



OPEN ACCESS

EDITED BY

Jianjun Meng,
Shanghai Jiao Tong University, China

REVIEWED BY

Guangye Li,
Shanghai Jiao Tong University, China
Romulo Fuentes Flores,
University of Chile, Chile

*CORRESPONDENCE

Tetiana Aksenova
✉ tetiana.aksenova@cea.fr

SPECIALTY SECTION

This article was submitted to
Brain-Computer Interfaces,
a section of the journal
Frontiers in Human Neuroscience

RECEIVED 20 October 2022

ACCEPTED 03 February 2023

PUBLISHED 06 March 2023

CITATION

Moly A, Aksenov A, Martel F and Aksenova T
(2023) Online adaptive group-wise sparse
Penalized Recursive Exponentially Weighted
N-way Partial Least Square for epidural
intracranial BCI.
Front. Hum. Neurosci. 17:1075666.
doi: 10.3389/fnhum.2023.1075666

COPYRIGHT

© 2023 Moly, Aksenov, Martel and Aksenova.
This is an open-access article distributed under
the terms of the [Creative Commons Attribution
License \(CC BY\)](https://creativecommons.org/licenses/by/4.0/). The use, distribution or
reproduction in other forums is permitted,
provided the original author(s) and the
copyright owner(s) are credited and that the
original publication in this journal is cited, in
accordance with accepted academic practice.
No use, distribution or reproduction is
permitted which does not comply with these
terms.

Online adaptive group-wise sparse Penalized Recursive Exponentially Weighted N-way Partial Least Square for epidural intracranial BCI

Alexandre Moly¹, Alexandre Aksenov², Félix Martel¹ and Tetiana Aksenova^{1*}

¹Université Grenoble Alpes, CEA, LETI, Clinatec, Grenoble, France, ²Independent Researcher, Montpellier, France

Introduction: Motor Brain-Computer Interfaces (BCIs) create new communication pathways between the brain and external effectors for patients with severe motor impairments. Control of complex effectors such as robotic arms or exoskeletons is generally based on the real-time decoding of high-resolution neural signals. However, high-dimensional and noisy brain signals pose challenges, such as limitations in the generalization ability of the decoding model and increased computational demands.

Methods: The use of sparse decoders may offer a way to address these challenges. A sparsity-promoting penalization is a common approach to obtaining a sparse solution. BCI features are naturally structured and grouped according to spatial (electrodes), frequency, and temporal dimensions. Applying group-wise sparsity, where the coefficients of a group are set to zero simultaneously, has the potential to decrease computational time and memory usage, as well as simplify data transfer. Additionally, online closed-loop decoder adaptation (CLDA) is known to be an efficient procedure for BCI decoder training, taking into account neuronal feedback. In this study, we propose a new algorithm for online closed-loop training of group-wise sparse multilinear decoders using L_p -Penalized Recursive Exponentially Weighted N-way Partial Least Square (PREW-NPLS). Three types of sparsity-promoting penalization were explored using L_p with $p = 0.$, 0.5 , and 1 .

Results: The algorithms were tested offline in a pseudo-online manner for features grouped by spatial dimension. A comparison study was conducted using an epidural ECoG dataset recorded from a tetraplegic individual during long-term BCI experiments for controlling a virtual avatar (left/right-hand 3D translation). Novel algorithms showed comparable or better decoding performance than conventional REW-NPLS, which was achieved with sparse models. The proposed algorithms are compatible with real-time CLDA.

Discussion: The proposed algorithm demonstrated good performance while drastically reducing the computational load and the memory consumption. However, the current study is limited to offline computation on data recorded with a single patient, with penalization restricted to the spatial domain only.

