveform to make a decision on the attended sound source. **Figure 1** illustrates the difference between the encoding and decoding approaches.

2.6. Parameter Estimation

The encoding and decoding models (10)–(11) can be more conveniently written in matrix-vector form as

$$Y_j = \mathcal{H}(U_i) B_i^f + E_j^f, \tag{12}$$

$$U_i = \sum_j \mathcal{H}(Y_j) A_j^b + E_i^b, \tag{13}$$

using the Hankel matrices defined in (4), and B_i^f and A_j^b are the vectors consisting of the coefficients of the polynomials $B_i^{f(q)}$ defined in (5) and $A_j^f(q)$ defined in (6), respectively.

The model in (12) defines an estimation error

$$\epsilon_j = Y_j - \mathcal{H}(U_i) B_i^f, \tag{14}$$

from which one can define an LS loss function

$$W(B_i^f) = \| Y_j - \mathcal{H}(U_i) B_i^f \|_2^2. \tag{15}$$

This loss function defines a quadratic function in the parameters B_i. Minimization provides the LS estimate as

$$\hat{B}_i^f = \underset{B_i^f}{\operatorname{argmin}}\, W(B_i^f) = \mathcal{H}(U_i)^\dagger Y_j \tag{16}$$

where $\mathcal{H}^\dagger(U_i) = [\mathcal{H}(U_i)^T \mathcal{H}(U_i)]^{-1} \mathcal{H}(U_i)^T$ denotes the Moore-Penrose pseudoinverse. Similarly,

$$\hat{A}_j^b = \underset{A_j^b}{\operatorname{argmin}}\, W(A_j^b) = \mathcal{H}(Y_j)^\dagger U_i \tag{17}$$

The corresponding operations in MATLAB are given below.

```
for i=1:nu
   for j=1:ny
      HUij = hankel(U(1:end-nb,1),
         U(end-nb:end,1));
      bhat(i,j,:) = HUij\ Y(nb:end,j);
      W(i,j) = norm(Y(nb:end,j) -
         HUij*squeeze(bhat(i,j,:)));
   end
end
```

The backslash operator solves the LS problem in a numerically stable way using a QR factorization of the Hankel matrix. For model structure selection, that is, the problem of selecting the model order n_b, the QR factorization enables all parameter estimates and cost functions for lower model orders to be obtained for free.However, model order selection is prone to overfitting; thus, in practice, one has to be careful when selecting n_b not only based on the LS cost function.

2.7. Regularization

Due to the challenge of avoiding overfitting, encoding and decoding techniques should be complemented with a regularization method, which basically adds a penalty for the

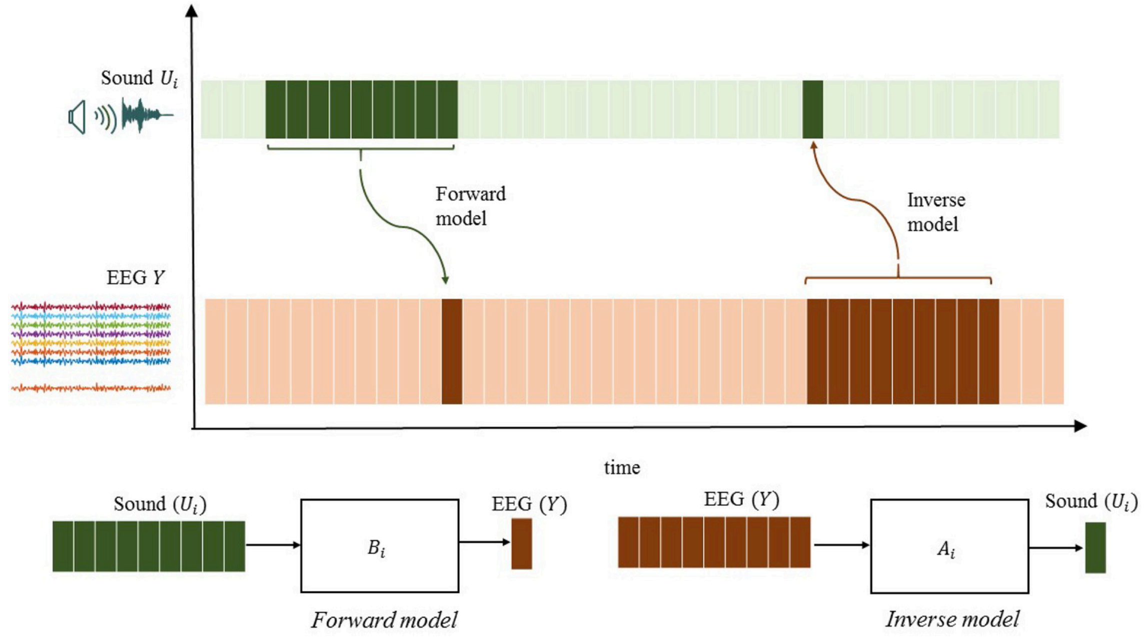


FIGURE 1 | Illustration of the essential difference between encoding and decoding methods.

model complexity to (15). In general terms, regularized LS can be expressed as

$$V_N(B_i^f) = W_N(B_i^f) + \lambda \mathbf{g}(B_i^f) \tag{18}$$

where N is the number of data and $\mathbf{g}$ is generally called a *regularizer* or *regularization function*, and it is typically non-smooth and possibly non-convex and $\lambda \in \mathbb{R}^+$ is a penalty parameter. The regularization function is most commonly selected as the l_p norm, i.e.,

$$\underset{B_i^f}{\text{minimize}}\ \frac{1}{2}\|Y_j - \mathcal{H}(U_i)B_i^f\|_2^2 + \lambda\|B_i^f\|_p \tag{19}$$

With l_2, the problem given in (19) has the analytic solution

$$\hat{B}_i^f = (\mathcal{H}(U_i)^T\mathcal{H}(U_i) + \lambda I)^{-1}\mathcal{H}(U_i)^T Y_j \tag{20}$$

Similarly,

$$\hat{A}_j^b = (\mathcal{H}(Y_j)^T\mathcal{H}(Y_j) + \lambda I)^{-1}\mathcal{H}(Y_j)^T U_i \tag{21}$$

However, l_2 regularization does not do a variable subset selection.

Methods that directly aim to limit the number of parameters n_b include Akaike's information criterion AIC, where $U_N = \log(W_N) + 2n_b/N$, and his improved suggestion Bayesian information criterion BIC $U_N = \log(W_N) + \log(n_b)/N$. Note that n_b is the l_0 norm of B_i^f, a fact that is used in many recent approaches of sparse modeling based on efficient algorithms for convex optimization. However, the l_0 term is not convex, but the l_1 norm is, and it is in practice a good approximation of the l_0 norm (Ramirez et al., 2013). This trick to obtain a feasible problem belongs to the class of convex relaxations.

The use of the l_1 norm to induce sparsity is frequently referred to as the *least absolute shrinkage and selection operator (LASSO)* (Tibshirani, 1996). This formulation can be used to identify the sparse spatial-temporal resolution and reveal information about the listening attention.

Conceptually, sparse signal estimation depicts a signal as a sparse linear combination of active elements, where only a few elements in B_i are non-zero. The sparse estimation can be further improved with group sparsity, in other words, grouping the elements in B_i^f (or A_j^b) and considering the groups of elements to be singletons, where a relatively small number of these groups is active at each time point. The group sparse estimation problem is frequently referred to as *group LASSO* (Yuan and Lin, 2006).

One way to solve sparse (l_1-regularized) optimization problems is to apply the Expectation Maximization (EM) algorithm. One such example is the sparse (l_1-regularized) recursive least squares (SPARLS) algorithm introduced in Babadi et al. (2010). The SPARLS algorithm estimates a sparse forward model using a dictionary of atoms, which is posed as a linear estimation problem. It has already been successfully used in AAD studies to estimate the encoding model (Akram et al., 2017). The authors concluded that the SPARLS algorithm could improve performances over the conventional (l_2-regularized) linear estimation methods. Another way to solve sparse (l_1-regularized) optimization problems is based on proximal splitting algorithms, one of which is a forward-backward splitting (FBS) algorithm, also referred to as the proximal gradient method (Combettes and Pesquet, 2011). Recently, Miran et al. (2018) suggested a Bayesian filtering approach for sparse estimation to tackle AAD. In their work, the authors used FBS procedure for decoding/encoding model estimation in real-time. In our examples, we use an algorithm called ADMM (alternating direction method of multipliers) to solve sparse (l_1-regularized) optimization problems in an efficient way that normally requires very few iterations of simple computations to converge. The reason is 2-fold: the ADMM is simpler and easier to work with, since its iterative solution can be implemented via simple analytical expressions, and it has a proven fast convergence (Boyd et al., 2011).

2.8. SIMO Formulation

For simplicity, we have thus far considered single-input single-output SISO models, where the model relates one sound source to one EEG signal, and conversely for the reverse model. It is, however, simple to extend the model to a single-input multiple-output (SIMO) model that aims to explain all EEG data based on one sound stimulus at a time. The principle is that the sound stimulus that best explains the observed EEG signals should correspond to the attended source.

The SIMO FIR model for each sound source is defined as

$$Y = \mathcal{H}(U_i)\boldsymbol{B}_i^f + E_i^f, \ \ i = 1, 2, \cdots, n_u, \tag{22}$$

where $\boldsymbol{B}_i^f$ is an $n_b \times n_y$ matrix.

In the literature, the filter $\boldsymbol{B}_i$ is frequently referred to as a *temporal response function* (TRF), and the corresponding case for the backward approach leads to an $n_a \times n_y$ matrix $\boldsymbol{A}^b$, where $\boldsymbol{A}^b = vec(A_j^b)$, referred to as a *decoder*.

2.8.1. Example 1

If we assume that $n_b = 10$ and $n_y = 6$, then we can estimate $\hat{\boldsymbol{B}}_i^f$, as shown in **Figure 2**. The first panel in **Figure 2** shows the "dense" filter $\boldsymbol{B}_i$, where all the elements are active (non-zero). The second panel in the same figure illustrates the sparse matrix resulting from *LASSO*. Here, LASSO finds the active elements in the filter $\boldsymbol{B}_i^f$ (elements in white are non-active or zero-valued elements). The prior knowledge of how the time lags and electrodes form the groups can be incorporated with group LASSO to obtain filters similar to those in the last two panels shown in **Figure 2**, respectively. If for instance some of the EEG signals are completely uncorrelated with the sound stimulus, the reconstruction error will not increase if these EEG signals are left out. A general rule of thumb for intuition in system identification is that zero is the best prediction of zero mean white noise. Any other prediction will increase the cost. That is the rationale with LASSO, don't attempt to predict white noise, even if reasons of over learning may indicate that it is possible.

2.9. CCA vs. Linear FIR Filters

The main difference between the forward and backward models is how the noise enters the models 7 and 8, respectively. The general rule in LS estimation is that the noise should be additive in the model. If this is not the case, then the result will be biased. However, if there is additive noise to both the input U_i and the output Y_j, then the total least squares (TLS) algorithm can be used. TLS basically weights both noise sources together in an optimal way. The standard implementation of TLS is based on a singular value decomposition (SVD) of the Hankel matrix $\mathcal{H}(U_i)$.

CCA combines the encoding and decoding approaches:

$$B_i^f(q)u_i[k] \sim \sum_{j=1}^{n_y} A_j^b(q)y_j[k] + e[k] \quad \text{(CCA)} \tag{23}$$

and involves *solving a generalized eigenvalue problem*.

Table 1 provides a summary of the discussed linear models.

Solving a generalized eigenvalue problem is more costly for high-dimensional data in a computational sense (Watkins, 2004). In particular, the sample covariance matrices of high-dimensional data become singular (do not have an inverse), which leads to more complex associated generalized eigenvalue problems.

A regularized CCA (rCCA) is often proposed to address this problem (Hardoon et al., 2004). This particular problem may be overcome by formulating CCA as an LS problem, as in Sun et al. (2011), where the classical CCA (and rCCA) is formulated as an LS problem, and LS optimization methods are used to solve it. However, this topic is beyond the scope of this paper and is left for future work.

2.10. Non-linear Models

Linear models should always be examined first in the spirit of "try simple things first." An alternative method to estimate the attended sound source would be to exploit non-linear models. There are, however, many problems in ML that require non-linear models. The principle is the same, but the algorithms are more complex. In short, the linear model $Y_j = \mathcal{H}(U_i)B_i + E_j$ in (12) is replaced with

$$Y_j = f(U_i, B_i) + E_j. \tag{24}$$

Among the standard model structures for the non-linear function f, we mention the Wiener and Hammerstein models, support vector machines and neural networks (Taillez et al., 2017; Deckers et al., 2018; Akbari et al., 2019). Indeed, non-linear models can be used to decipher attention, but the focus of this paper is on linear models because they are simpler to understand and implement.

3. EXAMINED DATASETS

We have used both simulated data and real datasets to evaluate the aforementioned algorithms. Simulations provide a simple way to test, understand and analyze complex algorithms in general, as well as in this case. We use synthetic sound and EEG signals to illustrate the aforementioned algorithms, but real data have to be used to evaluate the potential for applications.

In our contribution, we are revisiting two datasets that were anonymized and publicly available upon request by the previous authors. The publications from which the data originated (see references Power et al., 2012; Fuglsang et al., 2017) state that the data were collected with the approval of the corresponding ethical bodies and with due process of informed consent.

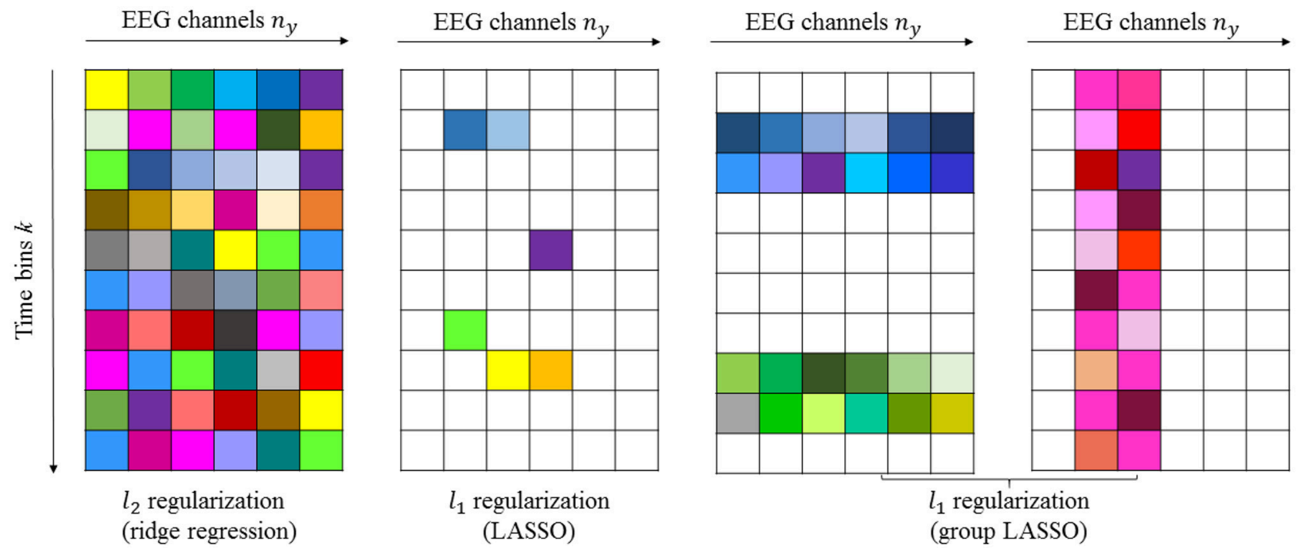

EEG channels n_y
EEG channels n_y
EEG channels n_y
EEG channels n_y
Time bins k
l_2 regularization (ridge regression)
l_1 regularization (LASSO)
l_1 regularization (group LASSO)

FIGURE 2 | Schematic illustration of dense and sparse modeling. The first panel shows the "dense" filter resulting from l_2-penalized LR. The second, third, and fourth panels show the sparse filters resulting from l_1-penalized LR generated by LASSO and group LASSO. Non-active (zero-valued) elements are shown in white.

The *first real dataset* is characterized as follows:

- The subjects were asked to attend to a sound source on either the left u_1 or the right u_2 side.
- The subjects maintained their attention on one sound source throughout the experiment.
- Each subject undertook 30 trials, each 1 min long.
- Each subject was presented with two works of classic fiction narrated in English in the left and right ears.
- Full-scalp EEG data were collected at a sampling frequency of 512 Hz with $n_y = 128$ number of electrodes.
- Sound data were presented at a sampling frequency of 44.1 kHz.

This dataset was first presented and analyzed in Power et al. (2012) and O'Sullivan et al. (2015). Henceforth, we refer to this dataset as the *O'Sullivan dataset*.

The *second dataset* can be described as follows:

- The subjects were asked to *selectively* attend to a sound source on the left u_1 or right u_2 side in different simulated acoustic environments (anechoic, mildly reverberant classroom, and highly reverberant Hagia Irene Church) throughout the experiment.
- The subjects switched their attention from one sound source to another throughout the experiment.
- Each subject was presented with two works of classic fiction narrated in Danish.
- Each subject undertook 60 trials, each 50 s long accompanied by multiple choice questions.
- Full-scalp EEG data were collected at a sampling frequency of 512 Hz with $n_y = 64$ number of electrodes.
- Sound data were presented at a sampling frequency of 44.1 kHz.

This dataset was first presented and analyzed in Fuglsang et al. (2017), and we will refer to this dataset as the *DTU dataset*.

We randomly selected twelve subjects from each dataset to assess the potential benefits that might result from the different linear models considered in this contribution. The reason for this approach is that our main contribution is to provide a tutorial of methods and examples of their use, not to obtain a final recommendation on which method is the best in general.

There are several toolboxes that are useful when working with real datasets. First, there are at least two toolboxes available for loading EEG data: (1) the EEGLab toolbox (https://sccn.ucsd.edu/eeglab/) (Delorme and Makeig, 2004) and (2) the FieldTrip toolbox (http://www.fieldtriptoolbox.org/) (Oostenveld et al., 2011). For more details on importing EEG data with EEGLab and FieldTrip, see **Appendix**. Then, linear trends can be removed, and the EEG data can be normalized using functions in the NoiseTools toolbox (de Cheveigné and Simon, 2008a,b; de Cheveigné, 2010, 2016).

4. COMPUTATIONAL MODELS IN PRACTICE

In this section, we apply the presented algorithms to the two datasets described in Section 3. All experiments were performed on a personal computer with an Intel Core(TM) i7 2.6 GHz processor and 16 GB of memory, using MATLAB R2015b. Note that for notational simplicity we shall take $\boldsymbol{A}^b = \boldsymbol{A}$ and $\boldsymbol{B}_i^f = \boldsymbol{B}_i$ in the remainder of this section.

We start by discussing two main alternatives to train the models and estimate the de/en - coders ($\boldsymbol{A}$ or $\boldsymbol{B}$):

1) Treating each trial as a single least-squares LS problem and estimating one de/en-coder for each training

TABLE 1 | An overview of linear methods.

Learning representations	Approaches	Mathematical formulations	Optimization problem	Relevant references
Correlation-based learning	*Cross-Correlation* CCA	$B_i^f(q)u_i[k] \sim \sum_{j=1}^{n_y} A_j^b(q)y_j[k] + e[k]$	Generalized eigenvalue problem	Biesmans et al., 2017; Dmochowski et al., 2017; de Cheveigné et al., 2018; de Cheveigné et al., 2019
Model-based learning	Forward modeling Supervised case: *Encoding*	$y_j[k] = B_i^f(q)u_i[k] + e_j^f[k]$	Least-squares	Ding and Simon, 2012a; Di Liberto et al., 2015; Alickovic et al., 2016, in rewiev; Fiedler et al., 2017, 2019; Hjortkjær et al., 2018; Kalashnikova et al., 2018; Lesenfants et al., 2018; Lunner et al., 2018; Verschueren et al., 2018; Wong et al., 2018
	Inverse/backward modeling Supervised case: *Decoding*	$u_i[k] = \sum_{j=1}^{n_y} A_j^b(q)y_j[k] + e_i^b[k]$		Mirkovic et al., 2015; O'Sullivan et al., 2015, 2017; Aroudi et al., 2016; Das et al., 2016, 2018; Presacco et al., 2016; Biesmans et al., 2017; Fuglsang et al., 2017; Van Eyndhoven et al., 2017; Zink et al., 2017; Bednar and Lalor, 2018; Ciccarelli et al., 2018; Etard et al., 2018; Hausfeld et al., 2018; Narayanan and Bertrand, 2018; Schäfer et al., 2018; Vanthornhout et al., 2018; Verschueren et al., 2018; Wong et al., 2018; Akbari et al., 2019; Somers et al., 2019

trial separately, and averaging over all training de/en-coders (Crosse et al., 2016).

$$B_i^{avg} = 1/K \sum_k \left[(\mathcal{H}(U_{i,k})^T \mathcal{H}(U_{i,k}))^{-1} \mathcal{H}(U_{i,k})^T Y_{j,k} \right] \quad (25)$$

2) Concatenating all training trials in a single LS problem (Biesmans et al., 2017).

$$B_i^{conc} = \left[\sum_k \mathcal{H}(U_{i,k})^T \mathcal{H}(U_{i,k}) \right]^{-1} \left[\sum_k \mathcal{H}(U_{i,k})^T Y_{j,k} \right] \quad (26)$$

Here K is a total number of trials. We may point to the following aspects that are to be considered when discussing the two alternatives:

- *Averaging LS per-trial estimates is not equivalent with the correct overall LS estimate.* It is easy to show that the two alternatives will result in different estimates, even if the discontinuities and boundary effects are correctly treated. One can show algebraically that -under some technical conditions- the second alternative will yield a better estimator with a lower (co-)variance on its entries. For a more detailed discussion, see section 2.2.1 in Gustafsson (2010).
- *Efficient cross-validation.* Note that the matrix $\mathcal{H}(U_i)^T \mathcal{H}(U_i)$ in (20) denotes the information matrix, and can also be expressed as $\sum_k \mathcal{H}(U_{i,k})^T \mathcal{H}(U_{i,k})$, where k is a trial index and $U_{i,k}$ contains the data from one trial. This trick of combining sufficient statistics for the different datasets saves a lot of computations. For a more detailed discussion, see sections 2.2.3, 2.2.4 in Gustafsson (2010).
- *Introducing artifacts from discontinuities between trials.* The issue of introducing artifacts from discontinuities between trials is due to the boundary effects when the filter shifts out of the window. One solution is to insert zeros in the Hankel matrix used for solving the LS problem. A better alternative is to delete the rows in the Hankel matrix affected by these boundaries, which yields an LS estimate without boundary effects. In a similar way, one can remove discontinuities between trials in the concatenation case. For more details, see section 6.3 in Gustafsson et al. (2010).

Although both alternatives have been widely used as tools for studying selective attention and AAD, we shall here consider the first alternative. A basic reason for this is that the first alternative has received somewhat more attention in the literature due primarily to being implemented in the publicly available mTRF toolbox. It is also important to note that the second alternative is often less sensitive to the choice of the regularization parameter, and for which regularization can sometimes even be omitted if sufficient data is available (Biesmans et al., 2017).

4.1. Canonical Correlation Analysis

We start by evaluating the CCA model. The simple CCA model consists of the following steps:

- Design a multichannel representation of the input sound signal, e.g., cochlear or any other auditory model, time-frequency analysis with spectrogram, or Mel-frequency cepstral coefficients (MFCC) (Slaney, 1998).
- Demand two linear transformations with CCA. Efficient CCA-based decoding implementations are available in (1) COCOHA toolbox (https://cocoha.org/the-cocoha-matlab-toolbox/), (2) NoiseTools toolbox, (3) http://www.imt.liu.se/~magnus/cca/ and (4) http://www.yelab.net/software/CCA/. A particularly simple way of implementing CCA is available in MATLAB 's *canoncorr.m* function. This function takes Hankel matrices $\mathcal{H}(U_i)$ and $\mathcal{H}(Y)$ with time lags [defined as in (4)] as inputs and computes the filters $\boldsymbol{A}, \boldsymbol{B}_i$ and correlation coefficients (Krzanowski, 2000).

- Select the first (few) component(s) for each transformation such that the highest possible correlation between the datasets is retrieved.

4.1.1. Example 2 (Attention Deciphering With CCA)

In this example, we consider one (randomly selected) subject from the first database who attended to the speech on his left side U_1. The task is to determine whether CCA can be used to identify whether the attended speech is actually U_1.

4.1.1.1. Preprocessing

We followed the very simple preprocessing scheme described in the last sentence of §2.1 and in Alickovic et al. (2016).

4.1.1.2. Modeling

Following the approach to CCA proposed here, see Equation (23), the encoding and decoding filters covered time lags ranging from −250 ms to 0 ms prestimulus (see Alickovic et al., 2016) and 0 ms to 250 ms poststimulus (see O'Sullivan et al., 2015), respectively.

4.1.1.3. Classification

After projecting data onto a lower-dimensional space, a linear SVM is applied for binary classification: attended vs. ignored sound. We select the correlation coefficient values as the classifier's inputs. In this example, we selected the first 10 coefficients, thus classifying two times with a 10-D vector, once for the attended sound and once for the ignored sound. This corresponds to a 2-fold match-mismatch classification scheme suggested in de Cheveigné et al. (2018). In the case that the classifier implies attention on both sounds (attended and ignored), we consider such classification as incorrect. Next, we generate 10 random partitions, i.e., 10-fold cross-validation (CV), of data into training (27 minutes) and test (3 minutes) sets, and we report the average performances.

4.1.1.4. Results

The average classification accuracy is ∼ 98%. The total computational time for training and CV is ∼ 20 s.

4.1.1.5. Remarks

Note that this accuracy could be further improved with more training data or further preprocessing (e.g., removing eye blinks from EEG data). However, because we aim to establish real-time systems, we attempt to reduce the preprocessing and thereby increase the speed of the system at the expense of a lower accuracy rate.

As for any data-driven model design, the choice of the classifier's inputs is left to the user. Our choice is based primarily on the desire to show that CCA is a promising tool for auditory attention classification. In the following sections, we further discuss the significance of CCA by comparing the results of the methods discussed here applied on the two large datasets described in section 3.

4.2. Decoding With Dense Estimation

SR is the most prominent decoding technique, see Equation (11), that aims to reconstruct the stimuli from the measured neural responses. The standard approach to SR in the literature is to use l_2-regularized (dense) LR techniques. The recent work of Crosse et al. (2016) provides a comprehensive description of the Multivariate Temporal Response Function (mTRF) toolbox (https://sourceforge.net/projects/aespa/)—a MATLAB toolbox for computing (dense) filters A_j or B_i (depending on a mapping direction) by using LR techniques.

4.2.1. Example 3 (Attention Deciphering With Dense SR)

Here, we consider the same subject as in the previous example. The task is now to determine the efficiency of the dense SR in classifying the attended speech.

4.2.1.1. Preprocessing

Identical to Example (4.1.1).

4.2.1.2. Modeling

The decoder $\boldsymbol{A}$ covers time lags up to 250 ms poststimulus. To find the decoder $\boldsymbol{A}$, the model presented in Equation (11) is applied. One decoder is produced for each stream of sound i for each segment $s = 1, \ldots, 30$, resulting in 30 attended decoders.

4.2.1.3. Classification

Next, 29 of these decoders are combined by simply averaging $\boldsymbol{A}$ matrices to the matrix $\boldsymbol{A}_{avg}$ in the training phase - LOOCV (leave-one-out CV); then, $\boldsymbol{A}_{avg}$ is used to produce the estimate of the stimulus $\hat{U}_i$ for the fresh data, i.e., the remaining segment. The correlation coefficient c is then assessed between the actual n_u test stimuli U_i and the estimate $\hat{U}_i$, and the sound stream with the greatest c is identified as the attended source. This procedure is repeated 30 times.

4.2.1.4. Results

The average classification accuracy is ∼ 80%. Note the drop in accuracy from ∼ 98% (obtained with CCA) to ∼ 80% (with SR) for this particular subject. The total computational time for training and CV is ∼ 58 s.

4.3. Decoding With Sparse Estimation

In this section, we consider SR, but we use l_1 (sparse) regularization rather than l_2 (dense) regularization (which is widely used in auditory research) to quantify the sparsity effect on the auditory attention classification.

4.3.1. Example 4 (Attention Deciphering With Sparse SR)

Using the data from the same subject as in Examples (4.1.1–4.2.1), the task is to evaluate the performances of l_1-regularized (sparse) SR.

4.3.1.1. Preprocessing

4.3.1.1.1. Preprocessing/Modeling/Classification Identical to Example (4.1.1).

4.3.1.2. Preprocessing

4.3.1.2.1. Results The average classification accuracy is ∼ 80%. The total computational time for training and CV is ∼ 6 s. Note the drop in computational time from ∼ 58 s (obtained with dense SR) to ∼ 6 s (obtained with sparse SR) for this particular subject.

4.3.1.3. Preprocessing

4.3.1.3.1. Remarks Note the substantial reduction in the computational time when l_1 regularization, implemented with the ADMM, is used rather than conventional l_2 regularization in the SR method.

4.4. Encoding With Dense Estimation

Here, we consider encoding, where we go in the forward direction from the speech to EEG data. The standard approach to encoding found in the auditory literature is to solve the optimization problem (10) for each EEG channel $j = 1, \ldots, n_y$ separately, which means that we will have n_y neural predictions for each stimulus. Recall that one single reconstruction for each stimulus in the decoding approach discussed above makes it easier to compare the correlation coefficient values (CCVs). One way to classify the attended sound source by using the encoding approach is to take the sum of all CCVs, compare these sums, and classify the attended sound as the one with the highest sum of the CCVs (similar to the decoding). We refer to this approach as *dense LOOCV encoding*.

4.4.1. Example 5 (Attention Deciphering With Dense LOOCV Encoding)

Here, we consider the same subject as in the previous examples. The task is now to determine the efficiency of the suggested approach to dense encoding in classifying the attended speech.

4.4.1.1. Preprocessing

Identical to Example (4.1.1).

4.4.1.2. Modeling

The TRF $\boldsymbol{B}_i$ covers time lags from -250 ms to 0 ms prestimulus. To find the TRF $\boldsymbol{B}_i$, the model presented in Equation (10) is applied. One TRF is produced for each stream of sound i for each segment $s = 1, \ldots, 30$, resulting in 30 attended TRFs.

4.4.1.3. Classification

Next, 29 of these TRFs are combined by simply averaging $\boldsymbol{B}_i$ matrices to the matrix $\boldsymbol{B}_{i,avg}$ in the training phase - LOOCV (leave-one-out CV); then, $\boldsymbol{B}_{i,avg}$ is used to predict the neural response $\hat{Y}_i$ for the fresh data, i.e., the remaining segment. The summed CCV is then assessed between the actual Y and predicted $\hat{Y}_i$, and the sound stream with the larger CCV is identified as the attended source, i.e.,

$$\hat{i} = \arg\max_i CCV_i \tag{27}$$

This procedure is repeated 30 times.

4.4.1.4. Results

The average classification accuracy is ∼ 77%. The total computational time for training and CV is ∼ 2.5 s. However, the main limitation of the dense encoding is that it is very sensitive to the regularization parameter λ, which must be selected very carefully. We will return to this issue in section 4.7.

4.4.1.5. Remarks

Note the substantial reduction in the computational time with dense encoding compared to the dense decoding (SR) method.

4.5. Encoding With Sparse Estimation

Here, we consider encoding with ADMM-based sparse estimation. We report similar performance in terms of both the classification accuracy rate and computational time as observed for the encoding with dense estimation for the data taken from the same subject used in the previous examples. We refer to this approach as *sparse LOOCV encoding*.

4.5.1. Example 6 (Attention Deciphering With Sparse LOOCV Encoding)

Here, we consider the same subject as in the previous examples. The task is now to determine the efficiency of the suggested approach to sparse LOOCV encoding in classifying the attended speech.

4.5.1.1. Preprocessing, Modeling & Classification

As in Example (4.4.1).

4.5.1.2. Results

The average classification accuracy is ∼ 80%. The total computational time for training and CV is ∼ 1.5 s. Note that LOOCV encoding could be quite sensitive to λ.

4.6. Encoding From the System Identification Perspective

Here, we take a different approach to the common classification approaches found in the auditory literature, using tools from the system identification area (Ljung, 1998). In the present work, we refer to this approach as *adaptive encoding*.

4.6.1. Example 7 (Attention Deciphering With the SI Approach)

We consider the same data used in our previous examples. The task is now to use our classification model.

4.6.1.1. Preprocessing

Identical to Example (4.1.1).

4.6.1.2. Modeling

The TRF B_i covers time lags from −250 ms to 0 ms prestimulus. The attended and ignored TRFs $\boldsymbol{B}_1$ and $\boldsymbol{B}_2$ are computed for each segment, and the cost for both TRFs is evaluated for each segment as Lunner et al. (2018)

$$V_i(B_i) = \| Y - U_i\boldsymbol{B}_i\|_F^2 + \lambda\|\bar{\boldsymbol{B}}_i\|_1 \tag{28}$$

$$\text{subject to } \boldsymbol{B}_i = \bar{\boldsymbol{B}}_i \tag{29}$$

4.6.1.3. Classification

We compare the costs for each segment and determine which speech signal provides the smallest cost, i.e.,

$$\hat{i} = \arg\min_i V_i(\boldsymbol{B}_i) \tag{30}$$

If λ is known a priori, then this model is unsupervised and requires no training. However, this is rarely the case, and λ must be computed separately for each subject by using the subject's own training data.

4.6.1.4. Results

We use the first 9 min of data to compute the value of the regularization parameter λ and the remaining time to assess the performances of the models given in (28)-(30). The average classification accuracy is $\sim$ 95%.

4.6.1.5. Remarks

Although the classification accuracy of the adaptive encoding approach is similar to that obtained with CCA, note the substantial decrease in training time, from 27 to only 9 min.

4.7. Sensitivity of the Regularization Parameter

The previously discussed models have all been sensitive to a regularization parameter λ. Therefore, we need to solve the optimization problem (19) for different λ values to identify the λ value that optimizes the mapping such that the optimal λ value minimizes the mean squared error (MSE) and maximizes the correlation between the predicted (reconstructed) and actual waveform. One way to perform this optimization is to have the inner CV loop on the training data to tune λ value. In the inner CV loop, we can implement either LOOCV or K-fold CV in a similar way to the outer LOOCV, with the difference that we repeat the process for different λ values and select the λ that yields either the lowest MSE or the highest correlation (Pearson r) value. For the l_2 (dense) regularization, a parameter sweep is generally performed between 10^{-6} and 10^{8} (Wong et al., 2018). From our experience, a good choice for this type of regularization is to set λ to 10^{3}. For the l_1 (sparse) regularization, the parameter sweep is typically performed between $10^{-6}\lambda_{max}$ and $0.95\lambda_{max}$, where λ_{max} is a critical value above which the filter becomes zero-valued (Boyd et al., 2011). From our experience, a good choice for this type of regularization is to set λ to $10^{-1}\lambda_{max}$. A similar approach was adapted for the adaptive encoding, with the only difference that the inner CV loop was implemented on 9 min of data.

4.8. Classification Performance Comparison

In this section, we verify that the proposed linear models discussed in the present contribution can identify the sound source of the listener's interest. Two different datasets, the O'Sullivan and DTU datasets, were used to evaluate the performances of different models. Here the window length over which the correlation coefficients are estimated for each method is the same as in the corresponding examples above and the trial lengths are the same as the trial lengths mentioned in section 3.

4.8.1. O'Sullivan Dataset

Table 2 shows part of the assessed performances when the subjects were asked to attend to an identical sound source throughout the experiment. As shown in this table, CCA and adaptive encoding approaches resulted in the highest classification rates and the lowest computational times (see the previous examples). Moreover, note that the sparse estimation outperformed the dense estimation for both SR and LOOCV encoding. The accuracy rates for sparse SR were $\sim$ 5% higher, on average, when sparse (ADMM-based) estimation was used to determine the (decoder) filter coefficients. This was also the case when estimating the encoding filter coefficients. Furthermore, there was a significant reduction in computational time, as shown in **Table 3**. Although it might seem natural that l_2 regularization would be faster as l_1 regularization is iterative process, what makes l_1 regularization faster is the ADMM algorithm that converges quickly enough, within few iteration steps and does not include inverting large matrices.

TABLE 2 | Classification rates on the O'Sullivan dataset for the different classification approaches discussed in this contribution.

	Subject	Dense SR	Sparse SR	Dense LOOCV encoding	Sparse LOOCV encoding	Adaptive encoding	CCA
Attend Right	1	86.21	93.10	86.21	89.66	100	97.86
	2	86.67	90.00	70.00	70.00	95.45	98.32
	3	96.67	100.00	86.67	86.67	100.00	97.93
	4	90.00	90.00	80.00	76.67	86.36	98.33
	5	90.00	96.67	90.00	93.33	95.45	98.03
	6	70.00	86.67	60.00	70.00	100.00	97.83
	Avg	86.59	92.74	78.81	81.05	96.21	98.05
Attend Left	7	80.00	86.67	63.33	73.33	100.00	98.33
	8	93.33	90.00	76.67	80.00	95.45	97.70
	9	80.00	80.00	73.33	73.33	95.45	97.08
	10	80.00	90.00	73.33	76.67	81.82	96.90
	11	76.67	80.00	66.67	83.33	95.45	98.25
	12	100.00	100.00	83.33	86.67	100.00	98.32
	Avg	85.00	87.78	72.78	78.89	94.70	97.76
Total avg		85.80	90.26	75.80	79.97	95.45	97.91

As shown in **Tables 2**, **3**, the best-performing linear methods for this dataset in terms of both accuracy and computational time are *adaptive encoding* and *CCA*.

4.8.2. DTU Dataset

Table 4 shows part of the assessed performances when the subjects were asked to switch their attention throughout the experiment. As shown, CCA results in the highest classification rates. Moreover, note that for this dataset, the sparse estimation also outperformed the dense estimation for both SR and LOOCV encoding. However, the adaptive encoding did not result in a high classification accuracy rate for the "switching" data compared to CCA. One reason for this result might be that CCA, as a "bidirectional" approach, captures more of the EEG-audio (stimulus-response) data relationship than when going in only one (forward) direction. To summarize, all linear methods have a high potential to be fully utilized in the identification of the subject's sound source of interest in "attention-switching scenarios," with CCA demonstrating a high potential to also be used as an efficient AAD tool.

The O'Sullivan dataset is known to be biased in the sense that subjects either always maintain their attention on the left sound

TABLE 3 | Computational times on the O'Sullivan dataset for the different classification approaches discussed in this contribution.

	Subject	Dense SR	Sparse SR	Dense LOOCV encoding	Sparse LOOCV encoding	Adaptive encoding	CCA
Attend Right	1	46.69	5.21	2.06	1.99	1.96	23.34
	2	47.65	2.20	2.09	86.67	2.05	23.73
	3	49.44	2.20	2.38	76.67	2.38	20.75
	4	47.98	2.20	2.55	93.33	2.45	19.83
	5	47.95	2.20	2.09	70.00	2.00	19.58
	6	47.75	2.17	2.56	70.00	2.36	27.83
	Avg	47.91	5.43	2.17	2.28	2.20	22.51
Attend Left	7	47.61	5.26	2.16	2.20	2.15	20.32
	8	42.34	6.08	2.19	2.16	2.12	21.19
	9	43.03	5.28	2.15	2.08	2.06	19.53
	10	44.79	6.26	2.18	2.45	2.37	19.82
	11	43.30	5.28	2.19	2.14	2.10	19.91
	12	49.73	5.29	2.22	2.04	2.01	21.19
	Avg	45.13	5.57	2.18	2.18	2.08	20.33
Total avg		46.52	5.50	2.18	2.23	2.13	2.16

TABLE 4 | Classification rates on the DTU dataset for the different classification approaches discussed in this contribution.

Subject	Dense SR	Sparse SR	Dense LOOCV encoding	Sparse LOOCV encoding	Adaptive encoding	CCA
1	83.33	83.33	71.67	71.67	80.39	87.23
2	78.33	90.00	78.33	76.67	70.59	81.93
3	86.67	81.67	66.67	73.33	86.27	80.73
4	90.00	96.67	70.00	66.67	78.43	98.75
5	81.67	81.67	75.00	60.00	70.59	82.90
6	70.00	73.33	68.33	71.67	84.31	100.0
7	76.67	80.00	78.33	78.33	80.39	94.63
8	91.67	93.33	71.67	73.33	70.59	81.08
9	81.67	85.00	80.00	75.00	80.39	97.97
10	85.00	88.33	70.00	75.00	84.31	96.18
11	91.67	90.00	60.00	73.33	78.43	82.54
12	88.33	88.33	63.33	66.67	80.72	85.77
Total avg	83.75	85.97	71.11	72.22	78.33	89.14

source or always maintain their attention on the right sound source. The subject-dependent decoders then tend to perform much better than when they are trained on both left- and right-attended trials of the same subject. This effect was shown in Das et al. (2016). This partially explains why the performance on the DTU dataset is noticeably lower.

It is, however, important to keep in mind that although the tables above may indicate different performance among the methods, no comparative conclusions can be drawn from these tables, since the parameter settings may not be fully optimized or comparable. It is not the purpose of the paper to make that performance comparison, and rather just illustrate the different working principles. To objectively compare methods, one should use the same cross-validation, same window lengths to make a decision, and then properly optimize all parameters for each method.

5. CONCLUSIONS

In this work, we investigated the similarities and differences between different linear modeling philosophies: (1) the classical correlation-based approach (CCA), (2) encoding/decoding models based on dense estimation, and (3) (adaptive) encoding/decoding models based on sparse estimation. We described the complete signal processing chain, from sampled audio and EEG data, through preprocessing, to model estimation and evaluation. The necessary mathematical background was described, as well as MATLAB code for each step, with the intention that the reader should be able to both understand the mathematical foundations in the signal and systems areas and implement the methods. We illustrated the methods on both simulated data and an extract of patient data from two publicly available datasets, which have been previously examined in the literature. We have discussed the advantages and disadvantages of each method, and we have indicated their performance on the datasets. These examples are to be considered as inconclusive illustrations rather than a recommendation of which method is best in practice.

Furthermore, we presented a complete, step-by-step pipeline on how to approach identifying the attended sound source in a cocktail party environment from raw electrophysiological data.

AUTHOR CONTRIBUTIONS

All authors designed the study, discussed the results and implications, and wrote and commented the manuscript at all stages.

FUNDING

This work was supported in part by a research grant from the Oticon Foundation. This work was also supported by the EU H2020-ICT COCOHA (Cognitive Control of a Hearing Aid) grant agreement no: 644732. This work was further supported by Swedish Research Council (Vetenskapsrådet, VR 2017-06092 Mekanismer och behandling vid åldersrelaterad hörselnedsättning).

ACKNOWLEDGMENTS

We would like to thank Edmund Lalor, Alan Power, and Jens Hjortkjær for providing the data that were used to evaluate the proposed models. We also thank James O'Sullivan, Carina Graversen, Sergi Rotger Griful, and Alejandro Lopez Valdes for their technical assistance. Finally, the authors would like to thank Alain de Cheveigné for his input on CCA and for promoting CCA as an attention decoding tool at the 2014 Telluride Neuromorphic Cognition Engineering Workshop and within COCOHA.

REFERENCES

Ahveninen, J., Kopčo, N., and Jääskeläinen, I. P. (2014). Psychophysics and neuronal bases of sound localization in humans. *Hear. Res.* 307, 86–97. doi: 10.1016/j.heares.2013.07.008

Akbari, H., Khalighinejad, B., Herrero, J., Mehta, A., and Mesgarani, N. (2019). Towards reconstructing intelligible speech from the human auditory cortex. *Sci. Rep.* 9, 874.

Akram, S., Presacco, A., Simon, J. Z., Shamma, S. A., and Babadi, B. (2016). Robust decoding of selective auditory attention from MEG in a competing-speaker environment via state-space modeling. *Neuroimage* 124(Pt A), 906–917. doi: 10.1016/j.neuroimage.2015.09.048

Akram, S., Simon, J. Z., and Babadi, B. (2017). Dynamic estimation of the auditory temporal response function from MEG in competing-speaker environments. *IEEE Trans. Biomed. Eng.* 64, 1896–1905. doi: 10.1109/TBME.2016.2628884

Alain, C., and Bernstein, L. J. (2015). Auditory scene analysis. *Music Percept. Interdiscipl. J.* 33, 70–82. doi: 10.1525/mp.2015.33.1.70

Alickovic, E., Lunner, T., and Gustafsson, F. (2016). "A system identification approach to determining listening attention from EEG signals," in *2016 24th European Signal Processing Conference (EUSIPCO)* (Budapest), 31–35.

Alickovic, E., Lunner, T., and Gustafsson, F. (in rewiev) A sparse estimation approach to modeling listening attention from EEG signals. *PLoS ONE*.

Aroudi, A., Mirkovic, B., De Vos, M., and Doclo, S. (2016). "Auditory attention decoding with EEG recordings using noisy acoustic reference signals," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (Shanghai: IEEE), 694–698.

Babadi, B., Kalouptsidis, N., and Tarokh, V. (2010). Sparls: the sparse rls algorithm. *IEEE Trans. Signal Process.* 58, 4013–4025. doi: 10.1109/TSP.2010.2048103

Bednar, A., and Lalor, E. C. (2018). Neural tracking of auditory motion is reflected by delta phase and alpha power of EEG. *Neuroimage* 181, 683–691. doi: 10.1016/j.neuroimage.2018.07.054

Biesmans, W., Das, N., Francart, T., and Bertrand, A. (2017). Auditory-inspired speech envelope extraction methods for improved EEG-based auditory attention detection in a cocktail party scenario. *IEEE Trans Neural Syst Rehabil. Eng.* 25, 402–412. doi: 10.1109/TNSRE.2016.2571900

Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found Trends Mach. Learn.* 3, 1–122. doi: 10.1561/2200000016

Bregman, A. S. (1994). *Auditory Scene Analysis: The Perceptual Organization of Sound*. London: MIT Press.

Cherry, E. C. (1953). Some experiments on the recognition of speech, with one and with two ears. *J. Acous. Soc. Am.* 25, 975–979. doi: 10.1121/1.1907229

Ciccarelli, G., Nolan, M., Perricone, J., Calamia, P., Haro, S., O'Sullivan, J., et al. (2018). Comparison of two-talker attention decoding from EEG with nonlinear neural networks and linear methods. *bioRxiv*. doi: 10.1101/504522

Combettes, P. L., and Pesquet, J.-C. (2011). "Proximal splitting methods in signal processing," in *Fixed-Point Algorithms for Inverse Problems in Science and Engineering* (New York, NY: Springer), 185–212.

Crosse, M. J., Di Liberto, G. M., Bednar, A., and Lalor, E. C. (2016). The multivariate temporal response function (mTRF) toolbox: a matlab toolbox for relating neural signals to continuous stimuli. *Front. Hum. Neurosci.* 10:604. doi: 10.3389/fnhum.2016.00604

Das, N., Bertrand, A., and Francart, T. (2018). EEG-based auditory attention detection: boundary conditions for background noise and speaker positions. *J. Neural Eng.* 15:066017. doi: 10.1088/1741-2552/aae0a6

Das, N., Biesmans, W., Bertrand, A., and Francart, T. (2016). The effect of head-related filtering and ear-specific decoding bias on auditory attention detection. *J. Neural Eng.* 13:056014. doi: 10.1088/1741-2560/13/5/056014

Das, N., Van Eyndhoven, S., Francart, T., and Bertrand, A. (2017). "EEG-based attention-driven speech enhancement for noisy speech mixtures using n-fold multi-channel wiener filters," in *2017 25th European Signal Processing Conference (EUSIPCO)* (Kos: IEEE), 1660–1664.

de Cheveigné, A. (2010). Time-shift denoising source separation. *J. Neurosci. Methods* 189, 113–120. doi: 10.1016/j.jneumeth.2010.03.002

de Cheveigné, A. (2016). Sparse time artifact removal. *J. Neurosci. Methods* 262, 14–20. doi: 10.1016/j.jneumeth.2016.01.005

de Cheveigné, A., di Liberto, G. M., Arzounian, D., Wong, D., Hjortkjær, J., Asp Fuglsang, S., et al. (2019). Multiway canonical correlation analysis of brain data. *NeuroImage*. 186, 728–740. doi: 10.1016/j.neuroimage.2018.11.026

de Cheveigné, A., and Simon, J. Z. (2008a). Denoising based on spatial filtering. *J. Neurosci. Methods* 171, 331–339. doi: 10.1016/j.jneumeth.2008.03.015

de Cheveigné, A., and Simon, J. Z. (2008b). Sensor noise suppression. *J. Neurosci. Methods* 168, 195–202. doi: 10.1016/j.jneumeth.2007.09.012

de Cheveigné, A., Wong, D., Di Liberto, G., Hjortkjær, J., Slaney, M., and Lalor, E. (2018). Decoding the auditory brain with canonical component analysis. *Neuroimage* 172, 206–216. doi: 10.1016/j.neuroimage.2018.01.033

Deckers, L., Das, N., Hossein Ansari, A., Bertrand, A., and Francart, T. (2018). EEG-based detection of the attended speaker and the locus of auditory attention with convolutional neural networks. *bioRxiv*. doi: 10.1101/475673

Delorme, A., and Makeig, S. (2004). EEGLAB: an open source toolbox for analysis of single-trial EEG dynamics including independent component analysis. *J. Neurosci. Methods* 134, 9–21. doi: 10.1016/j.jneumeth.2003.10.009

Di Liberto, G. M., O'Sullivan, J. A., and Lalor, E. C. (2015). Low-frequency cortical entrainment to speech reflects phoneme-level processing. *Curr. Biol.* 25, 2457–2465. doi: 10.1016/j.cub.2015.08.030

Ding, N., and Simon, J. Z. (2012a). Emergence of neural encoding of auditory objects while listening to competing speakers. *Proc. Natl. Acad. Sci. U.S.A.* 109, 11854–11859. doi: 10.1073/pnas.1205381109

Ding, N., and Simon, J. Z. (2012b). Neural coding of continuous speech in auditory cortex during monaural and dichotic listening. *J. Neurophysiol.* 107, 78–89. doi: 10.1152/jn.00297.2011

Dmochowski, J. P., Ki, J. J., DeGuzman, P., Sajda, P., and Parra, L. C. (2017). Extracting multidimensional stimulus-response correlations using hybrid encoding-decoding of neural activity. *Neuroimage* 180(Pt A), 134–146. doi: 10.1016/j.neuroimage.2017.05.037

Ekin, B., Atlas, L., Mirbagheri, M., and Lee, A. K. C. (2016). "An alternative approach for auditory attention tracking using single-trial EEG," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (Shanghai), 729–733.

Etard, O., Kegler, M., Braiman, C., Forte, A. E., and Reichenbach, T. (2018). Real-time decoding of selective attention from the human auditory brainstem response to continuous speech. *bioRxiv*. doi: 10.1101/259853

Evans, S., McGettigan, C., Agnew, Z. K., Rosen, S., and Scott, S. K. (2016). Getting the cocktail party started: masking effects in speech perception. *J. Cogn. Neurosci.* 28, 483–500. doi: 10.1162/jocn_a_00913

Fiedler, L., Wöstmann, M., Graversen, C., Brandmeyer, A., Lunner, T., and Obleser, J. (2017). Single-channel in-ear-EEG detects the focus of auditory attention to concurrent tone streams and mixed speech. *J. Neural Eng.* 14:036020. doi: 10.1088/1741-2552/aa66dd

Fiedler, L., Wöstmann, M., Herbst, S. K., and Obleser, J. (2019). Late cortical tracking of ignored speech facilitates neural selectivity in acoustically challenging conditions. *Neuroimage* 186, 33–42. doi: 10.1016/j.neuroimage.2018.10.057

Fritz, J. B., Elhilali, M., David, S. V., and Shamma, S. A. (2007). Auditory attention - focusing the searchlight on sound. *Curr. Opin. Neurobiol.* 17, 437–455. doi: 10.1016/j.conb.2007.07.011

Fuglsang, S. A., Dau, T., and Hjortkjær, J. (2017). Noise-robust cortical tracking of attended speech in real-world acoustic scenes. *Neuroimage*. 156, 435–444. doi: 10.1016/j.neuroimage.2017.04.026

Gao, S., Wang, Y., Gao, X., and Hong, B. (2014). Visual and auditory brain-computer interfaces. *IEEE Trans. Biomed. Eng.* 61, 1436–1447. doi: 10.1109/TBME.2014.2300164

Gustafsson, F. (2010). *Statistical Sensor Fusion, 1st Edn.* Lund.

Gustafsson, F., Ljung, L., and Millnert, M. (2010). *Signal Processing*. Lund: Studentlitteratur.

Gutschalk, A., and Dykstra, A. R. (2014). Functional imaging of auditory scene analysis. *Hear. Res.* 307, 98–110. doi: 10.1016/j.heares.2013.08.003

Haghighi, M., Moghadamfalahi, M., Akcakaya, M., and Erdogmus, D. (2018). EEG-assisted modulation of sound sources in the auditory scene. *Biomed. Signal Process. Control* 39, 263–270. doi: 10.1016/j.bspc.2017.08.008

Hardoon, D. R., Szedmak, S., and Shawe-Taylor, J. (2004). Canonical correlation analysis: an overview with application to learning methods. *Neural Comput.* 16, 2639–2664. doi: 10.1162/0899766042321814

Haufe, S., Meinecke, F., Görgen, K., Dähne, S., Haynes, J.-D., Blankertz, B., et al. (2014). On the interpretation of weight vectors of linear models in multivariate neuroimaging. *Neuroimage* 87, 96–110. doi: 10.1016/j.neuroimage.2013.10.067

Hausfeld, L., Riecke, L., Valente, G., and Formisano, E. (2018). Cortical tracking of multiple streams outside the focus of attention in naturalistic auditory scenes. *Neuroimage* 181, 617–626. doi: 10.1016/j.neuroimage.2018.07.052

Henry, M. J., Herrmann, B., and Obleser, J. (2014). Entrained neural oscillations in multiple frequency bands comodulate behavior. *Proc. Natl. Acad. Sci. U.S.A.* 111, 14935–14940. doi: 10.1073/pnas.1408741111

Hjortkjær, J., Märcher-Rørsted, J., Fuglsang, S. A., and Dau, T. (2018). Cortical oscillations and entrainment in speech processing during working memory load. *Eur. J. Neurosci.* 1–11. doi: 10.1111/ejn.13855

Holdgraf, C. R., Rieger, J. W., Micheli, C., Martin, S., Knight, R. T., and Theunissen, F. E. (2017). Encoding and decoding models in cognitive electrophysiology. *Front. Syst. Neurosci.* 11:61. doi: 10.3389/fnsys.2017.00061

Jääskeläinen, I. P., and Ahveninen, J. (2014). Auditory-cortex short-term plasticity induced by selective attention. *Neural Plastic.* 2014:216731. doi: 10.1155/2014/216731

Kalashnikova, M., Peter, V., Di Liberto, G. M., Lalor, E. C., and Burnham, D. (2018). Infant-directed speech facilitates seven-month-old infants cortical tracking of speech. *Sci. Rep.* 8, 1–8. doi: 10.1038/s41598-018-32150-6

Kaya, E. M. and Elhilali, M. (2017). Modelling auditory attention. *Philos. Trans. R. Soc. Lond. B Biol. Sci.* 372:20160101. doi: 10.1098/rstb.2016.0101

Khong, A., Jiangnan, L., Thomas, K. P., and Vinod, A. P. (2014). "BCI based multi-player 3-D game control using EEG for enhancing attention and memory," in *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)* (San Diego, CA), 1847–1852.

Krzanowski, W. (2000). *Principles of Multivariate Analysis*, Vol. 23. Oxford: Oxford University Press .

Lesenfants, D., Vanthornhout, J., Verschueren, E., Decruy, L., and Francart, T. (2018). Predicting individual speech intelligibility from the neural tracking of acoustic- and phonetic-level speech representations. *bioRxiv*. doi: 10.1101/471367

Li, Q., and Wu, J. (2009). "Multisensory interactions of audiovisual stimuli presented at different locations in auditory-attention tasks: A event-related potential (ERP) study," in *2009 International Conference on Mechatronics and Automation* (Changchun), 146–151.

Ljung, L. (1998). *System Identification.* Upper Saddle River, NJ: Springer.

Lunner, T. (2015). *Hearing Device with External Electrode.* US Patent 8,971,558.

Lunner, T., and Gustafsson, F. (2013). *Hearing Device With Brainwave Dependent Audio Processing.* US Patent App. 14/048,883.

Lunner, T., Gustafsson, F., Graversen, C., and Alickovic, E. (2018). *Hearing Assistance System Comprising an EEG-Recording and Analysis System.* US Patent App. 15/645,606.

Mesgarani, N., and Chang, E. F. (2012). Selective cortical representation of attended speaker in multi-talker speech perception. *Nature* 485, 233–236. doi: 10.1038/nature11020

Middlebrooks, J. C. (2017). "Spatial stream segregation," in *The Auditory System at the Cocktail Party*, eds J. C. Middlebrooks, J. Z. Simon, A. N. Popper, and R. R. Fay (Springer), 137–168.

Miran, S., Akram, S., Sheikhattar, A., Simon, J. Z., Zhang, T., and Babadi, B. (2018). Real-time tracking of selective auditory attention from M/EEG: a bayesian filtering approach. *Front. Neurosci.* 12:262. doi: 10.3389/fnins.2018.00262

Mirkovic, B., Debener, S., Jaeger, M., and Vos, M. D. (2015). Decoding the attended speech stream with multi-channel EEG: implications for online, daily-life applications. *J. Neural Eng.* 12:046007. doi: 10.1088/1741-2560/12/4/046007

Narayanan, A. M., and Bertrand, A. (2018). "The effect of miniaturization and galvanic separation of EEG sensor nodes in an auditory attention detection task," in *40th International Conference of the IEEE EMBS* (Honolulu, HI).

Obleser, J., and Weisz, N. (2011). Suppressed alpha oscillations predict intelligibility of speech and its acoustic details. *Cereb. Cortex* 22, 2466–2477. doi: 10.1093/cercor/bhr325

Oostenveld, R., Fries, P., Maris, E., and Schoffelen, J. M. (2011). Fieldtrip: open source software for advanced analysis of MEG, EEG, and invasive electrophysiological data. *Comput. Intell. Neurosci.* 2011:156869. doi: 10.1155/2011/156869

O'Sullivan, J., Chen, Z., Herrero, J., McKhann, G. M., Sheth, S. A., Mehta, A. D., et al. (2017). Neural decoding of attentional selection in multi-speaker environments without access to clean sources. *J. Neural Eng.* 14:056001. doi: 10.1088/1741-2552/aa7ab4

O'Sullivan, J. A., Power, A. J., Mesgarani, N., Rajaram, S., Foxe, J. J., Shinn-Cunningham, B. G., et al. (2015). Attentional selection in a cocktail party environment can be decoded from single-trial EEG. *Cereb. Cortex* 25, 1697–1706. doi: 10.1093/cercor/bht355

Pasley, B. N., David, S. V., Mesgarani, N., Flinker, A., Shamma, S. A., Crone, N. E., et al. (2012). Reconstructing speech from human auditory cortex. *PLoS Biol.* 10:e1001251. doi: 10.1371/journal.pbio.1001251

Power, A. J., Foxe, J. J., Forde, E.-J., Reilly, R. B., and Lalor, E. C. (2012). At what time is the cocktail party? A late locus of selective attention to natural speech. *Eur. J. Neurosci.* 35, 1497–1503. doi: 10.1111/j.1460-9568.2012.08060.x

Presacco, A., Simon, J. Z., and Anderson, S. (2016). Evidence of degraded representation of speech in noise, in the aging midbrain and cortex. *J. Neurophysiol.* 116, 2346–2355. doi: 10.1152/jn.00372.2016

Ramirez, C., Kreinovich, V., and Argaez, M. (2013). Why l_1 is a good approximation to l_0: a geometric explanation. *J. Uncertain Syst.* 7, 203–207.

Rao, N., Nowak, R., Cox, C., and Rogers, T. (2016). Classification with the sparse group lasso. *IEEE Trans. Signal Process.* 64, 448–463. doi: 10.1109/TSP.2015.2488586

Ru, P. (2001). *Multiscale Multirate Spectro-Temporal Auditory Model*. Ph.D. thesis, University of Maryland College Park.

Schäfer, P. J., Corona-Strauss, F. I., Hannemann, R., Hillyard, S. A., and Strauss, D. J. (2018). Testing the limits of the stimulus reconstruction approach: auditory attention decoding in a four-speaker free field environment. *Trends Hear.* 22, 1–12. doi: 10.1177/2331216518816600

Scott, S. K., and McGettigan, C. (2013). The neural processing of masked speech. *Hear. Res.* 303, 58–66. doi: 10.1016/j.heares.2013.05.001

Sepulcre, Y., Trigano, T., and Ritov, Y. (2013). Sparse regression algorithm for activity estimation in γ spectrometry. *IEEE Trans. Signal Process.* 61, 4347–4359. doi: 10.1109/TSP.2013.2264811

Simon, J. Z. (2017). "Human auditory neuroscience and the cocktail party problem," in *The Auditory System at the Cocktail Party*, eds J. C. Middlebrooks, J. Z. Simon, A. N. Popper, and R. R. Fay (Springer), 169–197.

Slaney, M. (1998). *Auditory Toolbox.* Technical Report. Interval Research Corporation.

Snyder, J., Gregg, M., Weintraub, D., and Alain, C. (2012). Attention, awareness, and the perception of auditory scenes. *Front. Psychol.* 3:15. doi: 10.3389/fpsyg.2012.00015

Somers, B., Verschueren, E., and Francart, T. (2019). Neural tracking of the speech envelope in cochlear implant users. *J. Neural Eng.* 16:016003. doi: 10.1088/1741-2552/aae6b9

Sun, L., Ji, S., and Ye, J. (2011). Canonical correlation analysis for multilabel classification: a least-squares formulation, extensions, and analysis. *IEEE Trans. Patt. Anal. Mach. Intell.* 33, 194–200.

Taillez, T., Kollmeier, B., and Meyer, B. T. (2017). Machine learning for decoding listeners attention from electroencephalography evoked by continuous speech. *Eur. J. Neurosci.* 1–8. doi: 10.1111/ejn.13790

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *J. R. Stat. Soc. B Methodol.* 58, 267–288. doi: 10.1111/j.2517-6161.1996.tb02080.x

Tsiami, A., Katsamanis, A., Maragos, P., and Vatakis, A. (2016). "Towards a behaviorally-validated computational audiovisual saliency model," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (Honolulu, HI), 2847–2851.

Van Eyndhoven, S., Francart, T., and Bertrand, A. (2017). EEG-informed attended speaker extraction from recorded speech mixtures with application in neuro-steered hearing prostheses. *IEEE Trans. Biomed. Eng.* 64, 1045–1056. doi: 10.1109/TBME.2016.2587382

Vanthornhout, J., Decruy, L., Wouters, J., Simon, J. Z., and Francart, T. (2018). Speech intelligibility predicted from neural entrainment of the speech envelope. *J. Assoc. Res. Otolaryngol.* 19, 181–191. doi: 10.1007/s10162-018-0654-z

Verschueren, E., Vanthornhout, J., and Francart, T. (2018). Semantic context enhances neural envelope tracking. *bioRxiv*. doi: 10.1101/421727

Wang, D., and Brown, G. J. (2006). *Computational Auditory Scene Analysis: Principles, Algorithms, and Applications*. New York, NY: Wiley-IEEE Press.

Wang, D., Kjems, U., Pedersen, M. S., Boldt, J. B., and Lunner, T. (2009). Speech intelligibility in background noise with ideal binary time-frequency masking. *J. Acous. Soc. Am.* 125, 2336–2347. doi: 10.1121/1.3083233

Watkins, D. S. (2004). *Fundamentals of Matrix Computations*, Vol. 64. New York, NY: John Wiley & Sons.

Weisz, N., Hartmann, T., Müller, N., and Obleser, J. (2011). Alpha rhythms in audition: cognitive and clinical perspectives. *Front. Psychol.* 2:73. doi: 10.3389/fpsyg.2011.00073

Wong, D. D., Fuglsang, S. A. A., Hjortkjær, J., Ceolini, E., Slaney, M., and de Cheveigné, A. (2018). A comparison of regularization methods in forward and backward models for auditory attention decoding. *Front. Neurosci.* 12:531. doi: 10.3389/fnins.2018.00531

Yang, X., Wang, K., and Shamma, S. A. (1992). Auditory representations of acoustic signals. *IEEE Trans. Inform. Theor.* 38, 824–839. doi: 10.1109/18.119739

Yuan, M., and Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *J. R. Stat. Soc. B* 68, 49–67. doi: 10.1111/j.1467-9868.2005.00532.x

Zink, R., Proesmans, S., Bertrand, A., Van Huffel, S., and De Vos, M. (2017). Online detection of auditory attention with mobile EEG: closing the loop with neurofeedback. *bioRxiv*. doi: 10.1101/218727

Zion Golumbic, E. M., Ding, N., Bickel, S., Lakatos, P., Schevon, C. A., McKhann, G. M., et al. (2013). Mechanisms underlying selective neuronal tracking of attended speech at a cocktail party. *Neuron* 77, 980–991. doi: 10.1016/j.neuron.2012.12.037

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

A. APPENDIX: EEG DATA IMPORT

A.1. Importing EEG Data With EEGLab

The key steps are as follows:

- Downloading the EEGLab toolbox.
- Starting MATLAB and adding the path.
- Loading the EEG data with the *pop_biosig* function.
- Excluding all non-scalp channels and reference to average all scalp channels as: *EEG = pop_select(EEG,'nochannel', 'channel names'); EEG = pop_reref(EEG, []);*
- Segmenting data correctly based on the trigger information with the *pop_epoch* function.
- Additionally, mean baseline value from each epoch can be removed with the *pop_rmbase* function.
- Saving the .mat file

A.2. Importing EEG Data With FieldTrip

The key steps are as follows:

- Downloading the FieldTrip toolbox.
- Starting MATLAB and adding the path.
- Using the *ft_defaults* function to configure default variable and path settings.
- Reading the EEG data with a *ft_read_data* function to a structure file and adding the needed values to the fields in the structure from the header with the function *ft_read_header*.
- Reading the event information if possible with *ft_read_event*.
- Segmenting the data correctly based on the relevant event(s).
- Selecting the scalp channels.
- Removing the mean and normalizing the data.

frontiers in Neuroscience

METHODS
published: 07 August 2018
doi: 10.3389/fnins.2018.00531

A Comparison of Regularization Methods in Forward and Backward Models for Auditory Attention Decoding

Daniel D. E. Wong[1,2]*, *Søren A. Fuglsang*[3], *Jens Hjortkjær*[3,4], *Enea Ceolini*[5], *Malcolm Slaney*[6] and *Alain de Cheveigné*[1,2,7]

[1] Laboratoire des Systèmes Perceptifs, CNRS, UMR 8248, Paris, France, [2] Département d'Études Cognitives, École Normale Supérieure, PSL Research University, Paris, France, [3] Department of Electrical Engineering, Danmarks Tekniske Universitet, Kongens Lyngby, Denmark, [4] Danish Research Centre for Magnetic Resonance, Copenhagen University Hospital Hvidovre, Hvidovre, Denmark, [5] Institute of Neuroinformatics, University of Zürich, Zurich, Switzerland, [6] AI Machine Perception, Google, Mountain View, CA, United States, [7] Ear Institute, University College London, London, United Kingdom

The decoding of selective auditory attention from noninvasive electroencephalogram (EEG) data is of interest in brain computer interface and auditory perception research. The current state-of-the-art approaches for decoding the attentional selection of listeners are based on linear mappings between features of sound streams and EEG responses (forward model), or vice versa (backward model). It has been shown that when the envelope of attended speech and EEG responses are used to derive such mapping functions, the model estimates can be used to discriminate between attended and unattended talkers. However, the predictive/reconstructive performance of the models is dependent on how the model parameters are estimated. There exist a number of model estimation methods that have been published, along with a variety of datasets. It is currently unclear if any of these methods perform better than others, as they have not yet been compared side by side on a single standardized dataset in a controlled fashion. Here, we present a comparative study of the ability of different estimation methods to classify attended speakers from multi-channel EEG data. The performance of the model estimation methods is evaluated using different performance metrics on a set of labeled EEG data from 18 subjects listening to mixtures of two speech streams. We find that when forward models predict the EEG from the attended audio, regularized models do not improve regression or classification accuracies. When backward models decode the attended speech from the EEG, regularization provides higher regression and classification accuracies.