KEYWORDS

Brain-Computer Interface, ECoG, penalization, tensor factorization, NPLS, group-wise sparsity, adaptive learning, incremental learning

1. Introduction

Brain–computer interfaces (BCIs) are systems that create a new communication pathway between the brain and an effector without neuromuscular activation. Motor BCIs aim to allow users with severe motor impairments to regain limb mobility by controlling orthoses and prostheses or by recovering motor control over their own limbs, e.g., using electrical stimulation (Mak and Wolpaw, 2009). Most BCI systems include a neural signal acquisition system, a transducer, an effector, and a feedback system (Schwartz et al., 2006). The transducer is typically composed of a neural feature extraction block and a decoder, with optional pre- and post-processing blocks. Decoders are generally user-specific and data-driven. They are created through a supervised tuning of parameters on the training dataset. Once the decoder is established, the transducer can be applied in real time to translate the user's intention into the control command of the effector.

As BCI systems function in real time, computing time and resource management are crucial aspects of such systems (Haufe et al., 2014). High-resolution neuronal activity recording systems are generally required to achieve high-dimensional control of complex effectors. It results in a large volume of data that needs to be processed.

Furthermore, in motor BCI, a high decision rate (8–10 Hz) is necessary to control complex effectors such as robotic arms, an exoskeleton, and so on (Marathe and Taylor, 2015; Shanechi et al., 2017). In addition to high computing power requirements and computing time, the high dimensionality of the feature space presents challenges such as the “curse of dimensionality” during decoder training (Bellman, 1961; Bishop, 2006; Nicolas-Alonso and Gomez-Gil, 2012; Remeseiro and Bolon-Canedo, 2019). The feature space often contains irrelevant and/or redundant features. Moreover, computational load is critical for the potential development of portable BCIs.

Reducing the dimensionality of the neuronal feature space is one approach to addressing these issues. Dimension reduction algorithms have been widely employed in numerous studies on BCI.

Both projection and feature selection methods were applied for dimensional reduction, in both online BCI experiments and offline analysis.

Projections algorithms were often used (Kim et al., 2006; Marathe and Taylor, 2013; Haufe et al., 2014; Bundy et al., 2016; Hsu et al., 2016; Sannelli et al., 2016; Schaeffer and Aksenova, 2016; Eliseyev et al., 2017; Jiang et al., 2017; Seifzadeh et al., 2017; Sreenath and Ramana, 2017; Bousseta et al., 2018; Choi et al., 2018; Lotte et al., 2018; Palmer and Hirata, 2018; Khan et al., 2019; Jafarifarmand and Badamchizadeh, 2020). They project the feature space into a space of a lower dimension by using a linear or non-linear combination of the initial features. The principal and independent component analyses (PCA and ICA), spatio-spectral decomposition (SSD), common spatial pattern (CSP), or partial least squares (PLS) (Kim et al., 2006; Marathe and Taylor, 2013; Haufe et al., 2014; Bundy et al., 2016; Hsu et al., 2016; Sannelli et al., 2016; Schaeffer and Aksenova, 2016; Eliseyev et al., 2017; Jiang et al., 2017; Seifzadeh et al., 2017; Sreenath and Ramana, 2017; Bousseta et al., 2018; Choi et al., 2018; Lotte et al., 2018; Palmer and Hirata,

2018; Khan et al., 2019; Jafarifarmand and Badamchizadeh, 2020) algorithms and variants were applied in BCI research. However, such methods may not improve the computing time as they do not optimize the feature extraction step. The irrelevant and/or redundant features are still computed.

The feature selection family regroups filter-based, wrapper-based, and embedded techniques (Bolón-Canedo et al., 2013; Khaire and Dhanalakshmi, 2019). The filter-based methods rank and select features independently without considering the decoder. Effective in computation time, these methods tend to select highly correlated (redundant) features. The wrapper-based techniques incorporate supervised learning algorithms to evaluate the possible interactions between the features. These techniques add features to the selected subset iteratively and evaluate the subset by combining it with the trained decoder (Lotte et al., 2018). These methods are effective but require a great deal of computing time. On the other hand, embedded techniques integrate the feature selection process directly into the decoding algorithm, thus combining the advantages of both the filter-based and wrapper-based methods (Khaire and Dhanalakshmi, 2019). Embedded feature selection is a promising approach as it is directly performed during the model learning process.