Keywords: temporal response function, speech decoding, electroencephalography, selective auditory attention, attention decoding

OPEN ACCESS

Edited by:
Einat Liebenthal,
Brigham and Women's Hospital,
Harvard Medical School,
United States

Reviewed by:
Michael J. Crosse,
Albert Einstein College of Medicine,
United States
Nai Ding,
Zhejiang University, China
Scott A. Beardsley,
Marquette University, United States

****Correspondence:***
Daniel D. E. Wong
ddewong@gmail.com

Specialty section:
This article was submitted to Auditory Cognitive Neuroscience, a section of the journal Frontiers in Neuroscience

Received: *16 March 2018*
Accepted: *16 July 2018*
Published: *07 August 2018*

Citation:
Wong DDE, Fuglsang SA, Hjortkjær J, Ceolini E, Slaney M and de Cheveigné A (2018) A Comparison of Regularization Methods in Forward and Backward Models for Auditory Attention Decoding. Front. Neurosci. 12:531. doi: 10.3389/fnins.2018.00531

1. INTRODUCTION

A fundamental goal of auditory neuroscience is to understand the mapping between auditory stimuli and the cortical responses they elicit. In magneto/electro-encephalography (M/EEG) studies, this mapping has predominantly been measured by examining the average cortical evoked response potential (ERP) to a succession of repeated short stimuli. More recently, these methods have been extended to continuous stimuli such as speech by using linear system-response models, broadly termed "temporal response functions" (TRFs), that are estimated using system-identification methods. The TRF is a stimulus-response model that characterizes how a unit impulse in an input feature corresponds to a change in the M/EEG data. TRFs can be used to generate continuous predictions about M/EEG responses as opposed to characterizing the response (ERP) to repetitions of the same stimuli. Importantly, it has been demonstrated that the stimulus-response models can be extracted both from EEG responses to artificial sound stimuli (Lalor et al., 2006, 2009; Power et al., 2011) but also from EEG responses to naturalistic speech (Lalor and Foxe, 2010). A number of studies have considered mappings between the slowly varying temporal envelope of a speech sound signal ($<$10 Hz) and the corresponding filtered M/EEG response (Lalor and Foxe, 2010; Ding and Simon, 2012a,b, 2013, 2014). However, TRFs are not just limited to the broadband envelope, but can also be obtained with the speech spectrogram (Ding and Simon, 2012a,b), phonemes (Di Liberto et al., 2015), or semantic features (Broderick et al., 2018). This has opened new avenues of research into cortical responses to speech, advancing the field beyond examining responses to repeated isolated segments of speech.

TRF methods have proven particularly apt for studying how the cortical processing of speech features are modulated by selective auditory attention. A number of studies have considered multi-talker "cocktail party" scenarios, where a listener attends to one speech source and ignores others. It has been demonstrated that both attended and unattended acoustic features can be linearly mapped to the cortical response (Ding and Simon, 2012a,b; Power et al., 2012; Zion Golumbic et al., 2013; Puvvada and Simon, 2017).

Conversely, the same linear model, which maps speech features to the cortical response (forward direction), can be adapted to provide a linear mapping from the cortical response to the speech features (backward direction) (Bialek et al., 1991; Mesgarani et al., 2009; Ding and Simon, 2012a,b; Mesgarani and Chang, 2012; Mirkovic et al., 2015; O'Sullivan et al., 2015; Fuglsang et al., 2017; Van Eyndhoven et al., 2017). The mapping from acoustic features to cortical responses is typically referred to as a forward model (or TRF), whereas the mapping from cortical responses to acoustic features is referred to as a backward model (Haufe et al., 2014). The quality of model fit reflects the degree to which cortical activity is driven by stimulation. In a cocktail party scenario, the quality of fit between each of the speech streams and the cortical activity can be used to infer which speech stream is being attended. Differences in the accuracy of forward/backward model-derived estimates between the attended and unattended speech signal can be used to predict or "decode" to whom a listener is attending based on unaveraged M/EEG data. Single-trial measures of auditory selective attention in turn suggests BCI applications, for instance, for cognitively-steered hearing aids (Das et al., 2016; O'Sullivan et al., 2017; Van Eyndhoven et al., 2017; Zink et al., 2017).

The ability of forward/backward stimulus-response models to generalize to new data is generally limited by the need to estimate a relatively large number of parameters based on noisy single-trial M/EEG responses. Like many aspects of machine learning, this necessitates regularization techniques that constrain the model coefficients to prevent overfitting (Crosse et al., 2016a; Holdgraf et al., 2017). A number of methods for regularizing the forward/backward stimulus-response models have been presented in various studies (Goutte et al., 2000; Theunissen et al., 2000, 2001; Machens et al., 2004; David et al., 2007; Thorson et al., 2015). Each of these methods attempt to address the challenge of having sufficient data to compute a reliable stimulus-response mapping function. To reduce the data requirement, regularization can be applied in the form of a smoothness and/or sparsity constraint.

To date, little work has been done to compare these methods against each other. A meta-analysis would be difficult as many variables, such as subjects, stimuli and data processing are different between each study. The present paper uses a standardized publicly available dataset[1] (Fuglsang et al., 2018), based on the attended-vs.-unattended talker discrimination task, as well as preprocessing and evaluation procedures to compare these algorithms. In addition, the present paper examines the relationship between different evaluation metrics to highlight their similarities and differences. The methods for computing forward/backward stimulus-response models have been implemented in the publicly available Telluride Decoding Toolbox[2].

2. MATERIALS AND METHODS

Temporal response functions can be used to predict the EEG response to a multi-talker stimulus from the attended speech envelope or, alternatively, the equation can be adapted to reconstruct the attended speech envelope from the EEG response. The first case is denoted as a "forward model" (as it maps from speech features to neural data) and the second as a "backward model" (as it maps from neural data back to speech features) (Haufe et al., 2014).

2.1. Stimulus-Response Models

The linear stimulus-response models below described below map a matrix **X** (stimulus features for a forward model, EEG for a backward model) to a matrix **Y** (EEG channels for a forward model, stimulus features for a backward model):

$$\hat{\mathbf{Y}} = \mathbf{X}\mathbf{W}, \tag{1}$$

where $\mathbf{X} = [x_{t,(f,c)}]$ is a multichannel data matrix (channels indexed by c), augmented to include time-lagged versions of the data (lags indexed by f), and $\hat{Y} = [y_t]$ is the model estimate in the form of a vector indexed by time t. Time lags, limited to a range such as -500 to + 500 ms, allow the model to handle delays and convolutional mismatch between **X** and **Y**. Dimensions c and f are combined when performing matrix multiplications.

In the following subsections we introduce different approaches to estimating the linear model parameters, **W**.

[1] http://doi.org/10.5281/zenodo.1199011
[2] http://www.ine-web.org/software/decoding

Each method uses different regularization techniques to optimize the generalizability of the mapping functions.

2.1.1. Ordinary Least Squares (OLS)

The cost function that is minimized when solving the regression model is:

$$\mathcal{L}(\mathbf{W}) = (\mathbf{Y} - \mathbf{XW})^T(\mathbf{Y} - \mathbf{XW}). \tag{2}$$

The filter coefficients of this model can be estimated via ordinary least squares:

$$\mathbf{W} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{Y}, \tag{3}$$

where $\mathbf{X}^T\mathbf{X}$ is the estimated autocovariance matrix and $\mathbf{X}^T\mathbf{Y}$ is the estimated cross-covariance matrix. The ordinary least-squares solution was here estimated using the Cholesky decomposition method, via the *mldivide* routine in Matlab. One advantage of the OLS estimator is that it has no additional hyperparameters that must be optimized. However, in practice the OLS estimator is often outperformed by the regularized solutions described in the following subsections. This is often the case when the regressor, $\mathbf{X}$, is high-dimensional and has a poorly estimated covariance matrix given limited amounts of training data, or contains auto-correlations and/or cross-channel correlations resulting in a low rank matrix. In other words, the inverse problem is ill-posed. Such is the case when using non-stochastic data for $\mathbf{X}$, such as speech or EEG data.

If $\mathbf{X}$ were white and standardized, the autocovariance matrix would be a multiple of the identity matrix, and the OLS and regularized approaches reduce to a straight-forward cross-correlation, also known as reverse correlation (Ringach and Shapley, 2004).

2.1.2. Ridge

Ridge regression minimizes the residual sum of squares, but adds an $L2$ constraint on the regression coefficients (Machens et al., 2003; Crosse et al., 2015; Di Liberto et al., 2015; Crosse et al., 2016b; Holdgraf et al., 2016; O'Sullivan et al., 2017; Broderick et al., 2018). An $L2$ constraint smooths the regression weights by penalizing the square of the weights in $\mathbf{W}$ with a regularization constant λ for the Ridge regression cost function:

$$\mathcal{L}(\mathbf{W})_\lambda = (\mathbf{Y} - \mathbf{XW})^T(\mathbf{Y} - \mathbf{XW}) + \lambda\mathbf{W}^T\mathbf{W} \tag{4}$$

(Hastie et al., 2001; Machens et al., 2004). Ridge regression corresponds to imposing a Gaussian prior on the filter coefficients (Wu et al., 2006). The Ridge solution is:

$$\mathbf{W} = \left(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}\right)^{-1}\mathbf{X}^T\mathbf{Y}, \tag{5}$$

where λ is the regularization parameter that controls the amount of parameter shrinking.

2.1.3. Low-Rank Approximation (LRA)

The LRA-based regression relies on a low-rank approximation of the covariance matrix, $\mathbf{X}^T\mathbf{X}$. This is achieved by employing a singular value decomposition (SVD) of $\mathbf{X}^T\mathbf{X}$:

$$\mathbf{X}^T\mathbf{X} = \mathbf{USV}^T, \tag{6}$$

where $\mathbf{U}$ and $\mathbf{V}$ are orthonormal matrices that contain respectively the left and right singular vectors, and where $\mathbf{S}$ is a diagonal matrix, $\mathbf{S} = \text{diag}(s_1, s_2, ..s_d)$ with sorted diagonal entries. Since $\mathbf{X}^T\mathbf{X}$ is a positive semidefinite matrix we have $\mathbf{U} = \mathbf{V}$. LRA uses a rank-K approximation of $\mathbf{X}^T\mathbf{X}$ by only retaining the first $1 \leq K \leq d$ diagonal elements of $\mathbf{S}$. The cost function is:

$$\begin{aligned}\mathcal{L}(\mathbf{W})_K &= (\mathbf{Y} - \mathbf{XW})^T(\mathbf{Y} - \mathbf{XW}) \\ &\quad -\mathbf{W}^T\mathbf{V}_{K+1...d}\mathbf{S}_{K+1...d,K+1...d}\mathbf{V}^T_{K+1...d}\mathbf{W},\end{aligned} \tag{7}$$

where $\mathbf{V}_{K+1...d}$ are the $K+1...d$ columns of $\mathbf{V}$ and $\mathbf{S}_{K+1...d,K+1...d}$ is the square matrix formed by taking the $K+1...d$ rows and columns of $\mathbf{S}$. By forming $\hat{\mathbf{S}}^{-1} = \text{diag}(1/s_1, 1/s_2, ..., 1/s_K, 0..0, 0, 0)$, the regression coefficients can be estimated from:

$$\mathbf{W} = \left(\mathbf{U}\hat{\mathbf{S}}^{-1}\mathbf{V}^T\right)\mathbf{X}^T\mathbf{Y}. \tag{8}$$

The number of diagonal elements, K, to retain are typically chosen such that a diagonal element is retained if the sum of the eigenvalues to be kept cover a fraction λ of the overall sum, or $0 < \frac{\sum_{i=1}^{K} s_i}{\sum_{i=1}^{d} s_i} < \lambda \leq 1$. Note that the regularization parameter, λ, here is analogous to λ for Ridge Regression, but that the values are not comparable between the two. LRA is the term used in systems identification (Marconato et al., 2014), however, this type of regression has also been referred to as normalized reverse correlation (NRC) in auditory neuroscience literature (Theunissen et al., 2000, 2001; David et al., 2004, 2007; Mesgarani et al., 2009; Mesgarani and Chang, 2012).

2.1.4. Shrinkage

Shrinkage (Friedman, 1989; Blankertz et al., 2011) is a method used for biasing the covariance matrix by flattening its eigenvalue spectrum with some tuning parameter, λ. In the context of regression, the Shrinkage cost function is:

$$\mathcal{L}(\mathbf{W})_\lambda = (\mathbf{Y} - \mathbf{XW})^T(\mathbf{Y} - \mathbf{XW}) + \lambda\mathbf{W}^T(\nu\mathbf{I} - \mathbf{X}^T\mathbf{X})\mathbf{W}, \tag{9}$$

where ν is here defined as the average eigenvalue trace of the covariance matrix $\left(\mathbf{X}^T\mathbf{X}\right)$. The solution for the cost function is:

$$\mathbf{W} = \left((1-\lambda)\mathbf{X}^T\mathbf{X} + \lambda\nu\mathbf{I}\right)^{-1}\mathbf{X}^T\mathbf{Y}. \tag{10}$$

When $\lambda = 0$, it becomes the standard ordinary least squares solution. When $\lambda = 1$, the covariance estimator becomes diagonal (i.e., it becomes spherical), reducing the Shrinkage equation to a cross-correlation (Blankertz et al., 2011).

These regularization schemes are related. Whereas Ridge Regression and Shrinkage both penalize extreme eigenvalues in a smooth way, LRA discards eigenvalues. Ridge and Shrinkage in other words flatten out the eigenvalue trace. Ridge shifts it up, and Shrinkage shrinks it toward an average value ν (Blankertz et al., 2011), whereas LRA cuts if off.

2.1.5. Tikhonov

The scheme that we shall refer to as *Tikhonov regularization*, is a first-derivative type of Tikhonov regularization (Tikhonov, 1963) that takes advantage of the fact that there is usually a strong correlation between adjacent columns of $\mathbf{X}$ when $\mathbf{X}$ includes time shifts, because of the strong serial correlation of the stimulus envelope (for the forward model) or the filtered EEG (for the backward model). In other words, Tikhonov regularization imposes *temporal smoothness* on the model. Tikhonov regularization achieves temporal smoothness by putting a constraint in the derivative of the filter coefficients (Goutte et al., 2000; Lalor et al., 2006; Lalor and Foxe, 2010; Crosse et al., 2015, 2016a). Here we focus on first order derivatives of the filter coefficients and assume that the first derivatives can be approximated by $\frac{\partial w_i}{\partial i} \approx (w_{i+1} - w_i)$ for any neighboring filter pairs w_{i+1} and w_i. This type of regularization is more generally referred to as 1st order Tikhonov regularization as it attempts to constrain the first derivative of the filter via central difference approximations. This gives the cost function:

$$\mathcal{L}(\mathbf{W})_\lambda = (\mathbf{Y} - \mathbf{XW})^T (\mathbf{Y} - \mathbf{XW}) + \lambda \sum_i (w_i - w_{i+1})^2 . \quad (11)$$

Tikhonov regularized model filters can, under this approximation, be implemented as:

$$\mathbf{W} = \left(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{M}\right)^{-1} \mathbf{X}^T\mathbf{Y}, \quad (12)$$

where

$$\mathbf{M} = \begin{bmatrix} 1 & -1 & 0 & 0 & \cdots & 0 \\ -1 & 2 & -1 & 0 & \cdots & 0 \\ 0 & -1 & 2 & -1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix}.$$

Note that cross-channel leakage can occur whenever the regressor, $\mathbf{X}$, reflects data recorded from multiple channels, as is the case with the backward model. This means that filter endpoints can be affected by neighboring channels as a result of the off-diagonal elements in the $\mathbf{M}$ matrix. Due to the potential for cross-channel leakage, Tikhonov has been primarily used for the forward modeling case (Crosse et al., 2016a). Despite the potential problems associated with cross-channel leakage, we also report results obtained with Tikhonov regularization for the backward model for completeness.

2.1.6. Elastic Net

Whereas the aforementioned regularization techniques often show improvements over the ordinary least regression in terms of generalizability, they tend to preserve all regressors in the models. This can e.g., result in nonzero filter weights assigned to irrelevant features. Lasso regression attempts to overcome this issue by putting an L1-constraint on the regression coefficients (Tibshirani, 1996). This serves to drive unnecessary coefficients in the model toward zero. Lasso has been found to perform well in many scenarios, although it was empirically demonstrated that it is outperformed by Ridge regression in nonsparse scenarios with highly correlated predictors (Tibshirani, 1996; Zou and Hastie, 2005). In such scenarios, *Elastic Net* regression (Zou and Hastie, 2005) has been found to improve the predictive power of Lasso by combining Lasso with the grouping effect of Ridge regression. The Elastic Net has two hyperparameters: α controlling the balance between L1 (lasso) and L2 (Ridge) penalties, and λ controlling the overall penalty strength. For the purpose of this paper, we use a readily available algorithm, GLMNET (Qian et al., 2013), for efficiently computing the Elastic Net problem. This is a coordinate descent algorithm for solving the following problem:

$$\underset{\mathbf{W}}{\operatorname{argmin}} \frac{1}{2N} \|\mathbf{Y} - \mathbf{XW}\|^2 + \lambda \left[(1-\alpha) \|\mathbf{W}\|^2/2 + \alpha\|\mathbf{W}\|\right]. \quad (13)$$

We used GLMNET for computing the Elastic Net solution for $\alpha = 0.25$, $\alpha = 0.50$, $\alpha = 0.75$ and $\alpha = 1.00$. We will henceforth refer the last case as the Lasso solution. The GLMNET has previously been used to estimate spectro-temporal receptive models (e.g., Willmore et al., 2016).

2.2. Evaluating Performance

2.2.1. Characterizing Model Fit

While the objective function of linear models is minimizing the mean-squared-error, the goodness of fit is typically analyzed in terms of Pearson's correlation between estimated and actual values for interpretability. The term *regression accuracy* will henceforth be used to characterize the goodness of fit for models trained and evaluated on attended audio features ($r_{attended}$). For forward models, regression accuracies were measured by the Pearson's correlation between the actual EEG and the EEG predicted by the attended envelope over the test folds. This was done separately for each EEG channel. Similarly, for backward models, regression accuracies were measured by the correlation between the attended envelope and its EEG-based reconstruction. The regression accuracies were computed on test folds, using the nested cross-validation scheme described in section 2.2.3. This procedure ensures that the test data is not used during any part of the training process, including hyperparameter tuning. The regression accuracies were averaged over all test folds. Other metrics for assessing the predictive/reconstructive performance of the models have been previously proposed (Schoppe et al., 2016). However, for simplicity and to be consistent with previous studies (Ding and Simon, 2012a,b; O'Sullivan et al., 2015), this paper characterizes the goodness of the fit using Pearson's correlation coefficients.

In the forward case, the response at multiple EEG channels is predicted by the model. Rather than using multiple correlation coefficients to characterize the regression accuracy in this case, we chose to take the average of the correlation coefficients between the predicted channels and the actual EEG data as a validation score. We used the same metric over the test set to characterize the fit of the model. In the backward case, characterizing the fit is straightforward as the model predicts a single audio envelope that can be correlated with the attended audio envelope.

2.2.2. Decoding Selective Auditory Attention

Performance was also evaluated on a classification task based on the forward/backward stimulus-response model. The task of the classifier was to decide, on the basis of the recorded EEG and the two simultaneous speech streams presented to the listener (see section 2.4), to which stream the subject was attending. The classifier had to make this decision on the basis of a segment of test data, the duration of which was varied as a parameter (1, 3, 5, 7, 10, 15, 20, and 30 s), which will be referred to as the decoding segment length. This duration includes the kernel length of the forward/backward model (500 ms). The position of this segment of data was stepped in 1s increments throughout the evaluated data.

As described further in section 2.2.3, a nested cross-validation loop was used to tune the forward/backward stimulus-response model regularization parameter (where applicable) on training/validation data and test the trained classifier on unseen test data.

The classification relied on correlation coefficients between EEG and the attended speech, and between the EEG and the unattended speech. These correlation coefficients were computed over the aforementioned restricted time window. These coefficients were used to classify whether the subject was attending to one stream or the other. For a backward model, classification hinged merely on which correlation coefficient was largest (stream A or stream B). Performance of this classifier was evaluated on the test set. For a forward model, the situation is more complex because there is one model per EEG channel. For each of the 66 channels a pair of correlation coefficients was calculated (one each for unattended and attended streams), and this set of pairs was used to train a support vector machine (SVM) classifier with a linear kernel and a soft margin constant of 1. SVM classifiers were trained on the correlation coefficient features over the validation set that was used for hyperparameter tuning. The SVM classifier performance was finally evaluated on data from the held out test fold.

The classifier score was averaged over all test folds. In every case, the classifier trained over the entire training/validation set was tested on a short interval of data, the duration of which was varied as a parameter, as explained above. An illustration of this classification task is shown in **Figure 1**.

Classification performance was characterized for different decoding segment durations using the raw classification score, receiver operating characteristic (ROC) curve, and information transfer rate (ITR). The raw classification score measured what proportion of trials were classified correctly. It should be noted that in measuring classification performance, the two classes were balanced. The ROC curve characterizes the true-positive and false-positive rates for decoding segment trials where the classifier discrimination function lies above a given threshold, as the threshold is varied. The classifier decision function is the distance between the classified point and the decision boundary, with the sign indicating the class label. In the case of an SVM

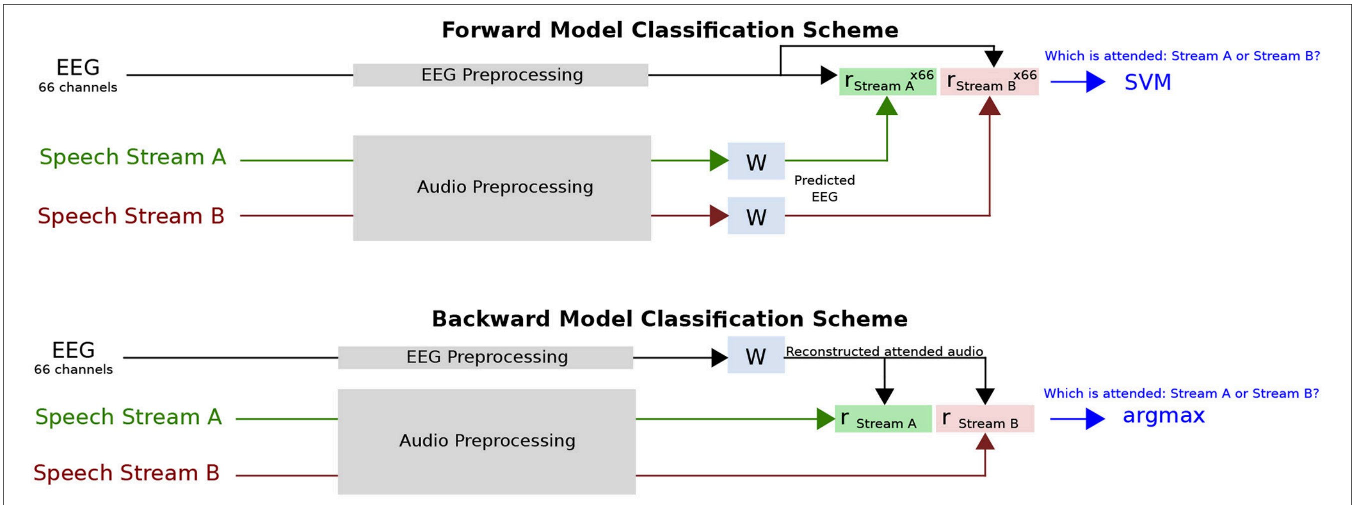


FIGURE 1 | Diagram of classification task. For the forward model, 66 EEG channels are predicted from the speech stream A and B envelopes using the same linear mapping function, W. After correlation with the 66 channel EEG data, this results in 66 correlation coefficients for each speech stream, which are used as features for the SVM to distinguish the attended talker. For the backward model, a single attended audio envelope channel is estimated from the EEG data using the linear mapping function, W. After correlation with the speech stream A and B envelopes, a single correlation coefficient for each speech stream is obtained. Classification of the attended talker is performed by determining the larger coefficient.

classifier for the forward model, the decision function is a weighted sum of the input features (correlations), plus a bias term. In the case of the argmax function for the backward model, the decision function is the difference of the correlations between the reconstructed attended audio and the two speech streams. Thresholding the classifier discrimination function throughout the range of values it yields in a dataset affects the number of correctly and incorrectly classified trials (above threshold) out of the total number of correctly and incorrectly classified trials, which are the true and false positive rates, respectively.

The ITR metric corresponds to the number of classifications that can be reliably made by the system in a given amount of time. The dependency of ITR on decoding segment length is a tradeoff between two effects. On one hand, longer decoding segments allow more reliable decisions. On the other, short durations allow a larger number of independent decisions. There is thus an optimal decoding segment duration. A number of metrics to compute the ITR have been proposed. The most common is the Wolpaw ITR (Wolpaw and Ramoser, 1998), which is calculated in bits per minute as:

$$ITR_W = V\left[\log_2 N + P\log_2 P + (1-P)\log_2 \frac{1-P}{N-1}\right], \quad (14)$$

where V is the speed in trials per minute, N is the number of classes, and P is the classifier accuracy. We also report the Nykopp ITR, which assumes that a classification decision does not need to be made on every trial (Nykopp, 2001). This can be done by first calculating the confusion matrix p for classifier outputs where the classifier decision function magnitude exceeds a given threshold. Typically the larger the classifier decision function magnitude, the more accurate the classifier prediction. As such, raising the threshold on the decision function magnitude results in more accurate classifications at the expense of foregoing a classification decision on more trials. To obtain the Nykopp information transfer rate, the threshold on the classifier decision function magnitude is adjusted to maximize:

$$\begin{aligned} ITR_N \;=\; V\Bigg[\max_{p(x)} \sum_{i=1}^{N}\sum_{j=1}^{M} p(w_i)p(\hat{w}_j|w_i)\log_2 p(\hat{w}_j|w_i) \\ -\sum_{j=1}^{M} p(\hat{w}_j)\log_2 p(\hat{w}_j)\Bigg], \quad (15)\end{aligned}$$

where $p(w_i)$ is the probability of the actual class being class i, $p(\hat{w}_j|w_i)$ is the probability of the predicted class being class j given the actual class being class i, and $p(\hat{w}_j)$ is the probability of the predicted class being class j. It is $p(\hat{w}_j|w_i)$ and $p(\hat{w}_j)$ that are affected by decision function magnitude thresholding as this limits the number of trials on which a classification decision is made.

2.2.3. Cross-Validation Procedure

The forward/backward stimulus-response models used in sections 2.2.1 and 2.2.2 were all trained and tested using cross-validation with a 10-fold testing procedure involving nested cross-validation loops. This procedure ensures that the test data used to evaluate the forward/backward model is not used during any part of the training process. During this cross-validation procedure the models were characterized under an N-fold testing framework where the data was divided into 10-folds. In this outer cross-validation loop, one fold was held out for testing (i.e., characterizing model fit and classifying the attended stream), while data from the remaining 9-folds were used to compute the forward/backward models using an inner cross-validation loop. This inner cross-validation loop was used to tune the hyperparameters. The stimulus-response models were in all cases fit to the envelope of the attended sound streams during the training phase. The regularization parameter was swept through a range of values to evaluate its effect on the correlation coefficient between the model prediction/reconstruction and the actual measured data for each inner cross-validation fold. For Ridge and Lasso regularization schemes that allowed a regularization parameter between zero and infinity, a parameter sweep was performed between 10^{-6} and 10^{8} in 54 logarithmically-spaced steps. This was done using the following formula:

$$\lambda_n = \lambda_0 \times 1.848^n, n \in [0, 53], \quad (16)$$

where $\lambda_0 \equiv 10^{-6}$. For LRA, Elastic Net, and Shrinkage schemes, where the regularization parameter range was between 0 and 1, a parameter sweep was performed between 10^{-6} and 1 using a log-sigmoid transfer function that compresses the values between 0 and 1 using the following iterative formula:

$$\lambda_{n+1} = \text{logsig}(\ln(\lambda_n) - \ln(1-\lambda_n) + 0.475), n \in [0, 40]. \quad (17)$$

The hyperparameter value that yielded the maximum correlation between the model prediction/reconstruction and actual measured data, averaged across all inner cross-validation folds, was used to evaluate the test set. Using this hyperparameter value, the weights of the models generated for each inner cross-validation fold were then averaged to generate an overall cross-validated model that could then be applied to the test set. It should be noted that for each test fold, the hyperparameter value was selected independently.

2.3. Implementation

The implementations of the forward/backward stimulus-response model algorithms used here are distributed as part of the Telluride Decoding Toolbox[2], specifically in the FindTRF.m function of that toolbox. Data preprocessing, model training, and evaluation were implemented with the COCOHA Matlab Toolbox[3].

2.4. Stimuli

A previous report gives a detailed description of the stimuli and data collection procedure (Fuglsang et al., 2017). This dataset is available online (Fuglsang et al., 2018). In brief, a set of speech stimuli were recorded by one male and one female

[3] http://doi.org/10.5281/zenodo.1198430

professional Danish speakers speaking different fictional stories. These recordings were performed in an anechoic chamber at the Technical University of Denmark (DTU). The recording sampling rate was 48 kHz. Each recording was divided into 50-s long segments for a total of 65 segments.

2.5. Experimental Procedure

The 50-s long speech segments were used to generate auditory scenes comprising a male and a female simultaneously speaking in anechoic or reverberant rooms. The two concurrent speech streams were normalized to have similar root-mean square values. The speech stimuli were delivered to the subjects via ER-2 insert earphones (Etymotic Research). The speech mixtures were presented binaurally to the listeners, with the two speech streams lateralized at respectively -60° and $+60^\circ$ along the azimuth direction and a source-receiver distance of 2.4 m. This was achieved using non-individualized head-related impulse responses that were simulated using the room acoustic modeling software, Odeon (version 13.02). Each subject undertook sixty trials in which they were presented the 50 s-long speech mixtures. Before each trial, the subjects were cued to listen selectively to one speech stream and ignore the other. After each trial, the subjects were asked a comprehension question related to the content of the attended speech stream. The position of the target streams as well as the gender of the target speaker were randomized across trials. Moreover, the type of acoustic room condition (either anechoic, mildly reverberant or highly reverberant) were pseudo-randomized over trials. In the analysis, data recorded from all acoustic conditions were pooled together. The reasons for doing this were twofold. Firstly, it provides sufficient data for the stimulus-response analysis. This is particularly important as insufficient data in worst case can lead to poorer model estimates (Mirkovic et al., 2016). Secondly, by using this approach we get a better idea of how well the models will generalize to different experimental conditions. This is an important practical aspect, as it gives a better estimate of how well a classifier will perform in different listening conditions (rather than just focusing on training on anechoic data and evaluating on anechoic data).

2.6. Data Collection

Electroencephalography (EEG) data were recorded from 19 subjects in an electrically shielded room while they were listening to the stimuli described above. Data from one subject were excluded from the analysis due to missing data from several trials. The data were recorded using a Biosemi Active 2 system, with a sampling rate of 512 Hz. Sixty-four channel EEG data (10/20-system) were recorded from the scalp. Six additional electrodes were used for recording the EEG at the mastoids, and vertical and horizontal electrooculogram (V and H-EOG). Approximately 1 h of EEG data was recorded per subject. This study was carried out in accordance with the recommendations of "Fundamental and applied hearing research in people with and without hearing difficulties, Videnskabsetiske komitee." The protocol was approved by the Science Ethics Committee for the Capital Region of Denmark. All subjects gave written informed consent in accordance with the Declaration of Helsinki.

2.7. Data Preprocessing

2.7.1. EEG Data

50 Hz line noise and harmonics in the EEG data were filtered out by convolution with a $\frac{512}{50}$ sample square window (the non-integer window size was implemented by interpolation) (de Cheveigné and Arzounian, 2017). The EEG data was then downsampled to 64 Hz using a resampling method based on the Fast Fourier Transform (FFT). To downsample, this method reduces the size of the FFT of the signal by truncating high frequency components. An inverse FFT is then used to restore the signal to the time domain. A 1st order detrend was performed on the EEG data to minimize filter startup artifacts. EEG data were highpassed at 0.1 Hz using a 4th order forward-pass Butterworth filter. The group delay was less than 2 samples above 1 Hz.

The joint decorrelation framework (de Cheveigné and Parra, 2014) was employed to remove eye artifacts in an automated fashion. Let $\mathbf{X} = [x_{tj}]$ be a matrix that contains EEG data from each electrode, j, for each time sample t. In this implementation, a conservative eye artifact time-point detection was first performed by computing a Z-score on 1–30 Hz bandpassed VEOG and HEOG bipolar channels and marking time samples where the absolute Z-score on either channel exceeded 4. This is similar to the eyeblink detection method implemented in the FieldTrip EEG processing toolbox (Oostenveld et al., 2011). This resulted in a subset of time samples, A, indexing the temporal locations of each EOG artifact. An artifact covariance matrix $\mathbf{R}_A = \mathbf{X}_A^T\mathbf{X}_A$ was then computed from the EEG (and EOG) data, $\mathbf{X}_A = [x_{aj}]$, at the artifact time samples $a \in A$. After using principal component analysis to whiten $\mathbf{R}_A$ and $\mathbf{R}$, the generalized eigenvalue problem was then solved for $\mathbf{R}_A\mathbf{v} = \lambda\mathbf{R}\mathbf{v}$, where $\mathbf{R} = \mathbf{X}^T\mathbf{X}$ is the covariance matrix for the entire EEG dataset. The resulting eigenvectors $\mathbf{V}$, sorted by eigenvalue, explain the maximum difference in variance between the artifact and data covariance matrices. Components corresponding to eigenvalues $> 80\%$ of the maximum eigenvalue were regressed out of the data. In practice, this 80% threshold is a conservative one, typically resulting in the removal of one or two components. Lastly, the EOG channels were removed from the data, which was then referenced to a common average over all channels.

For the forward/backward model analysis, the EEG was bandpassed between 1–9 Hz using a windowed sync type I linear-phase finite-impulse response (FIR) filter, shifted by its group delay to produce a zero-phase (Widmann et al., 2015) with a conservatively chosen order of 128 in order to minimize ringing effects. This frequency range was selected as it has been shown that cortical responses time-lock to speech envelopes in this range (O'Sullivan et al., 2015). As part of the cross-validation procedure, individual EEG channels were finally centered and standardized (Z-normalized) across the time dimension using the individual channel mean and standard deviation of the training data. A kernel length of 0.5 s (33 samples) was used when computing the forward/backward models.

2.7.2. Audio Features

The forward/backward stimulus-response model estimation methods used for attention decoding attempt to characterize a relationship between features of attended speech streams and EEG activity. We calculated temporal envelope representations from each of the clean speech streams (i.e., without reverberation). We did not try to derive them from the reverberant or mixed audio data, as explored elsewhere (Aroudi and Doclo, 2017; Fuglsang et al., 2017). In trials with reverberant speech mixtures, we used envelope representations of the underlying clean signals to estimate the models. To derive the envelope representations, we passed monaural versions of both attended and unattended speech streams through a 31-band gammatone filterbank with a frequency range of 80–8,000 Hz (Patterson et al., 1987). The envelope of each filterbank output was calculated via the analytic signal obtained with the Hilbert transform, raised to the power of 0.3. This rectification and compression step was intended to partially mimic that which is seen in the human auditory system (Plack et al., 2008). The audio envelope was then calculated by summing the rectified and compressed filterbank outputs across channels. The audio envelope data was subsequently downsampled to the same sampling frequency as the EEG (64 Hz) using an FFT-based resampling method. The EEG and envelopes were then temporally aligned using start-trigger events recorded in the EEG. The envelopes were subsequently lowpassed at 9 Hz. As part of the cross-validation procedure, audio envelopes were finally centered and standardized (Z-normalized) across the time dimension using the mean and standard deviation of the attended speech envelope in the training data.

2.8. Statistical Analysis

All statistical analyses were calculated using MATLAB. Repeated-measures analysis of variance (ANOVA) tests were used to assess differences between the regression accuracies (section 2.2.1) and classification performances section 2.2.2 obtained with the different forward/backward model estimation methods. Regression accuracies and classification performances for individual subjects were averaged across folds prior to statistical comparison.

Given the non-Gaussian distribution of regression accuracies (range -1 to 1) and classification performance metrics (range 0 to 1), Fisher Z-transforms and arcsine transforms were applied to these measures, respectively, prior to statistical tests and correlations.

3. RESULTS

The forward/backward stimulus-response model estimation methods introduced in section 2 were used to decode attended speech envelopes from low-frequency EEG activity. The following sections analyze results with metrics of (1) regression accuracy, (2) classification accuracy, (3) receiver operating characteristic (ROC), and (4) information transfer rate (ITR). Results are shown for each of the regularization schemes, for both forward and backward models. For each regularization scheme, the regularization parameter(s) are tuned to maximize regression accuracy. These parameter values are then used for all regression and classification comparisons. Regression accuracy compares different regularization schemes in predicting/reconstructing test data using the optimal regularization parameter. Classification accuracy uses the regression accuracy values to classify the attended/unattended talker and compares the different regularization schemes in performing this task. The ROC curve visualizes the relationship between the true and false-positive rates for different classifier discrimination function thresholds. Lastly, the ITR describes the impact of decoding segment length on the bit-rate, for different points on the ROC curve.

3.1. Regularization Parameter Tuning

The forward/backward model estimation methods, except for the OLS method, use regularization techniques to prevent overfitting and therefore require a selection of the appropriate tuning parameters. **Figure A1** in Supplementary Material shows the correlation coefficient between estimated (validation set) data and the actual target data (*regression accuracy*) over a range of regularization parameters. In general, there is a broad region where validation regression accuracy is flat, which peaks before quickly falling off with increasing λ. It is also apparent that the regression accuracies obtained with backward models generally are higher than those obtained with forward models.

Figure A2 in Supplementary Material shows regression accuracies for forward/backward models with Elastic Net penalties. Unlike the other linear models investigated in the present study the Elastic Net has two hyperparameters. The α parameter adjusts the balance between $L1$ and $L2$ penalties. Similar to the other regularization schemes, for each value of α, there is a broad range of λ values that give good correlation performance.

3.2. Regression Accuracy

For each regression method (and each value of α for Elastic Net), the forward/backward stimulus-response model was estimated and the optimum lambda estimated on the training/validation set. This optimal model was then applied to the test set, and the regression accuracy was compared between regression methods. This is shown in **Figure 2**. One might expect that the averaging of prediction-response correlations across channels for the forward model may have resulted in lower regression accuracies compared to the backward model. This was demonstrating using a t-test between the forward and backward models, over all regularization schemes and subjects [$\Delta = 0.083$, $T_{(107)} = 17.8$, $p = 1.1 \times 10^{-33}$]. However, when using maximum correlation across channels, instead of the average, for the forward model, there was still a significant difference [$\Delta = 0.045$, $T_{(107)} = 9.8$, $p = 9.4 \times 10^{-17}$].

For forward models, a repeated measures ANOVA with regularization method as the factor found no significant effect of regularization method on the average of correlation coefficients, even when using the average of the correlation coefficients of the 5 channels with the largest correlation coefficients for each subject. For the backward models, a similar repeated measures ANOVA, found a significant effect of regularization method on regression accuracy [$F_{(5,85)} = 78.0, p < 1.0 \times 10^{-16}$]. Tikhonov regularization yielded a regression accuracy that was significantly greater than each of the other schemes, using a Bonferonni

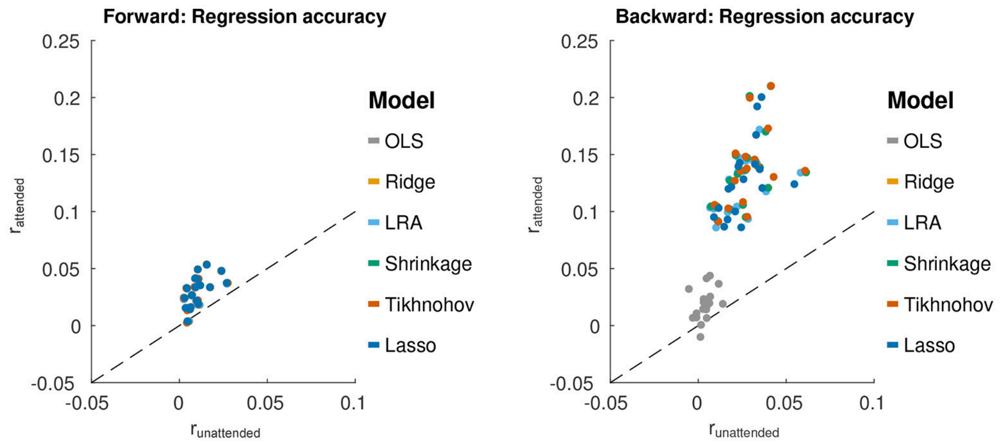


FIGURE 2 | Test set regression accuracies (r_{attend}) for each forward/backward model estimation method plotted against $r_{unattend}$. **(Left)** Results from the forward modeling approach. Points for each regularization scheme are close to each other, and thus appear to fall on top of each other. **(Right)** Results from the backward modeling approach. For each scheme (represented by a color), each point represents average data from one subject. The black line shows $r_{attend} = r_{unattend}$.

correction to account for the family-wise error rate ($p < 0.045$). This is contrary to the expectation that Ridge regression would outperform Tikhonov for the backward model due to the inter-channel leakage introduced by the Tikhonov kernel. Moreover, OLS had a regression accuracy that was significantly smaller than the other schemes (with Bonferonni correction, $p < 1.3 \times 10^{-10}$). This highlights the importance of regularization for the backward models.

For Elastic Net regularization, α values was characterized at 0.25, 0.5, 0.75, and 1 (Lasso) to sample different degrees of sparsity/smoothness. The value $\alpha = 0$ (Ridge) was not sampled due to sub-optimal solver performance near this point. A repeated measures ANOVA analysis with factors of α and subject, using optimal λ values, showed no significant effect of α for forward models. This means that adjusting the model sparsity had no significant effect on the regression accuracy. However, a significant effect of α was found for backward models [$F_{(3,51)} = 12.4, p = 3.3 \times 10^{-6}$]. A *post-hoc* paired *t*-test with a Bonferonni correction revealed that the best regression accuracy was obtained with $\alpha = 0.25$ ($p = 6.2 \times 10^{-4}$). It was, however, noted that the average difference between regression accuracies for $\alpha = 0.25$ and $\alpha = 1$ was only 8×10^{-4}.

To obtain an estimate of the significance of the regression accuracies presented in **Figure 2**, we randomized the phase of the audio data passed to the forward models, and the phase of the EEG data passed to the backward models. The goal was to provide an estimate of the correlation noise floor for the models. The models were those trained on unaltered data using each of the regularization schemes. Randomizations were performed 100 times per subject to yield an estimate of the noise floor regression accuracies. The regression accuracies were computed the same way as before. A two-sample Kolmogorov-Smirnov test conducted pairwise showed that, within subjects, the distribution of noise floor correlations were not significantly different between regularization schemes, or channels in the case of the forward model. The within-subject distributions were thus combined, and a two-sample Kolmogorov-Smirnov test was performed pairwise between subjects. No significant difference in distributions was found between subjects. As such, all distributions were combined. The 95% confidence interval of the noise floor correlations was [-0.001, 0.001] for the forward model and [-0.032, 0.032] for the backward model.

3.3. Classification Accuracy

We further sought to investigate how the different forward/backward models perform in terms of discriminating between attended and unattended speech on a limited segment of data. The duration of the segment was varied as a parameter (1, 3, 5, 7, 10, 15, 20, and 30 s). This was characterized on held-out test data for each TRF method, using the λ value that yielded the maximum regression accuracy in the validation data. The results from this analysis are shown in **Figure 3**. A 2-way repeated measures ANOVA with factors of regularization scheme and model (forward or backward), based on 30 s decoding segment lengths, found a main significant difference between backward and forward models [$F_{(1,17)} = 17.3, p = 6.5 \times 10^{-4}$], with a significant interaction with the effect of regularization scheme [$F_{(5,85)} = 208.9, p < 1.0 \times 10^{-16}$]. A *post-hoc* paired *t*-test showed that backward model performs better than the forward model for all regularization schemes excluding the case where ordinary least squares (OLS) was applied [$T_{(17)} = 9.35, p = 4.2 \times 10^{-8}$]. For OLS, the forward model outperformed the backward model [$T_{(17)} = 7.32, p = 1.2 \times 10^{-6}$].

The interaction of the effect of regularization scheme on the classification accuracy of forward and backward models was investigated. A repeated measures ANOVA with factors of regularization scheme, applied only to the forward TRF classification accuracy scores, found no significant effect of regularization scheme on classification accuracy. This is consistent with the lack of significant differences being detected in regression accuracies for different forward model

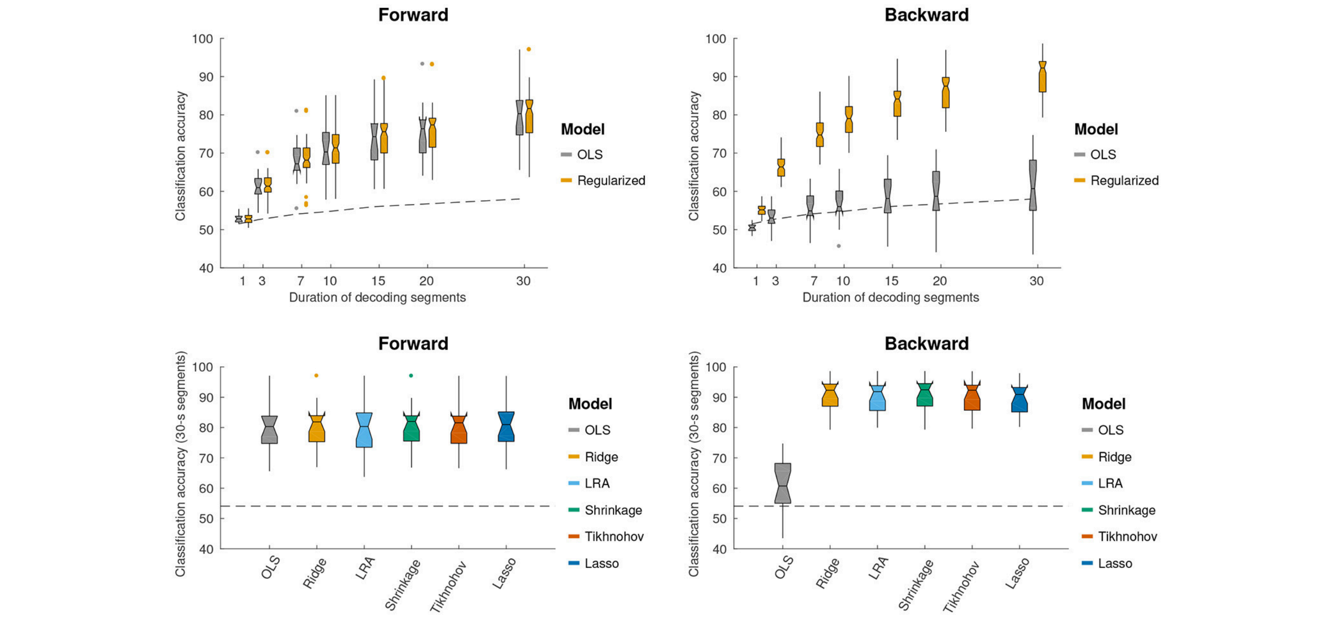


FIGURE 3 | Using different forward/backward models to decode selective auditory attention from multi-channel EEG data. Classification performance is shown for different decoding segment lengths (1, 3, 7, 10, 15, 20, 30 s). (**Top**, left and right) Show the classification performance for forward and backward models respectively. Performance is shown for the OLS scheme and an average across regularized schemes. Regularized schemes were averaged to concisely illustrate the higher classification accuracy obtained by these schemes compared to OLS for the backward model, but not the forward model. (**Bottom**, left and right) Show the classification performance for 30 s long decoding segments. The different regularization schemes are shown in different colors (see legend). Notched boxplots show median, and first and third quartiles. Whiskers show 1.5 × IQR. Dots indicate outliers. The dashed line shows the above-chance significance threshold at $p = 0.05$.

regularization schemes, even when limiting the number of channels to 5 with the highest regression accuracies. In this case, the SVM classifier can be viewed as a data-driven approach to select channels that are most relevant to attention classification. For the backward models, however, a significant effect of regularization scheme on classification accuracy was found [$F_{(5,85)} = 229.4$, $p < 1.0 \times 10^{-16}$]. A *post-hoc* paired *t*-test analysis with a Bonferonni correction revealed that the classification accuracy for the OLS scheme was significantly worse than each of the others ($\bar{\Delta} = -29.1$, $p < 7.9 \times 10^{-10}$). Lasso performed significantly worse than each of the remaining schemes ($\bar{\Delta} = -1.2$, $p < 0.040$). In short, regularized backward schemes outperform OLS by a relatively large margin, as seen in **Figure 3**.

For Elastic Net regularization, a repeated measures ANOVA with factors of α and subject did not find any significant effect of α on classification accuracy for forward or backward models.

In summary, for the forward model there was no difference between schemes (regularization and OLS), and for the backward model there was no difference between Ridge, Tikhonov, Shrinkage and LRA, but all regression methods were better than OLS.

3.3.1. Relation to Regression Accuracy

The discrimination between attended and unattended speech streams from EEG data is done in two stages: the computation of regression accuracies, followed by classification. We sought to investigate how the classification accuracies obtained with each model relate to the test set regression accuracies. A plot of this relationship is shown in **Figure 4**.

For forward models, the average correlation between regression accuracy and classification performance across decoding segments and over all regularization schemes is 0.69 [$T_{(108)} = 9.83$, $p = 2.2 \times 10^{-16}$]. For backward models, the correlation between the regression accuracy and classification performance is 0.89 [$T_{(108)} = 22.4$, $p < 1.0 \times 10^{-16}$]. This suggests that classification performance varies with regression accuracy. However, as was previously described for the backward models, while Tikhonov regularization achieved a significantly higher regression accuracy compared to all other methods, it did not achieve a significantly higher classification performance compared to Shrinkage, Ridge Regression or LRA. To explain this, we examined the classification feature in terms of the difference between class means ($\bar{r}_{attend} - \bar{r}_{unattend}$) and the within-class standard deviation ($\sqrt{0.5(\sigma^2_{r_{attend}} + \sigma^2_{r_{unattend}})}$). Both of these terms affect the separability between classes.

For backward models, Tikhonov regularization had a significantly larger difference between class means compared to Ridge Regression and Shrinkage [Tikhonov>Ridge: $T_{(17)} = 2.62$, $p = 0.018$], [Tikhonov>Shrinkage: $T_{(17)} = 2.59$, $p = 0.019$]. At the same time, the between-class standard deviation was also significantly larger for Tikhonov regularization [Tikhonov>$F_{(100,100)} = 2.37$, $p = 1.2 \times 10^{-5}$], [Tikhonov>Shrinkage: $F_{(100,100)} = 2.37$, 1.4×10^{-5}]. This

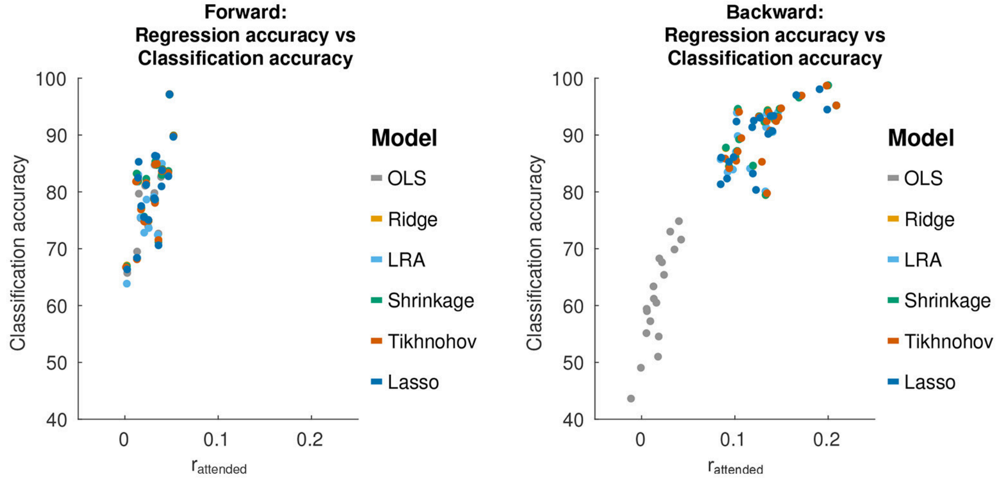


FIGURE 4 | Relationship between regression accuracy and classification accuracy, using 30 s decoding segment lengths.

suggests that while Tikhonov regularization yields a better regression accuracy (correlation coefficient), this is offset by an increased variance in the regression accuracy computed over short decoding segments, nullifying any potential gains in classification performance.

3.4. Receiver Operating Characteristic

The receiver operating characteristic (ROC) curve, shown in **Figure 5**, shows the relationship between the true-positive rate and false-positive rate for decoding segment trials where the classifier discrimination function lies above a given threshold, as the threshold is varied. The classification accuracy score that we report corresponds to the point on the ROC that lies along the line between (0,100) and (100,0). This is also the point at which the Wolpaw information transfer rate (ITR) is estimated, whereas the Nykopp ITR estimation finds a point that lies further left along the ROC curve. The area under the curve is highly correlated with classification accuracy (over all regularization schemes and decoding segment lengths, [$r = 0.99$, $T_{(862)} = 219.9$, $p < 1.0 \times 10^{-16}$]. The Nykopp ITR, on the other hand lies further left along the ROC curve, demonstrating that by avoiding the classification of some trials, it is possible to maximize the ITR.

3.5. Information Transfer Rate

The Wolpaw ITR represents the transfer rate when all decoding segments are classified, whereas the Nykopp ITR represents the maximum achievable transfer rate when some classifications are withheld based on classification discrimination function output. **Figure 6** shows the Wolpaw and Nykopp ITR values as a function of decoding segment duration, based on models computed with Tikhonov regularization. Both the Wolpaw and Nykopp ITR show an increase followed by a decrease with increasing decoding segment duration. The plots suggest that for brain computer interface applications with fixed decoding segment lengths, it may be advisable to use decoding segments of 3–5 s to maximize the ITR. While the Nykopp measure is an upper-bound, its increase over the Wolpaw ITR value [forward model, 5 s: $T_{(17)} = 13.1$, $p = 1.3 \times 10^{-10}$], [backward model, 5 s: $T_{(17)} = 16.7$, $p = 2.7 \times 10^{-12}$] demonstrates that by adjusting the classifier decision function cutoff, it could be possible to increase the ITR.

4. DISCUSSION

In this study, we systematically investigated the effects of forward/backward stimulus-response model estimation methods on the ability to decode and classify attended speech envelopes from single-trial EEG responses to speech mixtures. The performance of stimulus/EEG decoders based on forward models (mapping from attended speech envelopes to multi-channel EEG responses) and backward models (mapping from EEG response back to speech envelopes) were compared. It was found that the backward models outperformed the forward models in terms of regression and classification accuracies. While forward models could be expected to have higher regression accuracies due to the averaging of correlation coefficients across channels for forward models, the regression accuracy for the backward model was still higher when compared to the maximum correlation coefficient across channels for the forward model. We hypothesize that the models do a better job of reconstructing audio (the backward model) than predicting EEG data (the forward model) because the EEG data contains a lot of information from other brain functions. It is impossible to predict these signals from the stimulus, hence the limited success of a forward model, but it is possible to filter them out, hence the better performance of a backward model. There are also other fundamental differences between the models, such as statistical and structural properties of the regressor variable, and number of parameters estimated. For instance, the eigenspectrum of the EEG autocovariance matrix in **Figure A3** in Supplementary Material suggests that the matrix is ill-conditioned, particularly compared to that of

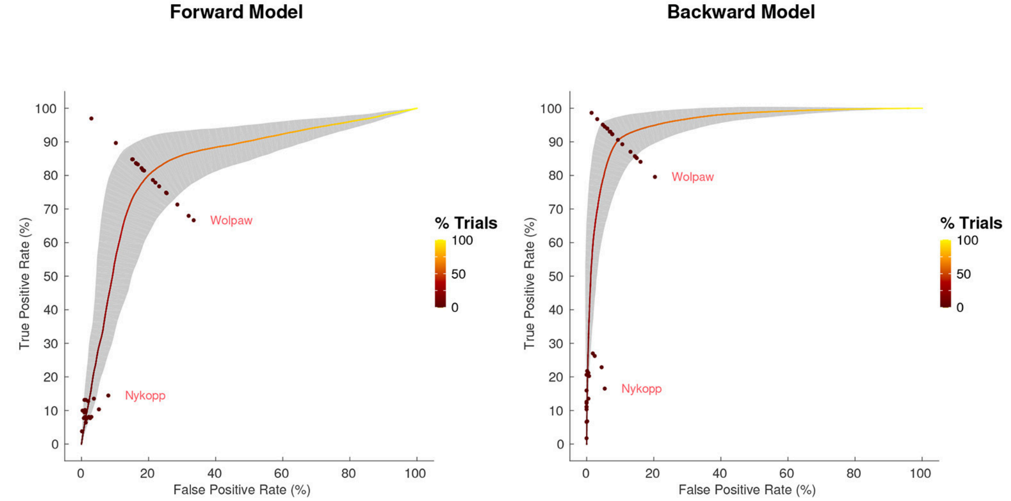


FIGURE 5 | Average receiver operating characteristic curve, with standard deviation band, for 30 s decoding segments using Tikhonov regularization. Points at which Wolpaw and Nykopp information transfer rates were evaluated for each subject are shown. Color along curve indicates percentage of decoding segment trials evaluated to obtain each point. The gray band indicates the standard deviation boundaries of the curve in both x and y directions.

the speech envelope. Different regularization schemes were not found to significantly affect the forward model classification accuracies. However, for the backward models, the decoding schemes that yielded the best classification accuracy were Ridge Regression, LRA, Shrinkage and Tikhonov. Lasso had a lower classification accuracy by a small but significant margin. Classification accuracy increased monotonically as a function of duration, reflecting the greater amount of discriminative information available in longer segments. ITR however peaked at an intermediate segment duration, reflecting the tradeoff between the accuracy of individual classification judgments (greater at long durations) and number of judgments (greater at short durations). The optimum was around 3–5 s.

For the analysis, we used different linear approaches to decode selective auditory attention from stimulus and EEG data. These analyses all relied on the explicit assumption that the human cortical activity selectively tracks attended and unattended speech envelopes. To fit the models, we made a number of choices based on common practices in literature, and with the goal of being able to compare forward/backward models and regularization schemes. For example, a 500 ms kernel was used as was done by others (Fuglsang et al., 2017). While shorter kernels have been explored as well (O'Sullivan et al., 2015), a longer one tests the ability of the model estimation method to handle a larger dimensionality and allows for a more flexible stimulus-response modeling capturing both early and late attentional modulations of the neural response. Additionally, we chose to focus on 1–9 Hz EEG activity as the attentional modulation of EEG data has been found prominent in this range. It is likely that other neural frequency bands robustly track attended speech (e.g., high gamma power Pasley et al., 2012) and that the neural decoders potentially could benefit from having access to other neural frequency bands. This is, however, outside the scope of this paper.

4.1. Decoding Selective Auditory Attention With Forward and Backward Models

The forward models performed significantly worse than the backward models in terms of classification accuracies. Single-trial scalp EEG signals are inherently noisy, in part because activity picked up by each electrode reflects a superposition of activity from signals that are not related to the selective speech processing (Blankertz et al., 2011). We refer here to any aspects of the EEG signals that systematically synchronize with the attended speech streams as target signals and anything that does not as noise. To improve the signal-to noise ratio one can efficiently use spatio-temporal filtering techniques. This in part relates to the fact that stimulus-irrelevant neural activity tends to be spatially correlated across electrodes. The spatio-temporal backward models implicitly exploit these redundancies to effectively filter out noise and improve signal-to-noise-ratio. This makes them fairly robust to spatially correlated artifact activity (e.g., electro-ocular and muscle artifacts) when trained on data from a large number of electrodes. This is also reflected in the high classification accuracies that were obtained with the backward models. However, for the relatively high number of electrodes used in this study, it was found that the spatio-temporal reconstruction filters were effective only when properly regularized.

The forward models, on the other hand, attempt to predict the neural responses of each electrode in a mass-univariate approach. These models do not, therefore, explicitly use cross-channel information to regress out stimulus-irrelevant activity. The relative contribution of the individual channels to the classification accuracies were instead found via an SVM trained on correlation coefficients computed per channel, over short time segments. In short, backward models remove spatial information prior to classification when regressing out non-stimulus-related

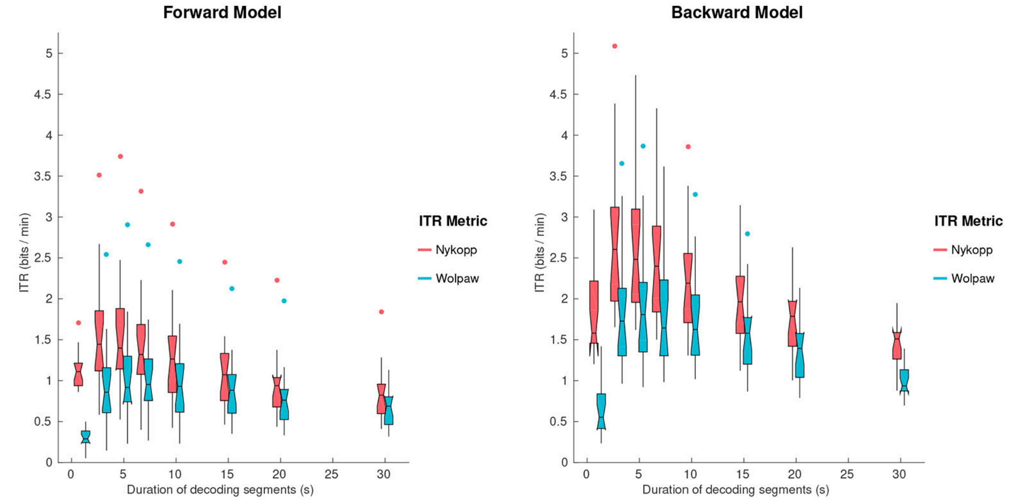


FIGURE 6 | Wolpaw and Nykopp information transfer rates (ITR) as a function of decoding segment duration for the forward and backward models, using Tikhonov regularization. Notched boxplots show median, and first and third quartiles. Whiskers show 1.5 × IQR. Dots indicate outliers.

activity, whereas forward models preserve this information, but do not regress out non-stimulus-related activity. It can therefore be beneficial to apply dimensionality reduction techniques [e.g., independent component analysis (Bell and Sejnowski, 1995) or joint decorrelation (de Cheveigné and Parra, 2014)] to represent the EEG data as a linear combination of fewer latent components prior to fitting the forward models. Alternatively, canonical component analysis can be used to jointly derive spatio-temporal filters for both audio and EEG such that the correlation between the filtered data is maximized (de Cheveigné et al., 2018).

4.1.1. Regularization

Each regularization scheme makes certain assumptions and simplifications that are therefore adopted by studies employing them. Because these methods have not been previously evaluated side by side, it is unknown how valid these assumptions are.

While no regularization (OLS) was found to work well for forward models in producing classification accuracies roughly in line with regularized models, this method performs relatively poorly when applied to backward models. This is likely reflective of the higher dimensional kernel required for the backward problem. For comparison, a forward model had 33 parameters (per channel) that needed to be fit, whereas a backward model had 2,178 parameters.

We generally found that the reconstruction accuracies (r_{attend}) plateaued over a large range of λ values for linear models (**Figure A1**).

Elastic net regularization permits the adjustment of the balance between L1 and L2 regularization via the α parameter. For the backward model, it was shown that a smaller α value improved the correlation between the reconstructed and attended audio stream by only a narrow margin.

The α value had no significant impact on classification accuracy for either forward or backward models. As such, the higher classification performance of Ridge Regression ($\alpha = 0$), compared to Lasso ($\alpha = 1$) may be a result of differences between the closed form solution used for Ridge Regression and the coordinate descent solution used for the Elastic Net, as well as between the solvers themselves (MATLAB's *mldivide* vs. GLMNET, Qian et al., 2013).

Another coordinate descent method, known as boosting, has been used in several studies (David et al., 2007; Calabrese et al., 2011; Thorson et al., 2015). It has been shown that boosting promotes sparse solutions in the context of spectro-temporal receptive fields with single-unit recordings (David et al., 2007). This method was not explored in the present study because boosting tends to be computationally intractable for backward models due to the high number of parameters, and because it involves a large set of hyperparameters. This makes a direct comparison of the regularization methods difficult. Instead we used the Elastic Net algorithm to investigate how the stimulus-response models could benefit from sparsity.

For the forward model, all regularization schemes yielded regression and classification accuracies that were not significantly different from each other. For the backward model, Tikhonov regularization yielded the best regression accuracy, despite the fact that cross-channel leakage may have lead to a suboptimal solution. However, it was found that the improved regression accuracy did not lead to a better classification accuracy compared to other regression schemes with closed-form solutions (i.e., Ridge, Shrinkage, and LRA) due to an associated increased variance in the correlation coefficient computed over short decoding segment lengths. It has been reported that, in practice, the Ridge Regression approach appears to perform better than LRA (Vajargah, 2013); however, no significant difference was found in the present study. LRA removes lower variance components after the eigendecomposition of $\mathbf{X}^T\mathbf{X}$, essentially performing a hard-threshold. In contrast, Ridge Regression is a smooth down-weighting of lower-variance components (Blankertz et al., 2011).

4.2. Realtime Performance

The information transfer rate results provide insight into how classification performance can be optimized. It is worth noting that the ITR measures represent particular points along the ROC curve, as is illustrated in **Figure 5**. For a binary classification problem, with balanced classes, the Wolpaw ITR corresponds to the point on the ROC curve along the line connecting the corners of the plot at coordinates (100,0) and (0,100). The Nykopp ITR, on the other hand corresponds to the point that maximizes the ITR, essentially trading the number of classified samples for increased classification accuracy. In practice, other considerations besides ITR can influence the choice of the point on the ROC. For instance, if there is a high penalty on incorrect classifications, then the classifier threshold may be adjusted to operate at another point on the ROC curve. In short, the ROC and ITR are useful tools in identifying a suitable balance between sensitivity and specificity.

The ITR results in the present study suggest a 3–5 s decoding segment length to achieve the maximum bit-rate. It should be noted that this assumes that switches in attention can occur frequently, on the order of the decoding segment length, such as in a real-world cognitive control setting where system response latency is an important constraint. In cases, where switches in attention are known to be sparse *a priori*, it may instead be more desirable to increase decoding segment length and sacrifice bit-rate to put more emphasis on accuracy, since the loss in bit rate due to long decoding segments is only evident during attention switches. Such an approach was taken by O'Sullivan et al. (2017), where the theoretical performance of a realtime backward model decoding system was characterized for switches in attention every 60 s. In that study, a decoding segment length between 15 and 20 s was reported as optimal to achieve the best speed-accuracy tradeoff.

4.3. Summary

There are many methods that can be used to compute forward/backward stimulus-response models. The present study uses a baseline dataset and procedures for the evaluation of these methods. In consideration of the multiple applications in which forward/backward models are used, primarily dealing with reconstruction accuracies or classification performance, this paper considered multiple metrics of performance. By characterizing the regularization and performance of the model estimation methods, and the relationship between performance metrics, a more complete understanding of the validity of the assumptions underlying each method is provided, as well as the impact of the assumptions on the end result. While these experiments were done with EEG data, we expect that the results apply equally to magnetoencephalography (MEG) data. The key findings from this study were (1) the importance of regularization for the backward model, (2) the superior performance of Tikhonov regularization in achieving higher regression accuracy although this does not necessarily entail superior classification performance, and (3) optimal ITR can be achieved in the 3–5 s range and by adjusting the classifier discrimination function threshold.

AUTHOR CONTRIBUTIONS

DW, SF, JH, EC, MS, and AdC contributed to the code used in the paper. DW, SF, JH, and AdC determined the data analysis procedure. DW created some of the figures, performed statistical analyses, wrote parts of the paper, and was responsible for the overall paper. SF created some of the figures, and wrote parts of the paper. JH, MS, and AdC provided critical feedback on the paper.

FUNDING

This work was supported by the EU H2020-ICT grant 644732 (COCOHA), and grants ANR-10-LABX-0087 IEC and ANR-10-IDEX-0001-02 PSL.

ACKNOWLEDGMENTS

This paper draws on work performed at the 2015 and 2016 Telluride Neuromorphic Cognition Engineering workshops.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fnins.2018.00531/full#supplementary-material

REFERENCES

Aroudi, A., and Doclo, S. (2017). "EEG-based auditory attention decoding: impact of reverberation, noise and interference reduction," in *39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (Jeju Island).

Bell, A. J., and Sejnowski, T. J. (1995). An information-maximization approach to blind separation and blind deconvolution. *Neural Comput.* 7, 1129–1159. doi: 10.1162/neco.1995.7.6.1129

Bialek, W., Rieke, F., de Ruyter Van Steveninck, R., and Warland, D. (1991). Reading a neural code. *Science* 252, 1854–1857. doi: 10.1126/science.2063199

Blankertz, B., Lemm, S., Treder, M., Haufe, S., and Müller, K. (2011). Single-trial analysis and classification of ERP components—a tutorial. *Neuroimage* 56, 814–825. doi: 10.1016/j.neuroimage.2010.06.048

Broderick, M. P., Anderson, A. J., Di Liberto, G. M., Crosse, M. J., and Lalor, E. C. (2018). Electrophysiological correlates of semantic dissimilarity reflect the comprehension of natural, narrative speech. *Curr. Biol.* 28, 803–809.e3. doi: 10.1016/j.cub.2018.01.080

Calabrese, A., Schumacher, J., Schneider, D., Paninski, L., and Woolley, S. (2011). A generalized linear model for estimating spectrotemporal receptive fields from responses to natural sounds. *PLoS ONE* 6:e16104. doi: 10.1371/journal.pone.0016104

Crosse, M. J., Butler, J., and Lalor, E. (2015). Congruent visual speech enhances cortical entrainment to continuous auditory speech in noise-free conditions. *J. Neurosci.* 35, 14195–14204. doi: 10.1523/JNEUROSCI.1829-15.2015

Crosse, M, J., Di Liberto, G., Bednar, A., and Lalor, E. (2016a). The multivariate temporal response function (mTRF) toolbox: a MATLAB toolbox for relating neural signals to continuous stimuli. *Front. Hum. Neurosci.* 10:604. doi: 10.3389/fnhum.2016.00604

Crosse, M. J., Di Liberto, G., and Lalor, E. (2016b). Eye can hear clearly now: inverse effectiveness in natural audiovisual speech processing relies on long-term crossmodal temporal integration. *J. Neurosci.* 36, 9888–9895. doi: 10.1523/JNEUROSCI.1396-16.2016

Das, N., Van Eyndhoven, S., Francart, T., and Bertrand, A. (2016). Adaptive attention-driven speech enhancement for EEG-informed hearing prostheses. *Conf. Proc. IEEE Eng. Med. Biol. Soc.* 2016, 77–80. doi: 10.1109/EMBC.2016.7590644

David, S. V., Mesgarani, N., and Shamma, S. (2007). Estimating sparse spectro-temporal receptive fields with natural stimuli. *Netw. Comput. Neural Syst.* 18, 191–212. doi: 10.1080/09548980701609235

David, S. V., Vinje, W., and Gallant, J. (2004). Natural stimulus statistics alter the receptive field structure of v1 neurons. *J. Neurosci.* 24, 6991–7006. doi: 10.1523/JNEUROSCI.1422-04.2004

de Cheveigné, A., and Arzounian, D. (2017). Robust detrending, rereferencing, outlier detection, and inpainting for multichannel data. *bioRxiv* : 232892. [preprint] doi: 10.1101/232892

de Cheveigné, A., and Parra, L. (2014). Joint decorrelation: a versatile tool for multichannel data analysis. *Neuroimage* 98, 487–505. doi: 10.1016/j.neuroimage.2014.05.068

de Cheveigné, A., Wong, D., Di Liberto, G., Hjortkjær, J., Slaney, M., and Lalor, E. (2018). Decoding the auditory brain with canonical component analysis. *Neuroimage* 172, 206–216. doi: 10.1016/j.neuroimage.2018.01.033

Di Liberto, G., O'Sullivan, J., and Lalor, E. (2015). Low-frequency cortical entrainment to speech reflects phoneme-level processing. *Curr. Biol.* 25, 2457–2465. doi: 10.1016/j.cub.2015.08.030

Ding, N., and Simon, J. (2012a). Emergence of neural encoding of auditory objects while listening to competing speakers. *Proc. Natl. Acad. Sci. U.S.A.* 109, 11854–11859. doi: 10.1073/pnas.1205381109

Ding, N., and Simon, J. (2012b). Neural coding of continuous speech in auditory cortex during monaural and dichotic listening. *J. Neurophysiol.* 107, 78–89. doi: 10.1152/jn.00297.2011

Ding, N., and Simon, J. (2013). Adaptive temporal encoding leads to a background-insensitive cortical representation of speech. *J. Neurosci.* 33, 5728–5735. doi: 10.1523/JNEUROSCI.5297-12.2013

Ding, N., and Simon, J. (2014). Cortical entrainment to continuous speech: functional roles and interpretations. *Front. Hum. Neurosci.* 8:311. doi: 10.3389/fnhum.2014.00311

Friedman, J. (1989). Regularized discriminant analysis. *J. Am. Stat. Assoc.* 84, 165–175. doi: 10.1080/01621459.1989.10478752

Fuglsang, S. A., Dau, T., and Hjortkjær, J. (2017). Noise-robust cortical tracking of attended speech in real-world acoustic scenes. *Neuroimage* 156, 435–444. doi: 10.1016/j.neuroimage.2017.04.026

Fuglsang, S., Wong, D., and Hjortkjær, J. (2018). Data from: EEG and audio dataset for auditory attention decoding. *Zenodo*. doi: 10.5281/zenodo.1199011

Goutte, C., Nielsen, F., and Hansen, K. (2000). Modeling the hemodynamic response in fmri using smooth fir filters. *IEEE Trans. Med. Imag.* 19, 1188–1201. doi: 10.1109/42.897811

Hastie, T., Tibshirani, R., and Friedman, J. (2001). "Linear methods for regression," in *The Elements of Statistical Learning Theory*, Ch. 3 (New York, NY: Springer), 43–100.

Haufe, S., Meinecke, F., Görgen, K., Dähne, S., Haynes, J., Blankertz, B., et al. (2014). On the interpretation of weight vectors of linear models in multivariate neuroimaging. *Neuroimage* 87, 96–110. doi: 10.1016/j.neuroimage.2013.10.067

Holdgraf, C. R., de Heer, W., Pasley, B., Rieger, J., Crone, N., Lin, J., et al. (2016). Rapid tuning shifts in human auditory cortex enhance speech intelligibility. *Nat. Commun.* 7:13654. doi: 10.1038/ncomms13654

Holdgraf, C. R., Rieger, J., Micheli, C., Martin, S., Knight, R., and Theunissen, F. (2017). Encoding and decoding models in cognitive electrophysiology. *Front. Syst. Neurosci.* 11:61. doi: 10.3389/fnsys.2017.00061

Lalor, E. C., and Foxe, J. (2010). Neural responses to uninterrupted natural speech can be extracted with precise temporal resolution. *Eur. J. Neurosci.* 31, 189–193. doi: 10.1111/j.1460-9568.2009.07055.x

Lalor, E. C., Pearlmutter, B., Reilly, R., McDarby, G., and Foxe, J. (2006). The VESPA: a method for the rapid estimation of a visual evoked potential. *Neuroimage* 32, 1549–1561. doi: 10.1016/j.neuroimage.2006.05.054

Lalor, E. C., Power, A. J., Reilly, R. B., and Foxe, J. J. (2009). Resolving precise temporal processing properties of the auditory system using continuous stimuli. *J. Neurophysiol.* 102, 349–359. doi: 10.1152/jn.90896.2008

Machens, C. K., Wehr, M., and Zador, A. (2004). Linearity of cortical receptive fields measured with natural sounds. *J. Neurosci.* 24, 1089–1100. doi: 10.1523/JNEUROSCI.4445-03.2004

Machens, C. K., Wehr, M., and Zador, A. M. (2003). "Spectro-temporal receptive fields of subthreshold responses in auditory cortex," in *Advances in Neural Information Processing Systems* (Vancouver, BC), 149–156.

Marconato, A., Ljung, L., Rolain, Y., and Schoukens, J. (2014). Linking regularization and low-rank approximation for impulse response modeling. *IFAC Proc. Vol.* 47, 4999–5004. doi: 10.3182/20140824-6-ZA-1003.00254

Mesgarani, N., and Chang, E. (2012). Selective cortical representation of attended speaker in multi-talker speech perception. *Nature* 485, 233–236. doi: 10.1038/nature11020

Mesgarani, N., David, S., Fritz, J., and Shamma, S. (2009). Influence of context and behavior on stimulus reconstruction from neural activity in primary auditory cortex. *J. Neurophysiol.* 102, 3329–3339. doi: 10.1152/jn.91128.2008

Mirkovic, B., Bleichner, M., De Vos, M., and Debener, S. (2016). Target speaker detection with concealed EEG around the ear. *Front. Neurosci.* 10:349. doi: 10.3389/fnins.2016.00349

Mirkovic, B., Debener, S., Jaeger, M., and De Vos, M. (2015). Decoding the attended speech stream with multi-channel EEG: implications for online, daily-life applications. *J. Neural Eng.* 12:046007. doi: 10.1088/1741-2560/12/4/046007

Nykopp, T. (2001). *Statistical Modelling Issues for the Adaptive Brain Interface.* MSc thesis, Helsinki University of Technology, Finland.

Oostenveld, R., Fries, P., Maris, E., and Schoffelen, J. (2011). Fieldtrip: open source software for advanced analysis of MEG, EEG, and invasive electrophysiological data. *Comput. Intell. Neurosci.* 2011:156869. doi: 10.1155/2011/156869

O'Sullivan, A. E., Crosse, M., Di Liberto, G., and Lalor, E. (2017). Visual cortical entrainment to motion and categorical speech features during silent lipreading. *Front. Hum. Neurosci.* 10:679. doi: 10.3389/fnhum.2016.00679

O'Sullivan, J., Chen, Z., Herrero, J., McKhann, G., Sheth, S., Mehta, A., et al. (2017). Neural decoding of attentional selection in multi-speaker environments without access to clean sources. *J. Neural Eng.* 14:056001. doi: 10.1088/1741-2552/aa7ab4

O'Sullivan, J. A., Power, A., Mesgarani, N., Rajaram, S., Foxe, J., Shinn-Cunningham, B., et al. (2015). Attentional selection in a cocktail party environment can be decoded from single-trial EEG. *Cereb. Cortex* 25, 1697–1706. doi: 10.1093/cercor/bht355

Pasley, B. N., David, S., Mesgarani, N., Flinker, A., Shamma, S., Crone, N., et al. (2012). Reconstructing speech from human auditory cortex. *PLoS Biol.* 10:e1001251. doi: 10.1371/journal.pbio.1001251

Patterson, R., Nimmo-Smith, I., Holdsworth, J., and Rice, P. (1987). "An efficient auditory filterbank based on the gammatone function," in *Meeting of the IOC Speech Group on Auditory Modelling at RSRE*, Vol. 2.

Plack, C. J., Oxenham, A., Simonson, A., O'Hanlon, C., Drga, V., and Arifianto, D. (2008). Estimates of compression at low and high frequencies using masking additivity in normal and impaired ears. *J. Acoust. Soc. Am.* 123, 4321–4330. doi: 10.1121/1.2908297

Power, A., Foxe, J., Forde, E., Reilly, R., and Lalor, E. (2012). At what time is the cocktail party? a late locus of selective attention to natural speech. *Eur. J. Neurosci.* 35, 1497–1503. doi: 10.1111/j.1460-9568.2012.08060.x

Power, A., Reilly, R., and Lalor, E. (2011). "Comparing linear and quadratic models of the human auditory system using EEG," in *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE* (Boston, MA: IEEE), 4171–4174. doi: 10.1109/IEMBS.2011.6091035

Puvvada, K., and Simon, J. (2017). Cortical representations of speech in a multitalker auditory scene. *J. Neurosci.* 37, 9189–9196. doi: 10.1523/JNEUROSCI.0938-17.2017

Qian, J., Hastie, T., Friedman, J., Tibshirani, R., and Simon, N. (2013). *Glmnet for Matlab*. Available online at: http://www.stanford.edu/~hastie/glmnet_matlab/

Ringach, D., and Shapley, R. (2004). Reverse correlation in neurophysiology. *Cogn. Sci.* 28, 147–166. doi: 10.1207/s15516709cog2802_2

Schoppe, O., Harper, N., Willmore, B., King, A., and Schnupp, J. (2016). Measuring the performance of neural models. *Front. Comput. Neurosci.* 10:10. doi: 10.3389/fncom.2016.00010

Theunissen, F., David, S., Singh, N., Hsu, A., Vinje, W., and Gallant, J. (2001). Estimating spatio-temporal receptive fields of auditory and visual neurons from their responses to natural stimuli. *Netw. Comput. Neural Syst.* 12, 289–316. doi: 10.1080/net.12.3.289.316

Theunissen, F., Sen, K., and Doupe, A. (2000). Spectral-temporal receptive fields of nonlinear auditory neurons obtained using natural sounds. *J. Neurosci.* 20, 2315–2331. doi: 10.1523/JNEUROSCI.20-06-02315.2000

Thorson, I., Liénard, J., and David, S. (2015). The essential complexity of auditory receptive fields. *PLoS Comput. Biol.* 11:e1004628. doi: 10.1371/journal.pcbi.1004628

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *J. R. Stat. Soc. B* 58, 267–288.

Tikhonov, A. N. (1963). Solution of incorrectly formulated problems and the regularization method. *Soviet Math. Dokl.* 4, 1035–1038.

Vajargah, K. (2013). Comparing ridge regression and principal components regression by monte carlo simulation basedon MSE. *J. Comput. Sci. Comput. Math.* 3, 25–29. doi: 10.20967/jcscm.2013.02.005

Van Eyndhoven, S., Francart, T., and Bertrand, A. (2017). EEG-informed attended speaker extraction from recorded speech mixtures with application in neuro-steered hearing prostheses. *IEEE Trans. Biomed. Eng.* 64, 1045–1056. doi: 10.1109/TBME.2016.2587382

Widmann, A., Schröger, E., and Maess, B. (2015). Digital filter design for electrophysiological data–a practical approach. *J. Neurosci. Methods* 250, 34–46. doi: 10.1016/j.jneumeth.2014.08.002

Willmore, B., Schoppe, O., King, A., Schnupp, J., and Harper, N. (2016). Incorporating midbrain adaptation to mean sound level improves models of auditory cortical processing. *J. Neurosci.* 36, 280–289. doi: 10.1523/JNEUROSCI.2441-15.2016

Wolpaw, J., and Ramoser, H. (1998). EEG-based communication: improved accuracy by response verification. *IEEE Trans. Rehabil. Eng.* 6, 326–333. doi: 10.1109/86.712231

Wu, M., David, S., and Gallant, J. (2006). Complete functional characterization of sensory neurons by system identification. *Annu. Rev. Neurosci.* 29, 477–505. doi: 10.1146/annurev.neuro.29.051605.113024

Zink, R., Proesmans, S., Bertrand, A., Van Huffel, S., and De Vos, M. (2017). Online detection of auditory attention with mobile EEG: closing the loop with neurofeedback. *BioRxiv*. doi: 10.1101/218727

Zion Golumbic, E., Ding, N., Bickel, S., Lakatos, P., Schevon, C., McKhann, G., et al. (2013). Mechanisms underlying selective neuronal tracking of attended speech at a "cocktail party". *Neuron* 77, 980–991. doi: 10.1016/j.neuron.2012.12.037

Zou, H., and Hastie, T. (2005). Regularization and variable selection via the elastic net. *J. R. Stat. Soc. Ser. B Stat. Methodol.* 67, 301–320. doi: 10.1111/j.1467-9868.2005.00503.x

Conflict of Interest Statement: MS was employed by Google.

The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Neuroscience

ORIGINAL RESEARCH
published: 01 May 2018
doi: 10.3389/fnins.2018.00262

Real-Time Tracking of Selective Auditory Attention From M/EEG: A Bayesian Filtering Approach

Sina Miran[1], *Sahar Akram*[2], *Alireza Sheikhattar*[1], *Jonathan Z. Simon*[1,3,4], *Tao Zhang*[5] and *Behtash Babadi*[1,3]*

[1] Department of Electrical and Computer Engineering, University of Maryland, College Park, MD, United States, [2] Facebook, Menlo Park, CA, United States, [3] Institute for Systems Research, University of Maryland, College Park, MD, United States, [4] Department of Biology, University of Maryland, College Park, MD, United States, [5] Starkey Hearing Technologies, Eden Prairie, MN, United States

Humans are able to identify and track a target speaker amid a cacophony of acoustic interference, an ability which is often referred to as the cocktail party phenomenon. Results from several decades of studying this phenomenon have culminated in recent years in various promising attempts to decode the attentional state of a listener in a competing-speaker environment from non-invasive neuroimaging recordings such as magnetoencephalography (MEG) and electroencephalography (EEG). To this end, most existing approaches compute correlation-based measures by either regressing the features of each speech stream to the M/EEG channels (the decoding approach) or vice versa (the encoding approach). To produce robust results, these procedures require multiple trials for training purposes. Also, their decoding accuracy drops significantly when operating at high temporal resolutions. Thus, they are not well-suited for emerging real-time applications such as smart hearing aid devices or brain-computer interface systems, where training data might be limited and high temporal resolutions are desired. In this paper, we close this gap by developing an algorithmic pipeline for real-time decoding of the attentional state. Our proposed framework consists of three main modules: (1) Real-time and robust estimation of encoding or decoding coefficients, achieved by sparse adaptive filtering, (2) Extracting reliable markers of the attentional state, and thereby generalizing the widely-used correlation-based measures thereof, and (3) Devising a near real-time state-space estimator that translates the noisy and variable attention markers to robust and statistically interpretable estimates of the attentional state with minimal delay. Our proposed algorithms integrate various techniques including forgetting factor-based adaptive filtering, ℓ_1-regularization, forward-backward splitting algorithms, fixed-lag smoothing, and Expectation Maximization. We validate the performance of our proposed framework using comprehensive simulations as well as application to experimentally acquired M/EEG data. Our results reveal that the proposed real-time algorithms perform nearly as accurately as the existing state-of-the-art offline techniques, while providing a significant degree of adaptivity, statistical robustness, and computational savings.

Keywords: attention, auditory, real-time, dynamic estimation, EEG, MEG, state-space models, Bayesian filtering

OPEN ACCESS

Edited by:
Malcolm Slaney,
Google, United States

Reviewed by:
Alain De Cheveigne,
École Normale Supérieure, France
Dan Zhang,
Tsinghua University, China
Edmund C. Lalor,
University of Rochester, United States

****Correspondence:***
Behtash Babadi
behtash@umd.edu

Specialty section:
This article was submitted to
Auditory Cognitive Neuroscience,
a section of the journal
Frontiers in Neuroscience

Received: *18 November 2017*
Accepted: *05 April 2018*
Published: *01 May 2018*

Citation:
Miran S, Akram S, Sheikhattar A, Simon JZ, Zhang T and Babadi B (2018) Real-Time Tracking of Selective Auditory Attention From M/EEG: A Bayesian Filtering Approach. Front. Neurosci. 12:262. doi: 10.3389/fnins.2018.00262

1. INTRODUCTION

The ability to select a single speaker in an auditory scene, consisting of multiple competing speakers, and maintain attention to that speaker is one of the hallmarks of human brain function. This phenomenon has been referred to as the cocktail party effect (Brungart, 2001; Haykin and Chen, 2005; McDermott, 2009). The mechanisms underlying the real-time process by which the brain segregates multiple sources in a cocktail party setting, have been the topic of active research for decades (Cherry, 1953; Middlebrooks et al., 2017). Although the details of these mechanisms are for the most part unknown, various studies have pointed to the role of specific neural processes involved in this function. As the acoustic signals propagate through the auditory pathway, they are decomposed into spectrotemporal features at different stages, and a rich representation of the complex auditory environment reaches the auditory cortex. It has been hypothesized that the perception of an auditory object is the result of adaptive binding as well as discounting of these features (Bregman, 1994; Griffiths and Warren, 2004; Fishman and Steinschneider, 2010; Shamma et al., 2011).

From a computational modeling perspective, there have been several attempts at designing so-called "attention decoders," where the goal is to reliably decode the attentional focus of a listener in a multi-speaker environment using non-invasive neuroimaging techniques like electroencephalography (EEG) (Power et al., 2012; Mirkovic et al., 2015; O'Sullivan et al., 2015; Zink et al., 2017) and magnetoencephalography (MEG) (Ding and Simon, 2012a,b; Akram et al., 2014, 2016, 2017). These methods are typically based on reverse correlation or estimating linear encoding/decoding models using off-line regression techniques, and thereby detecting specific lags in the model coefficients that are modulated by the attentional state (Kaya and Elhilali, 2017). For instance, encoding coefficients comprise salient peaks at a typical lag of ~100 ms for MEG (Ding and Simon, 2012a), and envelope reconstruction performance is optimal at a lag of ~200 ms for EEG (O'Sullivan et al., 2015).

Although the foregoing approaches have proven successful in reliable attention decoding, they have two major limitations that make them less appealing for emerging real-time applications such as Brain-Computer Interface (BCI) systems and smart hearing aids. First, the temporal resolution of existing approaches for reliable attention decoding is on the order of ~10 s, and their decoding accuracy drops significantly when operating at temporal resolutions of ~1 s, i.e., the time scale at which humans are able to switch attention from one speaker to another (Zink et al., 2016, 2017). Second, approaches based on linear regression (e.g., reverse correlation) need large training datasets, often from multiple subjects and trials, to estimate the decoder/encoder reliably. Access to such training data is only possible through repeated calibration stages, which may not always be possible in real-time applications with potential variations in recording settings. While recent results (Akram et al., 2014, 2016) address the first shortcoming by employing state-space models and thereby producing robust estimates of the attentional state from limited data at high temporal resolutions, they are not yet suitable for real-time applications as they operate in the so-called "batch-mode" regime, i.e., they require the entire data from a trial at once in order to estimate the attentional state.

In this paper, we close this gap by designing a modular framework for real-time attention decoding from non-invasive M/EEG recordings that overcomes the aforementioned limitations using techniques from Bayesian filtering. Our proposed framework includes three main modules. The first module pertains to estimating *dynamic* models of decoding/encoding in *real-time*. To this end, we use the forgetting factor mechanism of the Recursive Least Squares (RLS) algorithm together with the ℓ_1 regularization penalty from Lasso to capture the dynamics in the data while preventing overfitting (Sheikhattar et al., 2015a; Akram et al., 2017). The real-time inference is then efficiently carried out using a Forward-Backward Splitting (FBS) procedure (Combettes and Pesquet, 2011). In the second module, we extract an attention-modulated feature, which we refer to as "attention marker," as a function of the M/EEG recordings, the estimated encoding/decoding coefficients, and the auditory stimuli. For instance, the attention marker can be a correlation-based measure or the magnitude of certain peaks in the model coefficients. We carefully design the attention marker features to capture the attention modulation and thereby maximally separate the contributions of the attended and unattended speakers in the neural response in both MEG and EEG applications.

The extracted features are then passed to a novel state-space estimator in the third module, and thereby are translated into probabilistic, robust, and dynamic measures of the attentional state, which can be used for soft-decision making in real-time applications. The state-space estimator is based on Bayesian fixed-lag smoothing, and operates in *near real-time* with controllable delay. The fixed-lag design creates a trade-off between real-time operation and robustness to stochastic fluctuations. In addition, we modify the Expectation-Maximization algorithm and the nonlinear filtering and smoothing techniques of Akram et al. (2016) for real-time implementation. Compared to existing techniques, our algorithms require minimal supervised data for initialization and tuning, which makes them more suitable for the applications of real-time attention decoding with limited training data. In order to validate our real-time attention decoding algorithms, we apply them to both simulated and experimentally recorded EEG and MEG data in dual-speaker environments. Our results suggest that the performance of our proposed framework is comparable to the state-of-the-art results of Mirkovic et al. (2015), O'Sullivan et al. (2015), and Akram et al. (2016), while operating in near real-time with ~2 s delay.

The rest of the paper is organized as follows: In section 2, we develop the three main modules in our proposed framework as well as the corresponding estimation algorithms. We present the application of our framework to both synthetic and experimentally recorded M/EEG data in section 3, followed by discussion and concluding remarks in section 4.

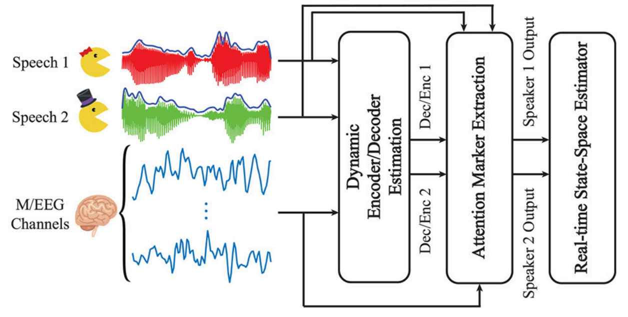

Speech 1
Speech 2
M/EEG Channels
Dynamic Encoder/Decoder Estimation
Dec/Enc 1
Dec/Enc 2
Attention Marker Extraction
Speaker 1 Output
Speaker 2 Output
Real-time State-Space Estimator

FIGURE 1 | A schematic depiction of our proposed framework for real-time tracking of selective auditory attention from M/EEG.

2. MATERIAL AND METHODS

Figure 1 summarizes our proposed framework for real-time tracking of selective auditory attention from M/EEG. In the *Dynamic Encoder/Decoder Estimation* module, the encoding/decoding models are fit to neural data in real-time. The *Attention Marker* module uses the estimated model coefficients as well as the recorded data to compute a feature that is modulated by the instantaneous attentional state. Finally, in the *State-Space Model* module, the foregoing features are refined through a linear state-space model with nonlinear observations, resulting in robust and dynamic estimates of the attentional state.

In section 2.1, we formally define the dynamic encoding and decoding models, and develop low-complexity and real-time techniques for their estimation. This is followed by section 2.2, in which we define suitable attention markers for M/EEG inspired by existing literature. In section 2.3, we propose a state-space model that processes the extracted attention markers in order to produce near real-time estimates of the attentional state with minimal delay.

2.1. Dynamic Encoding and Decoding Models

The role of a neural encoding model is to map the stimulus to the neural response. Inspired by existing literature on attention decoding (Ding and Simon, 2012a; O'Sullivan et al., 2015; Akram et al., 2016), we take the speech envelopes as covariates representing the stimuli. The neural response is manifested in the M/EEG recordings. Encoding models can be used to predict the neural response from the stimulus. In contrast, in a neural decoding model, the goal is to express the stimulus as a function of the neural response. Inspired by previous studies, we consider linear encoding and decoding models in this work.

The encoding and decoding models can be cast as mathematically dual formulations. In a dual-speaker environment, let $s_t^{(1)}$ and $s_t^{(2)}$ denote the speech envelopes (in logarithmic scale), corresponding to speakers 1 and 2, respectively, for $t = 1, 2, \dots, T$. Also, let e_t^c denote the neural response recorded at time t and channel c, for $c = 1, 2, \dots, C$. Throughout the paper, we assume the same sampling frequency f_s for both the M/EEG channels and the envelopes. Consider consecutive and non-overlapping windows of length W, and define $K := \left\lfloor \frac{T}{W} \right\rfloor$. We consider piece-wise constant dynamics for the encoding and decoding coefficients, in which the coefficients assume to be constant over each window. Note that we define the *temporal resolution* in an attention decoding procedure as the duration of a data segment to which a measure of the attentional state is attributed. Therefore, $\frac{W}{f_s}$ determines the temporal resolution in our attention decoding framework.

In the encoding setting, we define the vector $\mathbf{s}_t^{(i)} := [s_t^{(i)}, s_{t-1}^{(i)}, \dots, s_{t-L_e}^{(i)}]^\top$ for $i = 1, 2$, where L_e is the total lag considered in the model. Also, let E_t denote a generic linear combination of $e_t^1, e_t^2, \dots, e_t^C$ with some fixed set of weights. These weights can be set to select a single channel, i.e., $E_t = e_t^c$ for some c, or they can be pre-estimated from training data so that E_t represents the dominant auditory component of the neural response (de Cheveigné and Simon, 2008). The encoding coefficients then relate $\mathbf{s}_t^{(i)}$ to E_t. In the decoding setting, we define the vector $\mathbf{e}_t := [e_t^1, e_t^2, \dots, e_t^C]^\top$ and $\boldsymbol{\mathcal{E}}_t := \left[1, \mathbf{e}_t^\top, \mathbf{e}_{t+1}^\top, \dots, \mathbf{e}_{t+L_d}^\top\right]^\top$, where L_d is the total lag in the decoding model and determines the extent of future neural responses affected by the current stimuli. The decoding coefficients then relate $\boldsymbol{\mathcal{E}}_t$ to $s_t^{(i)}$.

Our goal is to recursively estimate the encoding/decoding coefficients in a real-time fashion as the new data samples become available. In addtion, we aim to simultaneously induce adaptivity of the parameter estimates and capture their sparsity. To this end, we employ the following generic optimization problem:

$$\hat{\boldsymbol{\theta}}_k = \underset{\boldsymbol{\theta}}{\arg\min} \sum_{j=1}^{k} \lambda^{k-j} \parallel \mathbf{y}_j - \mathbf{X}_j \boldsymbol{\theta} \parallel_2^2 + \gamma \parallel \boldsymbol{\theta} \parallel_1, \quad k = 1, 2, \dots, K \tag{1}$$

where $\mathbf{y}_j$ and $\mathbf{X}_j$ are respectively the vector of response variables and the matrix of covariates pertinent to window j, $\boldsymbol{\theta}$ is the parameter vector, $\lambda \in (0, 1]$ is the forgetting factor, and γ is a regularization parameter. The optimization problem of Equation 1 is a modified version of the LASSO problem (Tibshirani, 1996).

For the encoding problem, we define $\mathbf{y}_k := \left[E_{(k-1)W+1}, E_{(k-1)W+2}, \dots, E_{kW}\right]^\top$ and $\mathbf{X}_k^{(i)} := \left[\mathbf{s}_{(k-1)W+1}^{(i)}, \mathbf{s}_{(k-1)W+2}^{(i)}, \dots, \mathbf{s}_{kW}^{(i)}\right]^\top$, for $k = 1, 2, \dots, K$ and $i = 1, 2$. Therefore, the full encoding covariate matrix at the k^{th} window is defined as $\mathbf{X}_k := \left[\mathbb{1}_{W\times 1}, \mathbf{X}_k^{(1)}, \mathbf{X}_k^{(2)}\right]$, where the all-ones vector $\mathbb{1}_{W\times 1}$ corresponds to the regression intercept. In the decoding problem, we define $\mathbf{y}_k = \mathbf{s}_k^{(i)} := \left[s_{(k-1)W+1}^{(i)}, s_{(k-1)W+2}^{(i)}, \dots, s_{kW}^{(i)}\right]^\top$, where $i \in \{1, 2\}$. Also, the full decoding covariate matrix at the k^{th} window is $\mathbf{X}_k := \left[\boldsymbol{\mathcal{E}}_{(k-1)W+1}, \boldsymbol{\mathcal{E}}_{(k-1)W+1}, \dots, \boldsymbol{\mathcal{E}}_{kW}\right]^\top$, for $k = 1, 2, \dots, K$.

The optimization problem of Equation (1) has a useful Bayesian interpretation: if the observation noise were i.i.d. Gaussian, and the parameters were exponentially distributed, it is akin to the maximum *a posteriori* (MAP) estimate of the parameters. The quadratic terms correspond to the exponentially-weighted log-likelihood of the observations up to window k, and the ℓ_1-norm corresponds to the log-density of an independent exponential prior on the elements of $\boldsymbol{\theta}$. The exponential prior serves as an effective regularization to promote sparsity of the estimate $\hat{\boldsymbol{\theta}}_k$. Note that we have $\boldsymbol{\theta} \in \mathbb{R}^{1+2(L_e+1)}$ for the encoding model and $\boldsymbol{\theta} \in \mathbb{R}^{1+C(L_d+1)}$ for the decoding model in (1).

Remark 1. The hyperparameter λ provides a tradeoff between the adaptivity and the robustness of estimated coefficients, and it can be determined based on the inherent dynamics in the data. The case of $\lambda = 1$ corresponds to the natural data log-likelihood, i.e., the batch-mode parameter estimates. It has been shown that $\frac{W}{1-\lambda}$ can serve as the *effective* number of recent samples used to calculate $\hat{\boldsymbol{\theta}}_k$ in (1) (Sheikhattar et al., 2015b). The parameter $\frac{W}{1-\lambda}$ can also be viewed as the dynamic integration time: it needs to be chosen long enough so that the estimation is stable, but also short enough to be able to capture the dynamics of neural process involved in switching attention. The hyperparameter γ controls the tradeoff between the Maximum Likelihood (ML) fit and the sparsity of estimated coefficients, and it is usually determined through cross-validation.

Remark 2. In the decoding problem, Equation (1) is solved separately at each window for each speech envelope, resulting in a set of decoding coefficients per speaker. In the encoding setting, we combine the stimuli as explained and solve Equation (1) once at each window to obtain both of the encoder estimates. If the encoding/decoding coefficients are expected to be sparse in a basis represented by the columns of a matrix $\mathbf{G}$, such as the Haar or Gabor bases, we can replace $\mathbf{X}_j$ in (1) by $\mathbf{X}_j\mathbf{G}$, for $j = 1, 2, \ldots, k$, and solve for $\hat{\boldsymbol{\theta}}_k$ as before. Then, the final encoding/decoding coefficients are given by $\mathbf{G}\hat{\boldsymbol{\theta}}_k$. In the context of encoding models, the coefficients are referred to as the Temporal Response Function (TRF) (Ding and Simon, 2012a; Akram et al., 2017). The TRFs are known to exhibit some degree of sparsity on a basis consisting of shifted Gaussian kernels (see Akram et al., 2017 for details).

Remark 3. It is worth discussing the rationale behind the dynamic updating of the encoding/decoding models, as opposed to considering fixed *canonical* encoding/decoding models common in existing work. First, estimation of the canonical encoding/decoding models in existing literature requires large training datasets. In emerging real-time applications of attention decoding, access to such large supervised training datasets may not be feasible. In addition, slight changes to the electrode placement may require recalibration of the canonical encoders/decoders. Thus, by dynamic updating of the encoding/decoding models we aim at minimizing the amount of supervised training data, which can be a bottleneck in emerging real-time applications.

Second, recent results have shown that dynamics of the encoding/decoding models indeed carry important information regarding the underlying attention process (Ding and Simon, 2012a,b; Power et al., 2012; Zion Golumbic et al., 2013; Akram et al., 2017). Therefore, dynamic estimates of these models can be beneficial in attention decoding. In order to mitigate the variability of our dynamic estimates of the encoding/decoding models, we have employed the ℓ_1-regularized least squares estimation framework with a forgetting factor.

In summary, we argue that the dynamic framework used here is more preferable for real-time applications with limited training data and in the presence of attention dynamics. It is worth noting that our modular framework can still be used if the encoder/decoder models are pre-estimated and fixed. We refer the reader to section 2.3 and Remark 6 for more details.

Remark 4. Throughout the paper, we assume that the envelopes of the clean speech are available. Given that this assumption does not hold in practical scenarios, recent algorithms on the extraction of speech envelopes from acoustic mixtures (Biesmans et al., 2015, 2017; Aroudi et al., 2016; O'Sullivan et al., 2017; Van Eyndhoven et al., 2017) can be added as a pre-processing module to our framework.

Among the many existing algorithms for solving the modified LASSO problem of Equation (1), we choose the Forward-Backward Splitting (FBS) algorithm (Combettes and Pesquet, 2011), also known as the proximal gradient method. When coupled with proper step-size adjustment methods, FBS is well-suited for real-time and low-complexity updates of $\hat{\boldsymbol{\theta}}_k$ at each window. In this work, we have used the FASTA software package (Goldstein et al., 2014) available online (Goldstein et al., 2015), which has built-in features for all the FBS stepsize adjustment methods. A detailed overview of the FBS algorithm and its properties is given in section 1 of the Supplementary Material.

2.2. Attention Markers

We define the *attention marker* as a mapping function from the estimated encoding/decoding coefficients for each speaker as well as the data in each window to positive real numbers. To be more precise, at window k and for speaker i, in the context of encoding models, the attention marker takes the speaker's estimated encoding coefficients $\hat{\boldsymbol{\theta}}_k^{(i)}$, the speaker's covariate matrix $\mathbf{X}_k^{(i)}$, and the M/EEG responses $\mathbf{y}_k$ as inputs; similarly, in the context of decoding models, the attention marker takes the speaker's estimated decoding coefficients $\hat{\boldsymbol{\theta}}_k^{(i)}$, the M/EEG covariate matrix $\mathbf{X}_k$, and the speaker's speech envelope vector $\mathbf{y}_k^{(i)}$ as inputs. In both cases, the attention marker outputs a positive real number, which we denote by $m_k^{(i)}$ henceforth, for $i = 1, 2$ and $k = 1, 2, \ldots, K$. Thus, in the modular design of **Figure 1**, at each window k, the two outputs $m_k^{(1)}$ and $m_k^{(2)}$ are passed from the Attention Marker module to the State-Space Model module as measures of the attentional state at window k.

In O'Sullivan et al. (2015), a correlation-based measure has been adopted in the decoding model to classify the attended and the unattended speeches in a dual-speaker environment. The approach in O'Sullivan et al. (2015) is based on estimating an *attended* (resp. *unattended*) decoder from the training data to reconstruct the attended (resp. unattended) speech envelope from EEG for each trial. Then, the correlation of this reconstructed envelope with each of the two speech envelopes is computed, and the speaker with the larger correlation coefficient is deemed as the attended (resp. unattended) speaker. This method cannot be directly applied to the real-time setting, since the lack of abundant training data hinders reliable estimation of these decoders. However, assuming that the auditory M/EEG response is more influenced by the attended speaker than the unattended one, we can expect that the decoder corresponding to the *attended* speaker exhibits a higher performance in reconstructing the speech envelope it has been trained on. This can be inferred from the findings in O'Sullivan et al. (2015), where a trained *attended* decoder results in 10% more attention decoding accuracy than a trained *unattended* decoder, as well as the findings in Ding and Simon (2012a). Inspired by these results, we can define the attention marker in the decoding scenario as the correlation magnitude between the speech envelope and its reconstruction by the corresponding decoder, i.e., $m_k^{(i)} = f\left(\hat{\boldsymbol{\theta}}_k^{(i)}, \mathbf{X}_k, \mathbf{y}_k^{(i)}\right) := \left|\text{corr}\left(\mathbf{y}_k^{(i)}, \mathbf{X}_k\hat{\boldsymbol{\theta}}_k^{(i)}\right)\right|$ for $i = 1, 2$ and $k = 1, 2, \ldots, K$. As we will demonstrate later in section 3, this attention marker is suitable for the analysis of EEG recordings.

In the context of cocktail party studies using MEG, it has been shown that the magnitude of the negative peak in the TRF of the attended speaker around a lag of 100 ms, referred to as the M100 component, is larger than that of the unattended speaker (Ding and Simon, 2012a; Akram et al., 2016, 2017). Inspired by these findings, in the encoding scenario applied to MEG data, we can define the attention marker $m_k^{(i)}$ to be the magnitude of the $\hat{\boldsymbol{\theta}}_k^{(i)}$ coefficients corresponding to the M100 component, for $i = 1, 2$ and $k = 1, 2, \ldots, K$.

Due to the inherent uncertainties in the M/EEG recordings, the limitations of non-invasive neuroimaging in isolating the relevant neural processes, and the unknown and likely nonlinear processes involved in auditory attention, the foregoing attention markers derived from linear models are not readily reliable indicators of the attentional state. Given ample training data, nevertheless, these attention markers have been validated using batch-mode analysis. However, their usage in a real-time setting at high temporal resolution requires more care, as the limited data in real-time applications and computation over small windows add more sources of uncertainty to the foregoing list. To address this issue, a state-space model is required in the real-time setting to correct for the uncertainties and stochastic fluctuations of the attention markers caused by the limited integration time in real-time application. We will discuss in detail the formulation and advantages of such a state-space model in the following subsection.

2.3. State-Space Model

In order to translate the attention markers $m_k^{(1)}$ and $m_k^{(2)}$, for $k = 1, 2, \ldots, K$, into a robust and statistically interpretable measure of the attentional state, we employ state-space models. Inspired by the models used in Akram et al. (2016), we design a new state-space model and a corresponding estimator that operates in a fixed-lag smoothing fashion, and thereby admits real-time processing while maintaining the benefits of batch-mode state-space models. Recall that the index k corresponds to a window in time ranging from $t = (k-1)W + 1$ to $t = kW$; however, we refer to each index k as an *instance* when talking about the state-space model, so as not to conflate it with the sliding window in the forthcoming treatment.

Figure 2 displays the fixed-lag smoothing design of the state-space estimator. Suppose that we are at the instance $k = k_0$. We consider an *active* sliding window of length $K_A := K_B + K_F + 1$ as shown in **Figure 2**, where K_F and K_B are respectively called the forward-lag and the backward-lag. In order to carry out the computations in real-time, we assume all of the attentional state estimates to be fixed prior to this window and only update our estimates for the instances within, based on $m_k^{(1)}$'s and $m_k^{(2)}$'s inside the window. In a fixed-lag framework, at $k = k_0$, the goal is to provide an estimate of the attentional state at instance $k = k^*$, where $k^* = k_0 - K_F$. Thus, when using a decoding (resp. encoding) model, the *built-in* attention decoding delay of our framework is $(L_d + K_F W)/f_s$ (resp. $K_F W/f_s$) seconds. It is worth noting that in addition to the built-in delay, our attention decoding results are affected by another source of delay, which we refer to as the *transition* delay. The transition delay is due to the forgetting factor mechanism as well as the smoothing effect in the state-space estimation, which we will discuss further in section 3.1. The parameter K_F creates a tradeoff between real-time and robust estimation of the attentional state. For $K_F = 0$, the estimation is carried out fully in real-time; however, the estimates lack robustness to the fluctuations of the outputs of the attention marker block. The backward-lag K_B determines the attention marker samples prior to k^* that are used in the inference procedure, and it controls the computational cost of the state-space model for fixed values of K_F. Throughout the rest of the paper, we use the expression *real-time* for referring to algorithms that operate with a fixed forward-lag of K_F. We will discuss specific choices of K_F and K_B and their implications in section 3.

Suppose we have a sliding window of length K_A where the instances are indexed by $k = 1, 2, \ldots, K_A$. Inspired by Akram et al. (2016), we assume a linear state-space model on the logit-probability of attending to speaker 1. We define the binary random variable $n_k = 1$ when speaker 1 is attended and $n_k = 2$ when speaker 2 is attended, at instance k. The goal is to obtain estimates of $p_k := \mathrm{P}(n_k = 1)$ together with its confidence intervals for $1 \leq k \leq K_A$. The state dynamics are given by:

$$\begin{cases} p_k = \mathrm{P}(n_k = 1) = 1 - \mathrm{P}(n_k = 2) = \frac{1}{1+\exp(-z_k)} \\ z_k = c_0 z_{k-1} + w_k \\ w_k \sim \mathcal{N}(0, \eta_k) \\ \eta_k \sim \text{Inverse-Gamma}(a_0, b_0) \end{cases} \tag{2}$$

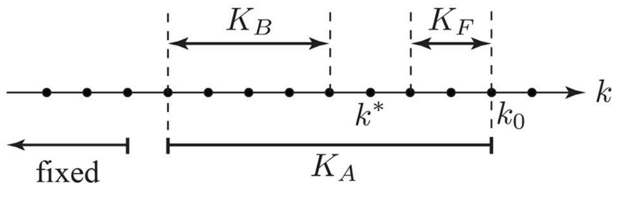


FIGURE 2 | The parameters involved in state-space fixed-lag smoothing.

The dynamics of the main latent variable z_k are controlled by its transition scale c_0 and state variance η_k. The hyperparameter $0 \leq c_0 \leq 1$ ensures the stability of the updates for z_k. The state variance η_k is modeled using an Inverse-Gamma conjugate prior with hyper-parameters a_0 and b_0. The log-prior of the Inverse-Gamma density takes the form $\ln \mathrm{P}(\eta_k) = -(a_0 + 1)\ln \eta_k - \frac{b_0}{\eta_k} + C$ for $\eta_k > 0$, where C is a normalization constant. By choosing a_0 greater than and sufficiently close to 2, the variance of the Inverse-Gamma distribution takes large values and therefore can serve as a non-informative conjugate prior. Considering the fact that we do not expect the attentional state to have high fluctuations within a small window of time, we can further tune the hyperparameters a_0 and b_0 for the prior to promote smaller values of η_k's. This way, we can avoid large consecutive fluctuations of the z_k's, and consequently the p_k's.

Next, we develop an observation model relating the state dynamics of Equation (2) to the observations $m_k^{(1)}$ and $m_k^{(2)}$ for $k = 1, 2, \ldots, K_A$. To this end, we use the latent variable n_k as the link between the states and observations:

$$\begin{cases} \begin{cases} m_k^{(i)} \,\Big|\, n_k = i \sim \text{Log-Normal}\left(\rho^{(a)}, \mu^{(a)}\right) \\ m_k^{(i)} \,\Big|\, n_k \neq i \sim \text{Log-Normal}\left(\rho^{(u)}, \mu^{(u)}\right) \end{cases}, \quad i = 1, 2 \\ \rho^{(a)} \sim \text{Gamma}\left(\alpha_0^{(a)}, \beta_0^{(a)}\right), \quad \mu^{(a)} \,\Big|\, \rho^{(a)} \sim \mathcal{N}\left(\mu_0^{(a)}, \rho^{(a)}\right) \\ \rho^{(u)} \sim \text{Gamma}\left(\alpha_0^{(u)}, \beta_0^{(u)}\right), \quad \mu^{(u)} \,\Big|\, \rho^{(u)} \sim \mathcal{N}\left(\mu_0^{(u)}, \rho^{(u)}\right) \end{cases} \tag{3}$$

When speaker $i = 1, 2$ is attended to, we use a Log-Normal distribution on $m_k^{(i)}$'s, with log-density given by $\ln \mathrm{P}\left(m_k^{(i)} \,\big|\, n_k = i\right) = -\ln m_k^{(i)} + \frac{1}{2}\ln \rho^{(a)} - \frac{\rho^{(a)}}{2}\left(\ln m_k^{(i)} - \mu^{(a)}\right)^2 + C^{(i)}$, where $\mu^{(a)} \in \mathbb{R}$, $\rho^{(a)} \in \mathbb{R}_{>0}$, and $C^{(i)}$ is a normalization constant, for $i = 1, 2$, and $k = 1, 2, \ldots, K_A$. Similarly, when speaker $i = 1, 2$ is *not* attended to, we use a Log-Normal distribution on $m_k^{(i)}$ with parameters $\rho^{(u)}$ and $\mu^{(u)}$. As mentioned before, choosing an appropriate attention marker results in a statistical separation between $m_k^{(1)}$ and $m_k^{(2)}$, if only one speaker is attended. The Log-Normal distribution is a unimodal distribution on $\mathbb{R}_{>0}$ which lets us capture this concentration in the values of $m_k^{(i)}$'s. In contrast to Akram et al. (2016), this distribution also leads to closed form update rules, which significantly reduces computational costs. We have also imposed conjugate priors on the joint distribution of (ρ, μ)'s, which factorizes as $\ln \mathrm{P}(\rho, \mu) = \ln \mathrm{P}(\rho) + \ln \mathrm{P}(\mu \mid \rho)$. The hyperparameters α_0, β_0, and μ_0 serve to tune the attended and the unattended Log-Normal distributions to create separation between the attended and unattended cases. These hyperparameters can be determined based on the mean and variance information of $m_k^{(i)}$'s in a supervised manner, in which the attended speaker labels are known, while enforcing large enough variances for the priors not to be too restrictive in estimating the Log-Normal distribution parameters. As will be discussed in our simulation and real-data analysis, this tuning step can be performed using a minimal amount of labeled data, which is significantly less than those required for reliable pre-estimation of encoder/decoder coefficients in existing approaches.

The parameters of the state-space model are therefore $\boldsymbol{\Omega} = \left\{z_{1:K_A}, \eta_{1:K_A}, \rho^{(a)}, \mu^{(a)}, \rho^{(u)}, \mu^{(u)}\right\}$, which have to be inferred from $m_{1:K_A}^{(1)}$ and $m_{1:K_A}^{(2)}$. As mentioned before, our goal in the fixed-lag smoothing approach is to estimate $p_{k^*} = 1/\left(1 + \exp(-z_{k^*})\right)$ as well as its confidence intervals in each window, where $k^* = K_A - K_F$. However, in order to do so in our model, we perform the inference step over all the parameters in $\boldsymbol{\Omega}$ and output the estimate of $z_{k^*} \in \boldsymbol{\Omega}$ and its confidence intervals. The calculation of confidence intervals is discussed in detail at the end of section 2 of the Supplementary Material. In short, the density of each z_k given the set of observed attention markers, estimated variances, and estimated Log-Normal distribution parameters is recursively approximated by a Gaussian density. Then, the mean of this Gaussian approximation is reported as the estimated z_k and its confidence intervals are determined based on the corresponding variance. The estimated $\boldsymbol{\Omega}$ would then serve as the initialization for parameter estimation in the next window. The parameters in $\boldsymbol{\Omega}$ can be inferred through two nested EM algorithms as in Akram et al. (2016). In section 2 of the Supplementary Material, we have given a detailed derivation of the EM framework and update rules in the real-time setting, as well as solutions to further reduce the computational costs thereof. From here on, we refer to the output of the introduced framework, which operates with the discussed built-in delay, as the *real-time (state-space) estimator*. In section 3.1, we compare the performance of the real-time estimator against that of the *batch-mode (state-space) estimator*. We define the batch-mode estimator as applying the state-space model in Equations (2) and (3) on all the computed attention markers in a trial *at once*, i.e., $K_A = K$, rather than in a fixed-lag sliding window fashion. In other words, the batch-mode estimator observes all the attention marker samples in a trial, i.e., $m_k^{(i)}$ for $i = 1, 2$ and $k = 1, \ldots, K$, and then infers the attention probabilities. In this sense, it is similar to the state-space estimator used in Akram et al. (2016). The batch-mode estimator provides a robust estimate of the attentional state at any instance by having access to all the future and past attention markers. Thus, it can serve as a performance benchmark for tuning the fixed-lag sliding window hyperparameters in the real-time estimator. We will further discuss this point in section 3.1.4.

Remark 5. The state-space models given in Equations (2) and (3) have two major differences with the one used in Akram et al. (2016). First, in Akram et al. (2016), the distribution over the correlative measure for the *unattended* speaker is assumed to be uniform. However, this assumption may not hold for other attention markers in general. For instance, the M100 magnitude of the TRF estimated from MEG data is a positive random variable, which is concentrated on higher values for the attended speaker compared to the unattended speaker. In order to address this issue, we consider a parametric distribution in Equation (3) over the attention marker corresponding to the unattended speaker and infer its parameters from the data. If this distribution is indeed uniform and non-informative, the variance of the unattended distribution, which is estimated from the data, would be large enough to capture the flatness of the distribution. Second, the parametrization of the observations using Log-Normal densities and their corresponding priors factorized using Gamma and Gaussian priors, admits fast and closed-form update equations in the real-time setting. As we have shown in section 2 of the Supplementary Material, these models also have the advantage of incorporating low-complexity updates by simplifying the EM procedure. In addition, the Log-Normal distribution as a generic unimodal distribution allows us to model a larger class of attention markers.

Remark 6. As mentioned in section 1, one limitation of existing approaches based on reverse-correlation is that their decoding accuracy drops significantly when operating at high temporal resolutions. The major source for this performance deterioration is the stochastic fluctuations and uncertainties in correlation values when computed over small windows of length ~ 1 s. Therefore, when enough training data is available for reliable pre-estimation of decoders/encoders, our real-time state-space module can be added as a complementary final step to the foregoing approaches in order to correct for the stochastic fluctuations in the calculated correlation values.

2.4. EEG Recording and Experiment Specifications

Sixty four-channel EEG was recorded using the actiCHamp system (Brain Vision LLC, Morrisville, NC, US) and active EEG electrodes with Cz channel being the reference. The data was digitized at a 10 kHz sampling frequency. Insert earphones ER-2 (Etymotic Research Inc., Elk Grove Village, IL, US) were used to deliver sound to the subjects while sitting in a sound-attenuated booth. The earphones were driven by the clinical audiometer Piano (Inventis SRL, Padova, Italy), and the volume was adjusted for every subject's right and left ears separately until the loudness in both ears was matched at a comfortably loud listening level. Three normal-hearing adults participated in the study. The mean age of subjects was 49.5 years with the standard deviation of 7.18 years. The study included a constant-attention experiment, where the subjects were asked to sit in front of a computer screen and restrict motion while any audio was playing. The data used in this paper corresponds to 3 subjects, 24 trials each.

The stimulus set contained eight story segments, each approximately 10 min long. Four segments were narrated by male speaker 1 (M1) and the other four by male speaker 2 (M2). The stimuli were presented to the subjects in a dichotic fashion, where the stories read by M1 were played in the left ear, and stories read by M2 were played in the right ear for all the subjects. Each subject listened to 24 trials of the dichotic stimulus. Each trial had a duration of approximately 1 min, and for each subject, no storyline was repeated in more than one trial. During each trial, the participants were instructed to look at an arrow at the center of the screen, which determined whether to attend to the right-ear story or to the left one. The arrow remained fixed for the duration of each trial, making it a constant-attention experiment. At the end of each trial, two multiple choice semantic questions about the attended story were displayed on the screen to keep the subjects alert. The responses of the subjects as well as their reaction time were recorded as a behavioral measure of the subjects' level of attention, and above eighty percent of the questions were answered correctly by each subject. Breaks and snacks were given between stories if requested. All the audio recordings, corresponding questions, and transcripts were obtained from a collection of stories recorded at Hafter Auditory Perception Lab at UC Berkeley.

2.5. MEG Recording and Experiment Specifications

MEG signals were recorded with a sampling rate of 1 kHz using a 160-channel whole-head system (Kanazawa Institute of Technology, Kanazawa, Japan) in a dimly lit magnetically shielded room (Vacuumschmelze GmbH & Co. KG, Hanau, Germany). Detection coils were arranged in a uniform array on a helmet-shaped surface on the bottom of the dewar with 25 mm between the centers of two adjacent 15.5 mm diameter coils. The sensors are first-order axial gradiometers with a baseline of 50 mm, resulting in field sensitivities of $5\,\frac{\mathrm{fT}}{\sqrt{\mathrm{Hz}}}$ or better in the white noise region.

The two speech signals were presented at 65 dB SPL using the software package Presentation (Neurobehavioral Systems Inc., Berkeley, CA, US). The stimuli were delivered to the subjects' ears with 50 Ω sound tubing (E-A-RTONE 3A; Etymotic Research), attached to E-A-RLINK foam plugs inserted into the ear canal. Also, the whole acoustic delivery system was equalized to give an approximately flat transfer function from 40 to $3,000$ Hz. A 200 Hz low-pass filter and a notch filter at 60 Hz were applied to the magnetic signal in an online fashion for noise removal. Three of the 160 channels are magnetometers separated from the others and used as reference channels. Finally, to quantify the head movement, five electromagnetic coils were used to measure each subject's head position inside the MEG machine once before and once after the experiment.

Nine normal-hearing, right-handed young adults (ages between 20 and 31) participated in this study. The study includes two sets of experiments: the constant-attention experiment and the attention-switch experiment, in each of which six subjects participated. Three subjects took part in both of the experiments. The experimental procedure were approved by the University of Maryland Institutional Review Board (IRB), and written informed consent was obtained from each subject before the experiment.

The stimuli included four non-overlapping segments from the book *A Child's History of England* by Charles Dickens. Two of the segments were narrated by a man and the other two by a woman. Three different mixtures, each 60 s long, were generated and used in the experiments to prevent reduction in the attentional focus of the subjects. Each mixture included a segment narrated by the male speaker and one narrated the the female speaker. In all trials, the stimuli were delivered diotically to both ears using tube phones inserted into the ear canals at a level of approximately 65 dB SPL. The constant-attention experiment consisted of two conditions: (1) attending to the male speaker in the first mixture,

(2) attending to the female speaker in the second mixture. In the attention-switch experiment, subjects were instructed to focus on the female speaker in the first 28 s of the trial, switch their attention to the male speaker after hearing a 2 s pause (28th to 30th s), and maintain their focus on the latter speaker through the end of the trial. Each mixture was repeated three times in the experiments, resulting in six trials per speaker for the constant-attention experiment and three trials per speaker for the attention-switch experiment. After the presentation of each mixture, subjects answered comprehensive questions related to the segment they were instructed to focus on, as a way to keep them motivated to attend to the target speaker. Eighty percent of the questions were answered correctly on average. Furthermore, a preliminary experiment for each of the nine participating subjects was performed prior to the main experiments. In this study, the subjects listened to a single speech stream, first segment in the stimuli set narrated by the male speaker, for three trials each 60 s long. The MEG recordings from the preliminary experiment were used to calculate the subject-specific linear combination of MEG channels which forms the auditory component of the response, as will be explained next. Note that for each subject, all the recordings were performed in a single session resulting in a minimal change of the subject's head position with respect to the MEG sensors.

3. RESULTS

In this section, we apply our real-time attention decoding framework to synthetic data as well as M/EEG recordings. Section 3.1 includes the simulation results, and Sections 3.2 and 3.3 demonstrate the results for the analysis of EEG and MEG recordings, respectively.

3.1. Simulations

In order to validate our proposed framework, we perform two sets of simulations. The first simulation pertains to our EEG analysis and employs a decoding model, which we describe below in full detail. The second simulation, for our MEG analysis using an encoding model, is deferred to the Supplementary Material section 4, in the interest of space.

3.1.1. Simulation Settings

In order to simulate EEG data under a dual-speaker condition, we use the following generative model:

$$e_t = w_t^{(1)}\left(s_t^{(1)} * h_t\right) + w_t^{(2)}\left(s_t^{(2)} * h_t\right) + \mu + u_t \tag{4}$$

where $s_t^{(1)}$ and $s_t^{(2)}$ are respectively the speech envelopes of speakers 1 and 2 at time t; the output e_t is the simulated neural response, which denotes an auditory component of the EEG or the EEG response at a given channel at time t for $t = 1, 2, \ldots, T$. Motivated by the analysis of LTI systems, h_t can be considered as the impulse response of the neural process resulting in e_t, and $*$ represents the convolution operator; the scalar μ is an unknown constant mean, and u_t denotes a zero-mean i.i.d Gaussian noise. The weight functions $w_t^{(1)}$ and $w_t^{(2)}$ are signals modulated by the attentional state which determine the contributions of speakers 1 and 2 to e_t, respectively. In order to simulate the attention modulation effect, we assume that when speaker 1 (resp. 2) is attended to at time t, we have $w_t^{(1)} > w_t^{(2)}$ (resp. $w_t^{(1)} < w_t^{(2)}$).

We have chosen two 60 s-long speech segments from those used in the MEG experiment (see section 2.5) and calculated $s_t^{(1)}$ and $s_t^{(2)}$ as their envelopes for a sampling rate of $f_s = 200$ Hz. Also, we have set $\mu = 0.02$ and $u_t \overset{\text{iid}}{\sim} \mathcal{N}(0, 2.5 \times 10^{-5})$ in Equation (4). **Figure 3A** shows the location and amplitude of the lag components in the impulse response, which is then smoothed using a Gaussian kernel with standard deviation of 10 ms to result in the final impulse response h_t, shown in **Figure 3B**. The significant components of h_t are chosen at 50 ms and 100 ms lags, with a few smaller components at higher latencies (Akram et al., 2016). It is noteworthy that existing results (Ding and Simon, 2012a; Power et al., 2012; Akram et al., 2017) suggest that this impulse response (i.e., the TRF) is not the same for the attended and unattended speakers, as discussed in section 2.2. However, we have considered the same h_t for both speakers in this simulation for simplicity, given that our focus here is to model the stronger presence of the attended speaker in the neural response in terms of the extracted attention markers. In section 4 of the Supplementary Material, we indeed use an encoding model consisting of different and attention-modulated TRFs for the two speakers. The weight signals $w_t^{(1)}$ and $w_t^{(2)}$ in Equation (4) are chosen to favor speaker 1 in the (0 s, 30 s) interval and speaker 2 in the (30 s, 60 s) interval.

3.1.2. Parameter Selection

We aim at estimating decoders in this simulation, which linearly map e_t and its lags to $s_t^{(1)}$ and $s_t^{(2)}$. To estimate the decoders, we have considered consecutive non-overlapping windows of length 0.25 s resulting in $K = 240$ windows of length $W = 50$ samples. Also, we have chosen $\gamma = 0.001$, through cross-validation, and $\lambda = 0.95$ in estimating the decoding coefficients, which results in an *effective* data length of 5 s for decoder estimation. The forward lags of the neural response have been limited to a 0.4 s window, i.e., $L_d = 80$ samples. Given that the decoder corresponds to the inverse of a smooth kernel h_t, it may not have the same smoothness properties of h_t. Hence, we do not employ a smooth basis for decoder estimation. We have used the FASTA package (Goldstein et al., 2014) with Nesterov's acceleration method to implement the forward-backward splitting algorithm for encoder/decoder estimation.

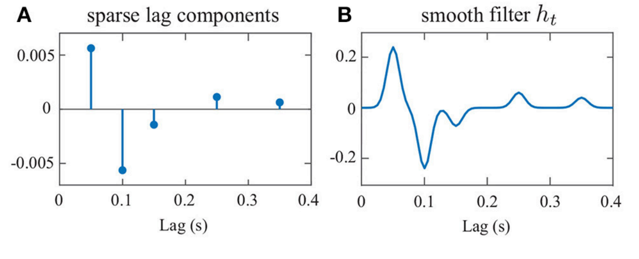


FIGURE 3 | Impulse response h_t used in Equation (4). **(A)** sparse lag components, **(B)** the smooth impulse response.

As for the state-space model estimators, we have considered 20 (inner and outer) EM iterations for the batch-mode estimators, while for the real-time estimators, we use 1 inner EM iteration and 20 outer EM iterations (see section 2 of the Supplementary Material for more details).

There are three criteria for choosing the fixed-lag smoothing parameters: First, how close to the true real-time analysis the system operates is determined by K_F. Second, the computational cost of the system is determined by K_A. Third, how close the output of the system is to that of the batch-mode estimator is determined by both K_F and K_A. These three criteria form a tradeoff in tuning the parameters K_A and K_F. Specific choices of these parameters are given in the next subsection.

For tuning the hyperparameters of the priors on the attended and unattended distributions, we have used a separate 15 s sample trial generated from the same simulation model in Equation (4) for each of the three cases. The parameters $\left(\alpha_0^{(a)}, \alpha_0^{(u)}, \beta_0^{(a)}, \beta_0^{(u)}, \mu_0^{(a)}, \mu_0^{(u)}\right)$ have been chosen by fitting the Log-Normal distributions to the attention marker outputs from the sample trials in a supervised manner (with known attentional state). The variance of the Gamma priors $\frac{\alpha_0^{(a)}}{\beta_0^{(a)2}}$ and $\frac{\alpha_0^{(u)}}{\beta_0^{(u)2}}$ have been chosen large enough such that the priors are non-informative. This step can be thought of as the initialization of the algorithms prior to data analysis. For the Inverse-Gamma prior on the state-space variances, we have chosen $a_0 = 2.008$ and $b_0 = 0.2016$, resulting in a mean of 0.2 and a variance of 5. This prior favors small values of η_k's to ensure that the state estimates are immune to large fluctuations of the attention markers, while the large variance (compared to the mean) results in a non-informative prior for smaller values of η_k's.

3.1.3. Estimation Results

Figure 4 shows the results of our estimation framework for a correlation-based attention marker. Row A in **Figure 4** shows three cases considered for modulating the weights $w_t^{(1)}$ and $w_t^{(2)}$, where the weights are contaminated with Gaussian noise $\mathcal{N}(0, 4 \times 10^{-4})$ to model extra uncertainties in determining the contribution of each speech to the neural response, arising from irrelevant or background neural processes. In order to probe the transition delay of the state-space estimates due to abrupt changes in the attentional state, the two weight vectors undergo step-like transition at 30 s. Cases 1, 2, and 3 exhibit increasing levels of difficulty in discriminating the contributions of the two speakers to the neural response. Rows B and C in **Figure 4** respectively show the decoder estimates for speakers 1 and 2. As expected, the significant components of the decoders around 50, 100, and 150 ms lags, are modulated by the attentional state, and the modulation effect weakens as we move from Case 1 to 3. In Case 1, these components are less significant overall for the decoder estimates of speaker 2 in the [0 s, 30 s] time interval and become larger as the attention switches to speaker 2 during the rest of the trial (red boxes in row C of Case 1). On the other hand, in Case 3, the magnitude of said components does not change notably across the 30 s mark. The TRF h_t in the forward generative model of Equation (4) is an FIR filter with significant components at lags which are multiples of 0.05 s (see **Figure 3B**). Therefore, the decoder estimates in **Figure 4** correspond to truncated IIR filters, which form approximate inverse filters of the TRF. Therefore, it is expected that they comprise significant components at lags which are multiples of 0.05 s as well, but decay exponentially fast.

We have considered two different attention markers for this simulation. Row D in **Figure 4** displays the output of a correlation-based attention marker for speakers 1 and 2, which is calculated as $m_k^{(i)} = \left|\text{corr}\left(\mathbf{y}_k^{(i)}, \mathbf{X}_k \hat{\boldsymbol{\theta}}_k^{(i)}\right)\right|$ for $i = 1, 2$ and $k = 1, 2, \ldots, K$. As discussed in section 2.2, this attention marker is a measure of how well a decoder can reconstruct its target envelope. As observed in row D of **Figure 4**, the attention marker is a highly variable surrogate of the attentional state at each instance, i.e., *on average* the attention marker output for speaker 1 is higher then that of speaker 2 in the (0 s, 30 s) interval and vice versa in the (30 s, 60 s) interval. The reliability of the attention marker significantly degrades going from Case 1 to 3. This highlights the need for state-space modeling and estimation in order to optimally exploit the attention marker.

Rows E and F in **Figure 4** respectively show the batch-mode and real-time estimator outputs as the inferred attentional state probabilities $p_k = \text{P}(n_k = 1)$ for $k = 1, \ldots, K$, for the correlation-based attention marker, where colored hulls indicate 90% confidence intervals. Row F in **Figure 4** corresponds to the fixed-lag smoother, using a window of length 15 s ($K_A = \lfloor 15 f_s / W \rfloor$), and a forward-lag of 1.5 s ($K_F = \lfloor 1.5 f_s / W \rfloor$). By accounting for the lag in the decoder (L_d), the built-in delay in estimating the attentional state is 1.9 s. Note that all the relevant figures showing the outputs of the *real-time* estimator are calibrated with respect to the built-in delay for the sake of illustration. Thus, these figures must be interpreted as non-causal when $K_F > 0$, since the estimated attentional state at each time depends on the future K_F samples of the attention marker. Recall that in the batch-mode estimator, all of the attention marker outputs across the trial are available to the state-space estimator, as opposed to the fixed-lag real-time estimator which has access to a limited number of the attention markers. Therefore, the output of the batch-mode estimator (Row E) is a more robust measure of the instantaneous attentional state as compared to the real-time estimator (Row F), since it is less sensitive to the stochastic fluctuations of the attention markers in row D. For example, in the instance marked by the red arrows in rows E and F of Case 2 in **Figure 4**, the batch-mode estimator classifies the instance correctly as attending to speaker 2, while the real-time estimator cannot make an informed decision since $p_k = 0.5$ falls within the 90% confidence interval of the estimate at this instance. However, the real-time estimator exhibits performance closely matching that of the batch-mode estimator for most instances, while operating in real-time with limited data access and significantly lower computational complexity. Comparing the state-space estimators with the raw attention markers in **Figure 4D**, we observe the smoothing effect of the state-space model which makes its output robust to the stochastic fluctuations in the attention marker at high temporal resolution. Section 3 of the Supplementary

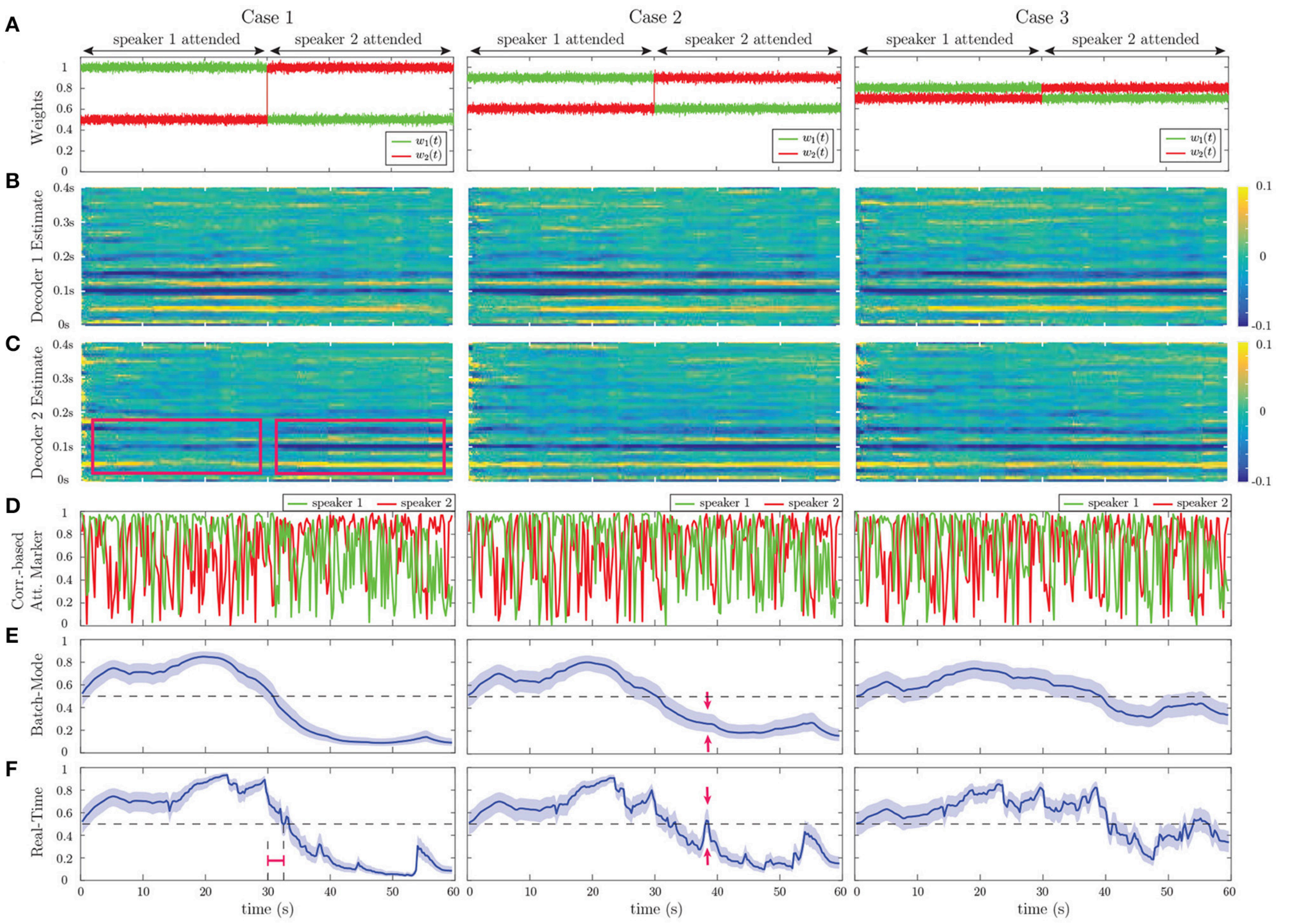


FIGURE 4 | Estimation results of application to simulated EEG data for the correlation-based attention marker: **(A)** Input weights $w_t^{(1)}$ and $w_t^{(2)}$ in Equation (4), which determine the relative effect of the two speeches on the neural response. Based on our generative model, the attention is on speaker 1 for the first half of each trial and on speaker 2 for the second half. Case 1 corresponds to a scenario where the effects of the attended and unattended speeches in the neural response are well-separated. This separation decreases as we move from Case 1 to Case 3. **(B)** Estimated decoder for speaker 1. **(C)** Estimated decoder for speaker 2. In Case 1, the significant components of the estimated decoders near the 50, 100, and 150 ms lags are notably modulated by the attentional state as highlighted by the red boxes. This effect weakens in Case 2 and visually disappears in Case 3. **(D)** Output of the correlation-based attention marker for each speaker. **(E)** Output of the batch-mode state-space estimator for the correlation-based attention marker as the estimated probability of attending to speaker 1. **(F)** Output of the real-time state-space estimator, i.e., fixed-lag smoother, for the correlation-based attention marker as the estimated probability of attending to speaker 1. The real-time estimator is not as robust as the batch-mode estimator to the stochastic fluctuations of the attention marker in row D and is more prone to misclassifications. The red arrows in rows E and F of Case 2 show that the batch-mode estimator correctly classifies the instance as attending to speaker 2, while the real-time estimator is unable to determine the attentional state.

Material includes a comparison of this smoothing effect with that of a typical Gaussian smoothing kernel applied directly to the attention markers.

Row A in **Figure 5** exhibits the output of another attention marker computed as the ℓ_1-norm of the decoder given by $m_k^{(i)} := \left\| \hat{\boldsymbol{\theta}}_k^{(i)} \right\|_1$ for $i = 1, 2$ and $k = 1, 2, \ldots, K$, where the first element of $\hat{\boldsymbol{\theta}}_k^{(i)} \in \mathbb{R}^{L_d+2}$ (the intercept parameter) is discarded in computing the ℓ_1-norm. This attention marker captures the effect of the significant peaks in the decoder. The rationale behind using the ℓ_1-norm based attention marker is the following: in the extreme case that the neural response is solely driven by the attended speech, we expect the unattended decoder coefficients to be small in magnitude and randomly distributed across the time lags. The attended decoder, however, is expected to have a sparse set of informative and significant components corresponding to the specific latencies involved in auditory processing. Thus, the ℓ_1-norm serves to distinguish between these two cases by capturing such significant components. Rows B and C in **Figure 5** show the batch-mode and real-time estimates of the attentional state probabilities for the ℓ_1-based attention marker, respectively, where colored hulls indicate 90% confidence intervals. Consistent with the results of the correlation-based attention marker (Rows E and F in **Figure 4**), the real-time estimator exhibits performance close to that of the batch-mode estimator. Comparing **Figures 4**, **5** reveals the dependence of

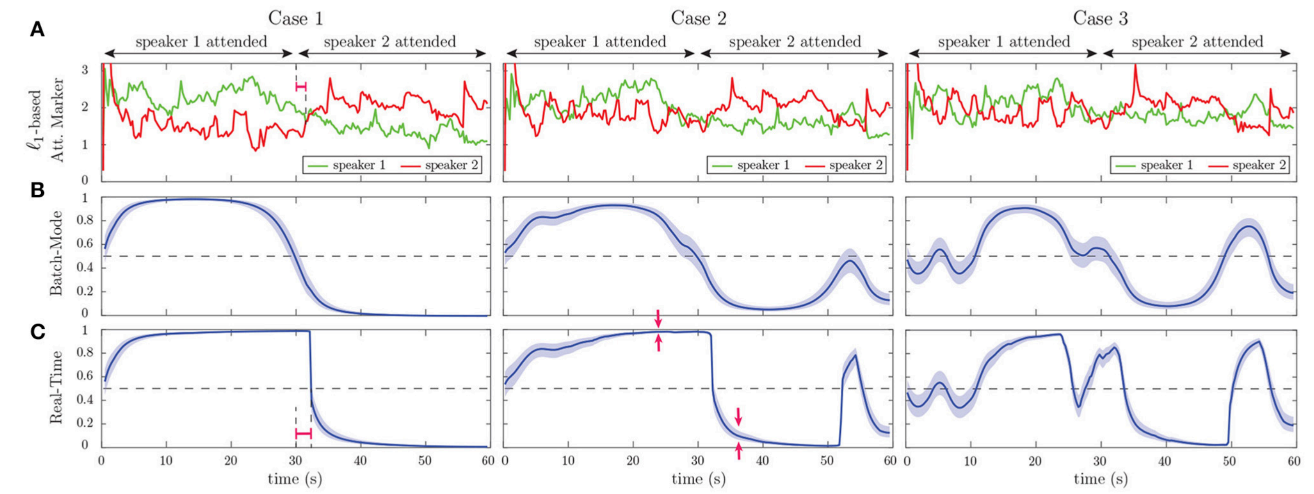


FIGURE 5 | Estimation results of application to simulated EEG data for the ℓ_1-based attention marker: **(A)** Output of the ℓ_1-based attention marker for each speaker, corresponding to the three cases in **Figure 4**. **(B)** Output of the batch-mode state-space estimator for the ℓ_1-based attention marker as the estimated probability of attending to speaker 1. **(C)** Output of the real-time state-space estimator for the ℓ_1-based attention marker as the estimated probability of attending to speaker 1. Similar to the preceding correlation-based attention marker, the classification performance degrades when moving from Case 1 (strong attention modulation) to Case 3 (weak attention modulation).

the attentional state estimation performance on the choice of the attention marker: while the correlation-based attention marker is more widely used, the ℓ_1-based attention marker provides smoother estimates of the attention probabilities, and can be used as an alternative to the correlation-based attention marker. Overall, this simulation illustrates that if the attended stimulus has a stronger presence in the neural response than the unattended one, both the correlation-based and ℓ_1-based attention markers can be attention modulated and can therefore potentially be used in real M/EEG analysis.

3.1.4. Discussion and Further Analysis

Going from Case 1 to Case 3 in **Figures 4**, **5**, we observe that the performance of all estimators degrades, causing a drop in the classification accuracy and confidence. This performance degradation is due to the declining power of the attention markers in separating the contributions of the attended and unattended speakers. However, comparing the outputs of the real-time and batch-mode estimators with their corresponding attention marker outputs in row D of **Figure 4** and row A of **Figure 5**, highlights the role of the state-space model in suppressing the stochastic fluctuations of the attention markers and thereby providing a robust and smooth measure of the attentional state.

In response to abrupt step-like changes in the attentional state, we define the *transition* delay as the time it takes for the output of the real-time estimator to reach the $p_k = 0.5$ level, which marks the point at which the classification label of the attended speaker changes. We calculate the transition delay after calibrating for the built-in delay, for all the real-time estimator outputs. Thus, the overall delay of the system in detecting abrupt attentional state changes is equal to the sum of the built-in and transition delays. The red intervals in Case 1 of row F in **Figure 4** and row C of **Figure 5** mark the transition delay of the real-time estimator corresponding to the correlation-based and ℓ_1-based attention markers, respectively. From the deflection point at 30 s, this delay is given by ∼2.3 s. The transition delay is due to the forgetting factor mechanism and the smoothing effect of the state-space estimation given the backward- and forward-lags, which have been set in place to increase the robustness of the decoding framework to stochastic fluctuations of the extracted attention markers. As a result, such classification delays in response to a sudden attention switches are expected by design. Specifically, the sole contribution of the forgetting factor mechanism to this delay can be observed as the red interval in Case 1 of row A in **Figure 5**, which precedes the application of the state-space estimation.

Comparing the batch-mode and the real-time estimators in **Figures 4**, **5**, we observe that the real-time estimators closely follow the output of the batch-mode estimators, while having access to data in an online fashion. A significant deviation between the batch-mode and real-time performance is observed in rows B and C (Cases 1 and 2) of **Figure 5** in the form of sharp drops in the real-time estimates of the attentional state probability. Given that the real-time estimator has only access to the attention marker within K_F samples in the future, the confidence intervals significantly narrow down within the first half of the trial, as all the past and near-future observations are consistent with attention to speaker 1. However, shortly after the 30 s mark, the estimator detects the change and the confidence bounds widen accordingly (see red arrows in row C of Case 2 in **Figure 5**).

In order to further quantify the performance gap between the batch-mode and real-time estimators, we define their relative

Mean Squared Error (MSE) as:

$$\text{MSE} = \frac{1}{K}\sum_{k=1}^{K}\left(\frac{1}{1+\exp\left(-\hat{z}_k^{(B)}\right)} - \frac{1}{1+\exp\left(-\hat{z}_k^{(R)}\right)}\right)^2 \qquad (5)$$

where $\hat{z}_{1:K}^{(R)}$ and $\hat{z}_{1:K}^{(B)}$ denote the real-time and batch-mode state estimates over a given trial, respectively. We have considered the logistic transformation of $\hat{z}_{1:K}^{(B)}$ and $\hat{z}_{1:K}^{(R)}$, which gives the probability of attending to speaker 1. The rationale behind this MSE metric is to measure the performance and robustness of the real-time estimator with respect to the batch-mode estimator, since they both operate on the same computed attention markers, but in different algorithmic fashions.

Figure 6 shows the effect of varying the forward-lag K_F from 0 s (i.e., fully real-time) to 5 s with 0.5 s increments for the two attention markers in Case 2 of **Figures 4**, **5**, as an example. All of the other parameters in the simulation have been fixed as before. The left panels in **Figure 6** show the MSE for different values of K_F in the real-time setting. As expected, for both attention markers, the MSE decreases as the forward-lag increases. The right panels in **Figure 6** display the incremental MSE defined as the change in MSE when K_F is increased by 0.5 s at each value, starting from $K_F = 0$. The incremental MSE is basically the discrete derivative of the displayed MSE plots and shows the amount of relative performance boost between two consecutive values of K_F, if we allow for a larger built-in delay. Notice that even a 0.5 s forward-lag significantly decreases the MSE from $K_F = 0$. The subsequent improvements of the MSE diminish as K_F is increased further. Our choice of K_F corresponding to 1.5 s in the foregoing analysis was made to maintain a reasonable tradeoff between the MSE improvement and the built-in delay in real-time operation. In summary, **Figure 6** shows that having larger forward-lags can make our estimates more robust but it creates a larger built-in delay. Whether higher levels of delay are tolerable or not depends on the particular attention decoding application.

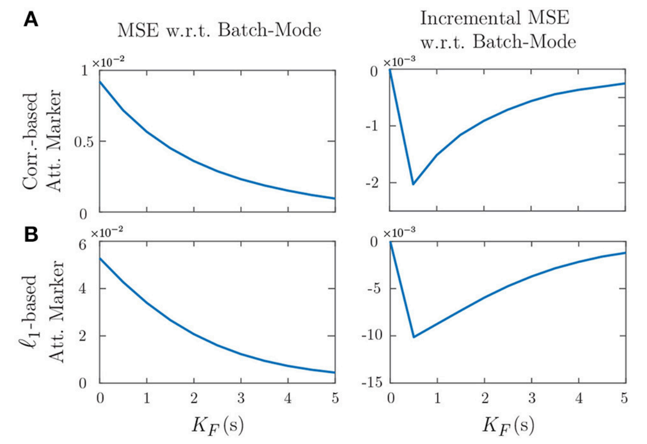


FIGURE 6 | Effect of the forward-lag K_F on the MSE for the two attention markers in case 2 of **Figures 4**, **5**. **(A)** Correlation-based attention marker, **(B)** ℓ_1-based attention marker. As the forward-lag increases, the MSE decreases, and the output of the real-time estimator becomes more similar to that of the batch-mode. This results in more robustness for the real-time estimator at the expense of more built-in delay in decoding the attentional state. The right panels show that the incremental improvement to the MSE decreases as K_F increases.

Finally, **Figure 7** shows the estimated attention probabilities and their 90% confidence intervals for the correlation-based attention marker in Case 2 of **Figure 4**, as an example of the output of the state-space estimator. The three curves correspond to the extreme values of K_F in **Figure 6** corresponding to 0 s (blue) and 5 s (red) forward-lags, and the batch-mode estimate (green). All the other parameters have been fixed as described above. The fixed-lag smoothing approach with K_F of 5 s is as robust as the batch-mode estimate. The fully real-time estimate with K_F of 0 s follows the same trend as the other two. However, it is susceptible to the stochastic fluctuations of the attention marker, which may lead to misclassifications (see the red arrows in **Figure 7**). The red interval in **Figure 7** displays the difference between the transition delays corresponding to the forward-lag of 0 s and 5 s. Although the built-in attention decoding delay of a 5 s forward-lag is more than that of 0 s by 5 s, the transition delay corresponding to the former is smaller due to observing the future attention marker samples up to 5 s. Therefore, the parameter K_F also provides a tradeoff in the overall delay of the framework in detecting abrupt attention switches, which equals the transition delay plus the built-in delay. The choice of 1.5 s for the forward-lag in our analysis was also aimed to minimize this overall delay.

3.2. Application to EEG

In this section, we apply our real-time attention decoding framework to EEG recordings in a dual-speaker environment. Details of the experimental procedures are given in section 2.4.

3.2.1. Preprocessing and Parameter Selection

Both the EEG data and the speech envelopes were downsampled to $f_s = 64$ Hz using an anti-aliasing filter. As the trials had variable

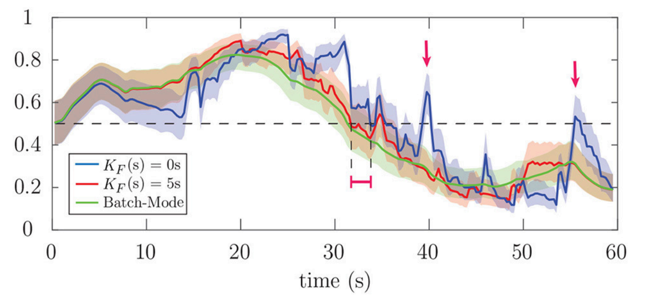


FIGURE 7 | Estimated attention probabilities together with their 90% confidence intervals for the correlation-based attention marker in Case 2 of **Figure 4**. The blue, red and green curves correspond to K_F of 0 s, K_F of 5 s, and batch-mode estimation, respectively. The estimator for K_F of 5 s is nearly as robust as the batch-mode. However, the fully real-time estimator with K_F of 0 s is sensitive to the stochastic fluctuations of the attention markers, which results in the misclassification of the attentional state at the instances marked by red arrows.

lengths, we have considered the first 53 s of each trial for analysis. We have considered consecutive windows of length 0.25 s for decoder estimation, resulting in $W = 16$ samples per window and $K = 212$ instances for each trial. Also, we have considered lags up to 0.25 s for decoder estimation, i.e., $L_d = 16$. The latter is motivated by the results of O'Sullivan et al. (2015) suggesting that the most relevant decoder components are within the first 0.25 s lags. Prior studies have argued that the effects of auditory attention and speech perception are strongest in the frontal and close-to-ear EEG electrodes (Kähkönen et al., 2001; Power et al., 2012; Bleichner et al., 2016; Khalighinejad et al., 2017). We have only considered 28 EEG channels in the decoder estimation problem, i.e., $C = 28$, including the frontal channels Fz, F1-F8, FCz, FC1-FC6, FT7-FT10, C1-C6, and the T complex channels T7 and T8. This subsampling of the electrodes is inspired by the results in Mirkovic et al. (2015), which show that using an electrode subset of the same size for decoding results in nearly the same classification performance as in the case of using all the electrodes. Note that for our real-time setting, a channel selection step can considerably decrease the computational cost and the dimensionality of the decoder estimation step, given that a vector of size $1+C(L_d+1)$ needs to be updated within each 0.25 s window.

We have determined the regularization coefficient $\gamma = 0.4$ via cross-validation and the forgetting factor $\lambda = 0.975$, which results in an *effective* data length of 10 s in the estimation of the decoder and is long enough for stable estimation of the decoding coefficients. It is worth noting that small values of λ, and hence small effective data lengths, may result in an under-determined inverse problem, since the dimension of the decoder is given by $1 + C(L_d + 1)$. Finally, in the FASTA package, we have used a tolerance of 0.01 together with Nesterov's accelerated gradient descent method to ensure that the processing can be done in an online fashion.

In studies involving correlation-based measures, such as O'Sullivan et al. (2015) and Akram et al. (2016), the convention is to train attended and unattended decoders/encoders using multiple trials and then use them to calculate the correlation measures over the test trials. The correlation-based attention marker, however, did not produce a statistically significant segregation of the attended and the unattended speakers in our analysis. This discrepancy seems to stem from the fact that the estimated encoders/decoders and the resulting correlations in the aforementioned studies are more informative and robust due to the use of batch-mode analysis with multiple trials for decoder estimation, as compared to our real-time framework. The ℓ_1-based attention marker, however, resulted in a meaningful statistical separation between the attended and the unattended speakers. Therefore, in what follows, we present our EEG analysis results using the ℓ_1-based attention marker.

The parameters of the state-space models have been set similar to those used in simulations, i.e., $K_A = \lfloor 15f_s/W \rfloor$, $K_F = \lfloor 1.5f_s/W \rfloor$, $a_0 = 2.008$, $b_0 = 0.2016$. Considering the 0.25 s lag in the decoder model, the built-in delay in estimating the attentional state for the real-time system is 1.75 s. For estimating the prior distribution parameters for each subject, we use the first 15s of each trial. As mentioned before, considering the 15 s-long sliding window, we can treat the first 15 s of each trial as a tuning step in which the prior parameters are estimated in a supervised manner and the state-space model parameters are initialized with the values estimated using these initial windows. Thus, similar to the simulations, $\left(\alpha_0^{(a)}, \alpha_0^{(u)}, \beta_0^{(a)}, \beta_0^{(u)}, \mu_0^{(a)}, \mu_0^{(u)}\right)$ for each subject have been set according to the parameters of the two fitted Log-Normal distributions on the ℓ_1-norm of the decoders in the first 15 s of the trials, while choosing large variances for the priors to be non-informative.

3.2.2. Estimation Results

Figure 8 shows the results of applying our proposed framework to EEG data. For graphical convenience, the data have been rearranged so that speaker 1 is always attended. The left, middle and right panels correspond to subjects 1, 2, and 3, respectively. For each subject, three example trials have been displayed in rows A, B, and C. Row A includes trials in which the attention marker clearly separates the attended and unattended speakers, while Row C contains trials in which the attention marker fails to do so. Row B displays trials in which on average the ℓ_1-norm of the estimated decoder is larger for the attended speaker; however, occasionally, the attention marker fails to capture the attended speaker.

Consistent with our simulations, the real-time estimates (third graphs in rows A, B, and C) generally follow the output of the batch-mode estimates (second graphs in rows A, B, and C). However, the batch-mode estimates yield smoother transitions and larger confidence intervals in general, both of which are due to having access to future observations.

Figure 9 shows the effect of forward-lag K_F on the performance of real-time estimates, similar to that shown in **Figure 6** for the simulations. The forward-lag K_F is increased from 0 s to 5 s with 0.5 s increments while all the other parameters of the EEG analysis remain the same. The MSE in **Figure 9** has been averaged over all trials for each subject. As we observe in the incremental MSE plot, even a 0.5 s lag can significantly decrease the MSE from the case of 0 s forward-lag (corresponding to the fully real-time setting). Similar to the simulations, we have chosen a K_F of 1.5 s for the EEG analysis, since the incremental MSE improvements are significant at this lag, and this choice results in a tolerable built-in delay for real-time applications.

Finally, **Figure 10** summarizes the *real-time* classification results of our EEG analysis at the group level, in order to present subject-specific and individual trial performances. **Figure 10A** shows a cartoon of the estimated attention probabilities for a generic trial in order to illustrate the classification conventions. We define an instance (i.e., one of the K consecutive windows of length W samples) to be correctly (incorrectly) classified if the estimated attentional state probability together with its 90% confidence intervals lie above (below) 0.5. If the 90% confidence interval at an instance includes the 0.5 attention probability line, we do not classify it as either correct or incorrect. **Figure 10B** displays the correctly classified instances (y-axis) vs. those incorrectly classified (x-axis) for each trial. The subjects are color-coded and each circle corresponds to one trial. The average classification results over all trials for each subject are shown in **Figure 10C**. In summary, our framework provides ∼80% average

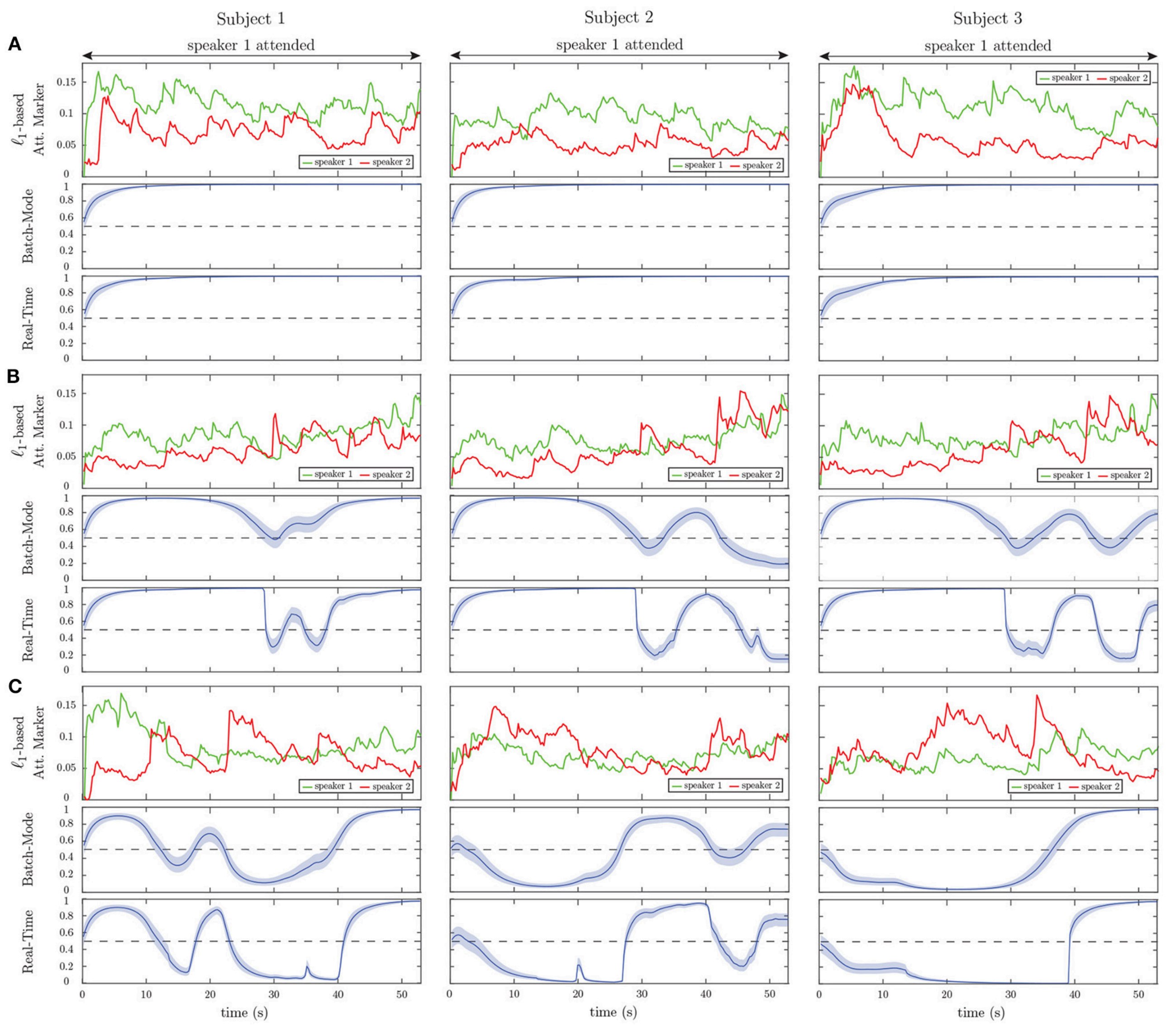


FIGURE 8 | Examples of the ℓ_1-based attention markers (first panels), batch-mode (second panels), and real-time (third panels) state-space estimation results for nine selected EEG trials. **(A)** Representative trials in which the attention marker reliably separates the attended and unattended speakers. **(B)** Representative trials in which the attention marker separates the attended and unattended speakers on average over the trial. **(C)** Representative trials in which the attention marker either does not separate the two speakers or results in a larger output for the unattended speaker.

hit rate and ∼15% average false-alarm per trial per subject. The group-level hit rate and false alarm rate are respectively given by 79.63 and 14.84%.

3.3. Application to MEG

In this section, we apply our real-time attention decoding framework to MEG recordings of multiple subjects in a dual-speaker environment. The MEG experimental procedures are discussed in section 2.5.

3.3.1. Preprocessing and Parameter Selection

The recorded MEG responses were band-pass filtered between 1 and 8 Hz (delta and theta bands), corresponding to the slow temporal modulations in speech (Ding and Simon, 2012a,b), and downsampled to 200 Hz. MEG recordings, like EEG, include both the stimulus-driven response as well as the background neural activity, which is irrelevant to the stimulus. For the encoding model used in our analysis, we need to extract the stimulus-driven portion of the response, namely the auditory component. In Särelä and Valpola (2005) and de Cheveigné and Simon (2008), a blind source separation algorithm called the Denoising Source Separation (DSS) is described which decomposes the data into temporally uncorrelated components ordered according to their trial-to-trial phase-locking reliability. In doing so, DSS only requires the responses in different trials and not the stimuli. Similar to Akram et al. (2016, 2017), we only use the first DSS

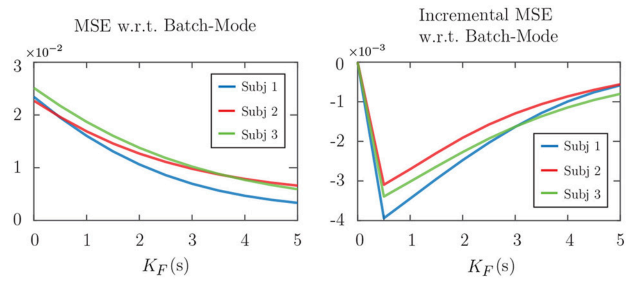


FIGURE 9 | Effect of the forward-lag K_F on MSE in application to real EEG data. The left panel shows the MSE with respect to the batch-mode output averaged over all the trials for each subject. The right panel displays the incremental MSE at each lag, from K_F of 0 s to K_F of 5 s with 0.5 s increments.

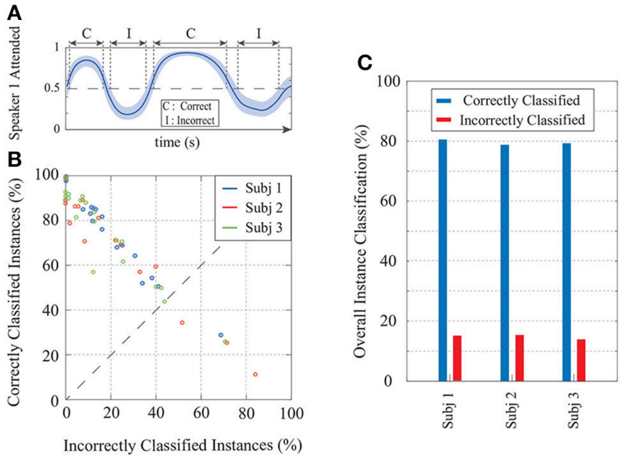


FIGURE 10 | Summary of the real-time classification results in application to real EEG data. **(A)** A generic example of the state-space output for a trial illustrating the classification conventions. **(B)** Classification results per trial for all subjects; each circle corresponds to a trial and the subjects are color-coded. The trials falling below the dashed line have more incorrectly classified instances than correctly classified ones. **(C)** Average classification performance over all trials for the three subjects.

component as the auditory component, since it tends to capture a significant amount of stimulus information and to produce a bilateral stereotypical auditory field pattern.

Since DSS is an *offline* algorithm operating on all the data at once, we cannot readily use it for real-time attention decoding. Instead, we apply DSS to the data from preliminary trials from each subject in order to calculate the *subject-specific* linear combination of the MEG channels that compose the first DSS component. We then use these channel weights to extract the MEG auditory responses during the constant-attention and attention-switch experiments in a real-time fashion. Note that the MEG sensors are not fixed with respect to the head position across subjects and are densely distributed in space. Therefore, it is not reasonable to use the same MEG channel weights for all subjects. The preliminary trials for each subject can thus serve as a training and tuning step prior to the application of our proposed attention decoding framework.

The MEG auditory component extracted using DSS is used as E_t in our encoding model. Similar to our foregoing EEG analysis, we have considered consecutive windows of length 0.25 s resulting in $W = 50$ samples per window and a total number of $K = 240$ instances, at a sampling frequency of 200 Hz. The TRF length, or the total encoder lag, has been set to 0.4 s resulting in $L_e = 80$ in order to include the most significant TRF components (Ding and Simon, 2012a). The ℓ_1-regularization parameter γ in Equation (1) has been adjusted to 1 through two-fold cross-validation, and we have chosen a forgetting factor of $\lambda = 0.975$, resulting in an *effective* data length of 10 s, long enough to ensure estimation stability.

As for the encoder model, we have used a Gaussian dictionary $\mathbf{G_0}$ to enforce smoothness in the TRF estimates. The columns of $\mathbf{G}_0$ consist of overlapping Gaussian kernels with the standard deviation of 20 ms whose means cover the 0 s to 0.4 s lag range with $T_s = 5$ ms increments. The 20 ms standard deviation is consistent with the average full width at half maximum (FWHM) of an auditory MEG evoked response (M50 or M100), empirically obtained from MEG studies (Akram et al., 2017). Thus, the overall dictionary discussed in Remark 2 takes the form $\mathbf{G} = \text{diag}(1, \mathbf{G_0}, \mathbf{G_0})$. Also, similar to Akram et al. (2017), we have used the logarithm of the speech envelopes as the regression covariates. Finally, the parameters of the FASTA package in encoder estimation have been chosen similar to those in the foregoing EEG analysis.

The M100 component of the TRF has shown to be larger for the attended speaker than the unattended speaker (Ding and Simon, 2012a; Akram et al., 2017). Thus, at each instance k, we extract the magnitude of the negative peak close to the 0.1 s delay in the real-time TRF estimate of each speaker as the attention markers $m_k^{(1)}$ and $m_k^{(2)}$. For the state-space model and the fixed-lag window, we have used the same configuration as in our foregoing EEG analysis, i.e., $K_A = \lfloor 15 f_s/W \rfloor$, $K_F = \lfloor 1.5 f_s/W \rfloor$, $a_0 = 2.008$, and $b_0 = 0.2016$. Note that the built-in delay in estimating the attentional state is now only 1.5 s, given that we use an encoding model for our MEG analysis. Furthermore, the prior distribution parameters for each subject were chosen according to the two fitted Log-Normal distributions on the extracted M100 values in the first 15 s of the trials, while choosing large variances for the Gamma priors to be non-informative. Similar to the preceding cases, the first 15 s of each trial can be thought of as an initialization stage.

3.3.2. Estimation Results

Figure 11 shows our estimation results for four sample trials from the constant-attention (cases 1 and 2) and attention-switch (cases 3 and 4) experiments. For graphical convenience, we have rearranged the MEG data such that in the constant-attention experiment, the attention is always on speaker 1, and in the attention-switch experiment, speaker 1 is attended from 0 to 28 s. Cases 1 and 3 corresponds to trials in which the extracted M100 values for the attended speaker are more significant than those of the unattended speaker during most of the trial duration. Cases 2 and 4, on the other hand, correspond to trials in which

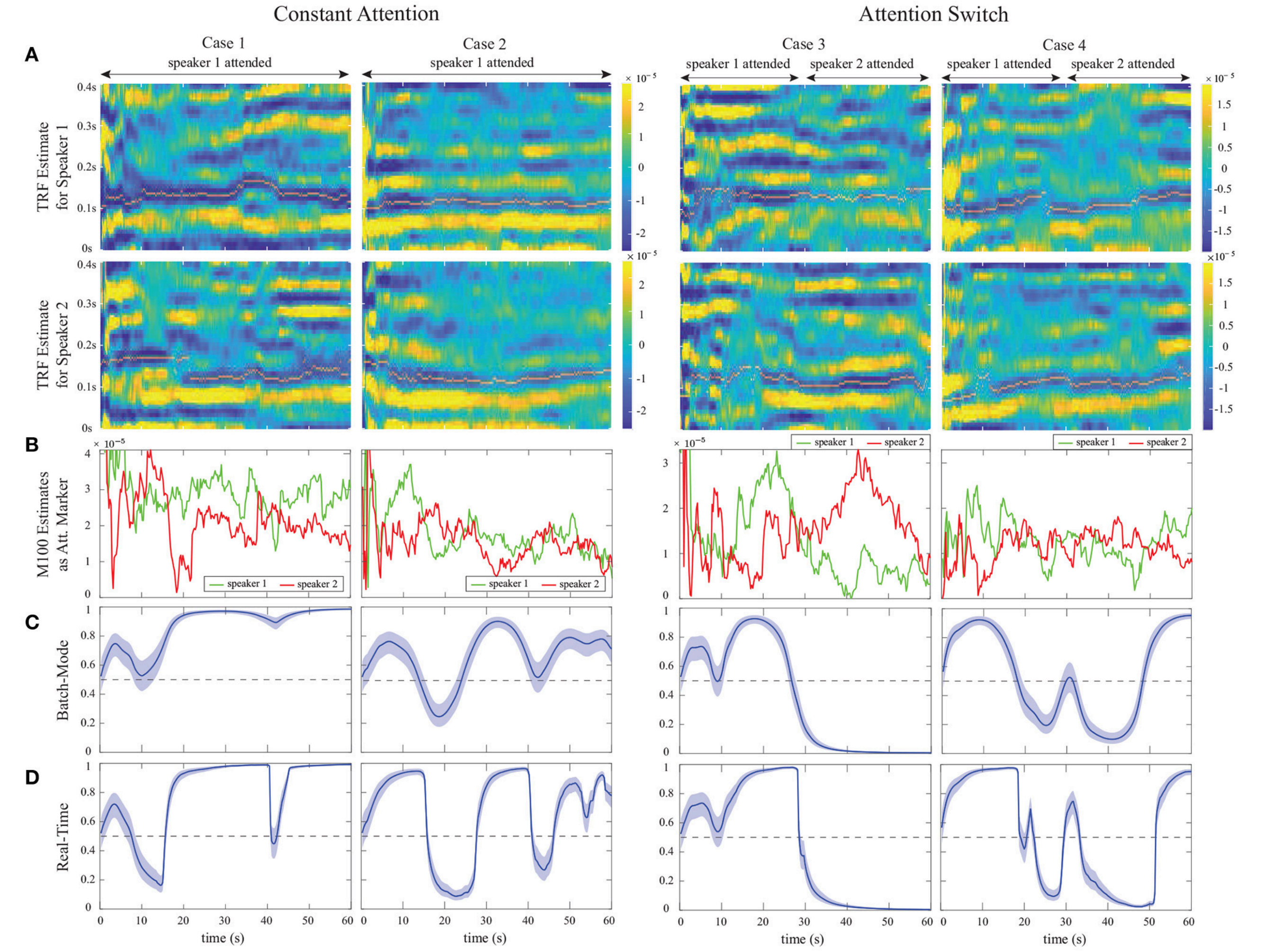


FIGURE 11 | Examples from the constant-attention and attention-switch MEG experiments, using the M100 attention marker, for trials with reliable (cases 1 and 3) and unreliable (cases 2 and 4) separation of the attended and unattended speakers. **(A)** TRF estimates for speakers 1 and 2 over time with the extracted M100 peak positions tracked by a narrow yellow line. **(B)** Extracted M100 peak magnitudes over time for speakers 1 and 2 as the attention marker. In cases 1 and 3, the M100 components exhibit a strong modulation effect of the attentional state, i.e., the attended speaker has a larger M100 peak, in contrast to cases 2 and 4, where there is a weak modulation. **(C)** Batch-mode state-space estimates of the attentional state. **(D)** Real-time state-space estimates of the attentional state. The strong or weak modulation effects of attentional state in the extracted M100 components directly affects the classification accuracy and the width of the confidence intervals for both the batch-mode and real-time estimators.

the extracted M100 values are not reliable representatives of the attentional state. Row A in **Figure 11** shows the estimated TRFs for speakers 1 and 2 in time for each of the four cases. The location of the M100 peaks is shown and tracked with a narrow line (yellow) on the extracted M100 components (blue). The M50 components are also evident as positive peaks occurring around the 50 ms lag. The M50 components do not strongly depend on the attentional state of the listener (Chait et al., 2004, 2010; Ding and Simon, 2012a; Akram et al., 2017), which is consistent with those shown in **Figure 11A**. It is worth noting that real-time estimation of the TRFs makes the estimates heavily affected by the dynamics of neural response and the background neural activity. Therefore, the estimates contain longer latency components which are typically suppressed in the offline estimates of TRFs common in the literature, which use multiple trial averaging to extract the stimulus-driven response (Ding and Simon, 2012a; Power et al., 2012). The width of the extracted components in **Figure 11** is due to the usage of a Gaussian dictionary matrix to represent the TRFs.

Row B in **Figure 11** displays the extracted M100 peak magnitudes over time for speakers 1 and 2. The attention modulation effect is more significant in cases 1 and 3. Rows C and D respectively show the batch-mode and real-time estimates of the attentional state based on the extracted M100 values. As expected, the batch-mode output is more robust to the fluctuations in the extracted M100 peak values, with smoother transitions and larger confidence intervals. Despite the poor attention modulation effect in cases 2 and 4, we observe that both the real-time and the batch-mode state-space models show reasonable performance in translating the extracted M100 peak values to a robust measure of the attentional state. This effect is notable in Rows C and D of Case 4. We performed the same analysis as in **Figure 9** to assess the effect of the forward-lag parameter K_F. Since the results were quite similar to those in **Figures 6**, **9**, we have omitted them for brevity and chose the same forward-lag of 1.5 s.

Finally, **Figure 12** summarizes the *real-time* classification results for the constant-attention (left panels) and attention-switch (right panels) MEG experiments. The classification convention is similar to that used in our EEG analysis, and is illustrated in **Figure 12A** for the completeness. For the attention-switch experiment, the 28–30 s interval is removed from the classification analysis, as it pertains to a silence period during which the subject is instructed to switch attention. **Figure 12B** shows the corresponding classification results, consisting of 36 trials for the constant-attention and 18 trials for the attention-switch experiments. Each circle corresponds to a single trial and the subjects in each experiment are color-coded. The average classification results per trial are shown in **Figure 12C** for each subject. The average hit rate and false alarm rates in the constant-attention experiments are respectively given by 71.67 and 20.81%. These quantities for the attention-switch experiment are respectively given by 64.12 and 26.16%, showing a reduction in hit rate and increase in false alarm.

4. DISCUSSION

In this work, we have proposed a framework for real-time decoding of the attentional state of a listener in a dual-speaker environment from M/EEG. This framework consists of three modules. In the first module, the encoding/decoding coefficients, relating the neural response to the envelopes of the two speech

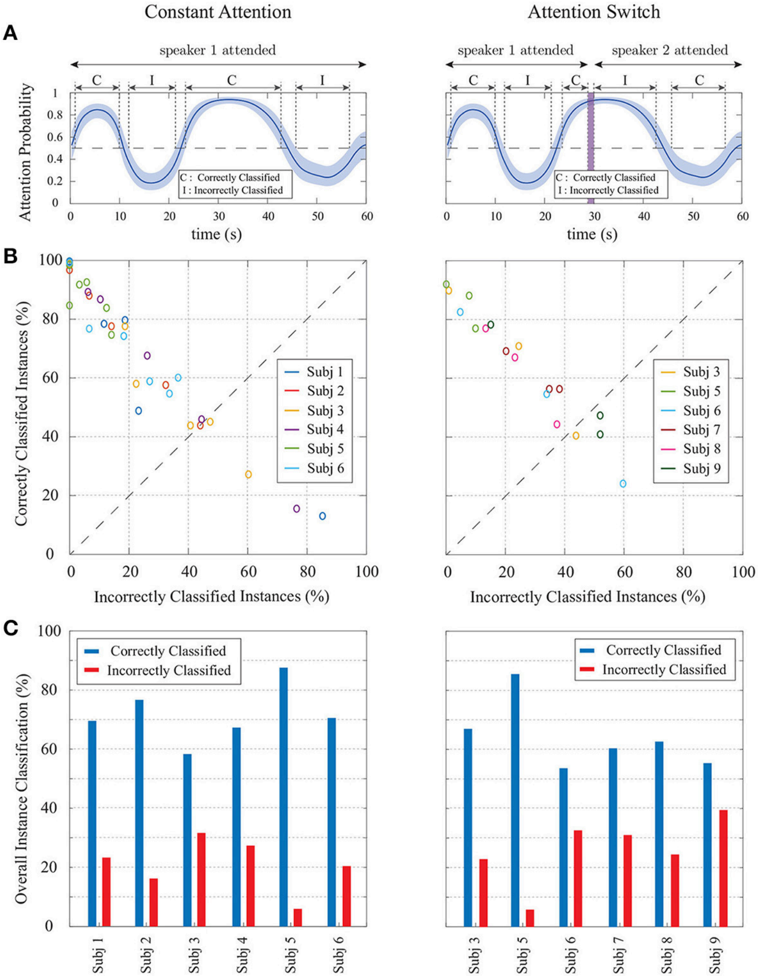


FIGURE 12 | Summary of real-time classification results for the constant-attention (left panels) and attention-switch (right panels) MEG experiments. **(A)** a generic instance of the state-space output for a trial illustrating the classification convention. **(B)** Classification results per trial for all subjects; each circle corresponds to a trial and the subjects are color-coded. The trials falling below the dashed line have more incorrectly classified instances than correctly classified ones. **(C)** Average classification performance over all trials for the six subjects.

streams, are estimated in a low-complexity and real-time fashion. Existing approaches for encoder/decoder estimation operate in an offline fashion using multiple experiment trials or large training datasets (O'Sullivan et al., 2015; Akram et al., 2016; Aroudi et al., 2016; Van Eyndhoven et al., 2017), and hence are not suitable for real-time applications with limited amount of training data and potential variability in the recording setup. To address this issue, we have integrated the forgetting factor mechanism used in adaptive filtering with ℓ_1-regularization, in order to capture the coefficient dynamics and mitigate overfitting.

In the second module, a function of the estimated encoding/decoding coefficients and the acoustic data, which we refer to as the *attention marker*, is calculated in real-time for each speaker. The role of the attention marker is to provide dynamic features that create statistical separation between the attended and the unattended speakers. Examples of such attention markers include correlation-based measures (e.g., correlation of the acoustic envelopes and their reconstruction from neural response), or measures solely based on the estimated decoding/encoding coefficients (e.g., the ℓ_1-norm of the decoder coefficients or the M100 peak of the encoder).

Finally, the attention marker is passed to the third module consisting of a near real-time state-space estimator. To control the delay in state estimation, we adopt a fixed-lag smoothing paradigm, in which the past and near future data are used to estimate the states. The role of the state-space model is to translate the noisy and highly variable attention markers to robust measures of the attentional state with minimal delay. We have archived a publicly available MATLAB implementation of our framework on the open source repository GitHub in order to ease reproducibility (Miran, 2017).

We validated the performance of our proposed framework using simulated EEG and MEG data, in which the ground truth attentional states are known. We also applied our proposed methods to experimentally recorded MEG and EEG data. As for a comparison benchmark to study the effect of the parameter choices in our real-time estimator, we considered the offline state-space attention decoding approach of Akram et al. (2016). Our MEG analysis showed that although the proposed real-time estimator has access to significantly fewer data points, it closely matches the outcome of the offline state-space estimator in Akram et al. (2016), for which the entire data from multiple trials are used for attention decoding. In particular, our analysis of the MEG data in constant-attention conditions revealed a hit rate of ~70% and a false alarm rate of ~20% at the group level. While the performance is slightly degraded compared to the offline analysis of Akram et al. (2016), our algorithms operate in real-time with 1.5 s built-in delay, over single trials, and using minimal tuning. Similarly, our analysis of EEG data provided ∼80% hit rate and ∼15% false alarm rate at a single trial level. These performance measures are slightly degraded compared to the results of offline approaches such as O'Sullivan et al. (2015).

Our proposed modular design admits the use of any attention-modulated statistic or feature as the attention marker, three of which have been considered in this work. While some attention markers perform better than the rest in certain applications, our goal in this work was to provide different examples of attention markers which can be used in the encoding/decoding models based on the literature, rather than comparing their performance against each other. The choice of the best attention marker that results in the highest classification accuracy is a problem-specific matter. Our modular design allows to evaluate the performance of a variety of attention markers for a given experimental setting, while fixing the encoding/decoding estimation and state-space modules, and to choose one that provides the desired classification performance. Our state-space module can also operate on the output of existing methods with encoder/decoder coefficients that are pre-estimated using training datasets (O'Sullivan et al., 2015; Zink et al., 2017) to provide a robust and statistically interpretable measure of the attentional state at high temporal resolutions.

A practical limitation of our proposed methodology in its current form is the need to have access to clean acoustic data in order to form regressors based on the speech envelopes. In a realistic scenario, the speaker envelopes have to be extracted from the noisy mixture of speeches recorded by microphone arrays. Thanks to a number of fairly recent results in attention decoding literature (Biesmans et al., 2015, 2017; Aroudi et al., 2016; O'Sullivan et al., 2017; Van Eyndhoven et al., 2017), it is possible to integrate our methodology with a pre-processing module that extracts the acoustic features of individual speech streams from their noisy mixtures. We view this extension as a future direction of research.

The proposed approach requires a minimal amount of *labeled* training data for tuning purposes. However, we can determine the attended speaker in an unlabeled dataset as the speaker whose speech signal best fits the EEG data or whose encoder/decoder estimates have larger peaks at certain time lags, and then train the decoders or hyperparameters with these data-driven labels. This can be done both in existing methods such as that of O'Sullivan et al. (2015) for attended decoder estimation and in our approach for capturing the statistical properties of attention markers for hyperparameter tuning. We view this extension to deal with unlabeled data as a future direction of research.

Our proposed framework has several advantages over existing methodologies. First, our algorithms require minimal amount of offline tuning or training. The subject-specific hyperparameters used by the algorithms are tuned prior to real-time application in a supervised manner. The only major offline tuning step in our framework is computing the subject-specific channel weights in the encoding model for MEG analysis in order to extract the auditory component of the neural response. This is due to the fact that the channel locations are not fixed with respect to the head position across subjects. It is worth noting that this step can be avoided if the encoding model treats the MEG channels separately in a multivariate model. Given that recent studies suggest that the M100 component of the encoder obtained from the MEG auditory response is a reliable attention marker (Ding and Simon, 2012a,b; Akram et al., 2017), we adopted the DSS algorithm for computing the channel weights that compose the auditory response in an offline fashion.

Second, our framework yields robust attention decoding performance at a temporal resolution in the order of ∼1 second, comparable to that at which humans switch their attention from one speaker to another. The accuracy of existing methods, however, significantly degrades when they operate at these temporal resolutions (Zink et al., 2016, 2017). Our proposed framework operates in a near real-time fashion, where the attention decoding delay can be adjusted for controlling the trade-off between robustness and adaptivity of the attentional state estimates. In addition, the probabilistic output of our attentional state decoding framework can be used for further statistical analysis and soft-decision mechanisms which are desired in smart hearing aid applications. Finally, the modular design of our framework facilitates its adaptation to more complex auditory scenes (e.g., with multiple speakers and realistic noise and reverberation conditions) and integration of other covariates relevant to real-time applications (e.g., electrooculography measurements).

AUTHOR CONTRIBUTIONS

TZ, JZS, and BB designed the research. SM and BB performed the research, with contributions to the methods by AS, SA, and TZ and experimental data supplied by TZ and JZS. All authors participated in writing the paper.

FUNDING

This material is based on work supported in part by the National Science Foundation Awards No. 1552946 and 1734892, and a research gift from the Starkey Hearing Technologies to the University of Maryland.

ACKNOWLEDGMENTS

The authors would like to thank Dr. Tom Goldstein for helpful remarks on adapting the FASTA package options to our decoder/encoder estimation problem.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fnins.2018.00262/full#supplementary-material

REFERENCES

Akram, S., Presacco, A., Simon, J. Z., Shamma, S. A., and Babadi, B. (2016). Robust decoding of selective auditory attention from MEG in a competing-speaker environment via state-space modeling. *NeuroImage* 124, 906–917. doi: 10.1016/j.neuroimage.2015.09.048

Akram, S., Simon, J. Z., and Babadi, B. (2017). Dynamic estimation of the auditory temporal response function from MEG in competing-speaker environments. *IEEE Trans. Biomed. Eng.* 64, 1896–1905. doi: 10.1109/TBME.2016.2628884

Akram, S., Simon, J. Z., Shamma, S. A., and Babadi, B. (2014). "A state-space model for decoding auditory attentional modulation from MEG in a competing-speaker environment," in *Advances in Neural Information Processing Systems* (Montreal, QC), 460–468.

Aroudi, A., Mirkovic, B., De Vos, M., and Doclo, S. (2016). "Auditory attention decoding with EEG recordings using noisy acoustic reference signals," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on* (Shanghai: IEEE), 694–698.

Biesmans, W., Das, N., Francart, T., and Bertrand, A. (2017). Auditory-inspired speech envelope extraction methods for improved EEG-based auditory attention detection in a cocktail party scenario. *IEEE Trans. Neural Syst. Rehabil. Eng.* 25, 402–412. doi: 10.1109/TNSRE.2016.2571900

Biesmans, W., Vanthornhout, J., Wouters, J., Moonen, M., Francart, T., and Bertrand, A. (2015). "Comparison of speech envelope extraction methods for EEG-based auditory attention detection in a cocktail party scenario," in *Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE* (Milan: IEEE), 5155–5158.

Bleichner, M. G., Mirkovic, B., and Debener, S. (2016). Identifying auditory attention with ear-EEG: cEEGrid versus high-density cap-EEG comparison. *J. Neural Eng.* 13:066004. doi: 10.1088/1741-2560/13/6/066004

Bregman, A. S. (1994). *Auditory Scene Analysis: The Perceptual Organization of Sound*. Cambridge, MA: MIT Press.

Brungart, D. S. (2001). Informational and energetic masking effects in the perception of two simultaneous talkers. *J. Acoust. Soc. Am.* 109, 1101–1109. doi: 10.1121/1.1345696

Chait, M., de Cheveigné, A., Poeppel, D., and Simon, J. Z. (2010). Neural dynamics of attending and ignoring in human auditory cortex. *Neuropsychologia* 48, 3262–3271. doi: 10.1016/j.neuropsychologia.2010.07.007

Chait, M., Simon, J. Z., and Poeppel, D. (2004). Auditory m50 and m100 responses to broadband noise: functional implications. *Neuroreport* 15, 2455–2458. doi: 10.1097/00001756-200411150-00004

Cherry, E. C. (1953). Some experiments on the recognition of speech, with one and with two ears. *J. Acoust. Soc. Am.* 25, 975–979. doi: 10.1121/1.1907229

Combettes, P. L., and Pesquet, J.-C. (eds.). (2011). "Proximal splitting methods in signal processing," in *Fixed-Point Algorithms for Inverse Problems in Science and Engineering* (New York, NY: Springer), 185–212.

de Cheveigné, A., and Simon, J. Z. (2008). Denoising based on spatial filtering. *J. Neurosci. Methods* 171, 331–339. doi: 10.1016/j.jneumeth.2008.03.015

Ding, N., and Simon, J. Z. (2012a). Emergence of neural encoding of auditory objects while listening to competing speakers. *Proc. Natl. Acad. Sci. U.S.A.* 109, 11854–11859. doi: 10.1073/pnas.1205381109

Ding, N., and Simon, J. Z. (2012b). Neural coding of continuous speech in auditory cortex during monaural and dichotic listening. *J. Neurophysiol.* 107, 78–89. doi: 10.1152/jn.00297.2011

Fishman, Y. I., and Steinschneider, M. (2010). Neural correlates of auditory scene analysis based on inharmonicity in monkey primary auditory cortex. *J. Neurosci.* 30, 12480–12494. doi: 10.1523/JNEUROSCI.1780-10.2010

Goldstein, T., Studer, C., and Baraniuk, R. (2014). A field guide to forward-backward splitting with a FASTA implementation. *arXiv*:abs/1411.3406.

Goldstein, T., Studer, C., and Baraniuk, R. (2015). *FASTA: A Generalized Implementation of Forward-Backward Splitting*. Available online at: http://arxiv.org/abs/1501.04979

Griffiths, T. D., and Warren, J. D. (2004). What is an auditory object? *Nat. Rev. Neurosci.* 5:887. doi: 10.1038/nrn1538

Haykin, S., and Chen, Z. (2005). The cocktail party problem. *Neural Comput.* 17, 1875–1902. doi: 10.1162/0899766054322964

Kähkönen, S., Ahveninen, J., Jääskeläinen, I. P., Kaakkola, S., Näätänen, R., Huttunen, J., et al. (2001). Effects of haloperidol on selective attention: a combined whole-head MEG and high-resolution EEG study. *Neuropsychopharmacology* 25, 498–504. doi: 10.1016/S0893-133X(01)00255-X

Kaya, E. M., and Elhilali, M. (2017). Modelling auditory attention. *Philos. Trans. R. Soc. B* 372:20160101. doi: 10.1098/rstb.2016.0101

Khalighinejad, B., da Silva, G. C., and Mesgarani, N. (2017). Dynamic encoding of acoustic features in neural responses to continuous speech. *J. Neurosci.* 37, 2176–2185. doi: 10.1523/JNEUROSCI.2383-16.2017

McDermott, J. H. (2009). The cocktail party problem. *Curr. Biol.* 19, R1024–R1027. doi: 10.1016/j.cub.2009.09.005

Middlebrooks, J. C., Simon, J. Z., Popper, A. N., and Fay, R. R. (eds.). (2017). "The auditory system at the cocktail party," in *The Springer Handbook of Auditory Research Series* (Springer), 60.

Miran, S. (2017). *Real-Time Tracking of Selective Auditory Attention MATLAB Code*. Available omline at: https://github.com/sinamiran/Real-Time-Tracking-of-Selective-Auditory-Attention

Mirkovic, B., Debener, S., Jaeger, M., and De Vos, M. (2015). Decoding the attended speech stream with multi-channel EEG: implications for online, daily-life applications. *J. Neural Engi.* 12:046007. doi: 10.1088/1741-2560/12/4/046007

O'Sullivan, J., Chen, Z., Herrero, J., McKhann, G. M., Sheth, S. A., Mehta, A. D., et al. (2017). Neural decoding of attentional selection in multi-speaker environments without access to clean sources. *J. Neural Eng.* 14:056001. doi: 10.1088/1741-2552/aa7ab4

O'Sullivan, J. A., Power, A. J., Mesgarani, N., Rajaram, S., Foxe, J. J., Shinn-Cunningham, B. G., et al. (2015). Attentional selection in a cocktail party environment can be decoded from single-trial EEG. *Cereb. Cortex* 25, 1697–1706. doi: 10.1093/cercor/bht355

Power, A. J., Foxe, J. J., Forde, E.-J., Reilly, R. B., and Lalor, E. C. (2012). At what time is the cocktail party? A late locus of selective attention to natural speech. *Eur. J. Neurosci.* 35, 1497–1503. doi: 10.1111/j.1460-9568.2012.08060.x

Särelä, J., and Valpola, H. (2005). Denoising source separation. *J. Mach. Learn. Res.* 6, 233–272. Available online at: http://www.jmlr.org/papers/volume6/sarela05a/sarela05a.pdf

Shamma, S. A., Elhilali, M., and Micheyl, C. (2011). Temporal coherence and attention in auditory scene analysis. *Trends Neurosci.* 34, 114–123. doi: 10.1016/j.tins.2010.11.002

Sheikhattar, A., Fritz, J. B., Shamma, S. A., and Babadi, B. (2015a). "Adaptive sparse logistic regression with application to neuronal plasticity analysis," in *Signals, Systems and Computers, 2015 49th Asilomar Conference on* (Pacific Grove, CA: IEEE), 1551–1555.

Sheikhattar, A., Fritz, J. B., Shamma, S. A., and Babadi, B. (2015b). Recursive sparse point process regression with application to spectrotemporal receptive field plasticity analysis. *IEEE Trans. Signal Process.* 64, 2026–2039. doi: 10.1109/TSP.2015.2512560

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *J. R. Stat. Soc. Ser. B (Methodol.)* 58, 267–288.

Van Eyndhoven, S., Francart, T., and Bertrand, A. (2017). EEG-informed attended speaker extraction from recorded speech mixtures with application in neuro-steered hearing prostheses. *IEEE Trans. Biomed. Eng.* 64, 1045–1056. doi: 10.1109/TBME.2016.2587382

Zink, R., Baptist, A., Bertrand, A., Van Huffel, S., and De Vos, M. (2016). "Online detection of auditory attention in a neurofeedback application," in *Proceedings of the 8th International Workshop on Biosignal Interpretation* (Osaka), 1–4.

Zink, R., Proesmans, S., Bertrand, A., Van Huffel, S., and De Vos, M. (2017). Online detection of auditory attention with mobile EEG: closing the loop with neurofeedback. *bioRxiv* 218727. doi: 10.1101/218727

Zion Golumbic E. M., Ding, N., Bickel, S., Lakatos, P., Schevon, C. A., McKhann, G. M., et al. (2013). Mechanisms underlying selective neuronal tracking of attended speech at a "cocktail party." *Neuron* 77, 980–991. doi: 10.1016/j.neuron.2012.12.037

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest. Author TZ is employed by Starkey Hearing Technologies. Author SA is employed by Facebook. All other authors declare no competing interests.

Advantages of publishing in Frontiers

OPEN ACCESS
Articles are free to read for greatest visibility and readership

FAST PUBLICATION
Around 90 days from submission to decision

HIGH QUALITY PEER-REVIEW
Rigorous, collaborative, and constructive peer-review

TRANSPARENT PEER-REVIEW
Editors and reviewers acknowledged by name on published articles

Frontiers
Avenue du Tribunal-Fédéral 34
1005 Lausanne | Switzerland

Visit us: www.frontiersin.org
Contact us: info@frontiersin.org | +41 21 510 17 00

REPRODUCIBILITY OF RESEARCH
Support open data and methods to enhance research reproducibility

DIGITAL PUBLISHING
Articles designed for optimal readership across devices

FOLLOW US
@frontiersin

IMPACT METRICS
Advanced article metrics track visibility across digital media

EXTENSIVE PROMOTION
Marketing and promotion of impactful research

LOOP RESEARCH NETWORK
Our network increases your article's readership