Embedded feature selection in BCI is based on regularization/penalization methods (Cincotti et al., 2008; Lotte and Guan, 2011; Flamary and Rakotomamonjy, 2012; Eliseyev and Aksenova, 2016; Mishra et al., 2018; Nagel and Spüler, 2019). Regularization strategies incorporate a penalty term into the model parameter optimization process to restrict the degree of freedom of the model. Numerous regularization strategies were designed, such as L_0 , L_1 (Lasso), L_2 (Ridge), elastic net regularization, and so on. The L_1 regularization process adds a penalty term equal to the sum of the absolute values of model coefficients. L_2 regularization integrates a penalty term equal to the sum of squared values, whereas elastic net regularization is a combination of both L_1 and L_2 penalizations (Bishop, 2006). L_0 (Sreeja and Himanshu, 2019), L_1 (Lotte and Guan, 2011; Eliseyev et al., 2012; Flamary and Rakotomamonjy, 2012; Zhang et al., 2013; López-Larraz et al., 2014), L_2 (Cincotti et al., 2008; Flamary and Rakotomamonjy, 2012; Seifzadeh et al., 2017; Nagel and Spüler, 2019) regularization, and an elastic net (Kim et al., 2018; Peterson et al., 2019) were applied in the BCI field to improve the neural signal decoding performance or other properties, e.g., prediction smoothness or sparsity. Less common regularization methods designed and/or applied in BCI are regularization using polynomial regression (Eliseyev and Aksenova, 2016), sparse regularization based on automatic relevance determination (ARD) (Toda et al., 2011; Nakanishi et al., 2017), and Kullback–Leibler regularization in the Riemannian mean (Mishra et al., 2018). Among regularization techniques, L_p regularization, $0 \leq p \leq 1$, using a penalty equal to the L_p ($0 \leq p \leq 1$) norm/pseudo-norm of the model coefficients, is known to be effective in discarding irrelevant/correlated features, promoting a sparse solution (Bishop, 2006; Hastie et al., 2015).

Regularization/penalization is generally performed in a single-wise manner. Features are regularized independently and are not evaluated as belonging to a group of features. However, there are many applications, particularly in BCI, with structurally grouped input features. This approach allows the simultaneous setting

of zero of the model coefficients within a group, which can be beneficial in cases where the input features are structurally grouped, such as excluding sensors (Eliseyev et al., 2017). For such applications, a single-wise sparsity-promoting penalty may be suboptimal. Group-wise regularization algorithms perform the feature selection process by grouping and applying penalization to the groups at once (Martínez-Montes et al., 2008; Eliseyev et al., 2012; Giordani and Rocci, 2013; Zhang et al., 2013; Hastie et al., 2015). This grouping can cluster features across various modalities, such as electrodes and frequency bands (van Gerven et al., 2009). Group-wise sparsity is ideal for naturally structured data, allowing for the elimination of variables (such as electrodes or frequency) from the signal processing workflow and reducing computational costs. Additionally, it may simplify the model's interpretation.

Despite being potentially beneficial for studies on BCI, group-wise penalization has rarely been applied in this field. The natural structure of BCI features space grouping features over modalities, such as electrodes, frequency bands, and time delay, which are heavily exploited (van Gerven et al., 2009; Eliseyev et al., 2012; Motrenko and Strijov, 2018; Wu et al., 2019). For the penalization of such groups, data may be viewed in the form of a tensor (Hastie et al., 2015; Eliseyev and Aksenova, 2016). Tensors, or multiway arrays, are higher-order generalizations of vectors and matrices (for more details, see Ref. Hsu et al., 2016). Tensors have several dimensions, also known as ways of analysis or modes. In BCIs, recorded signals are primarily analyzed in spatial, frequency, and temporal domains. To extract neural features, each epoch of neural signal recording is commonly mapped to temporal-frequency-spatial (or frequency-spatial) space (e.g., Chao et al., 2010; Schaeffer and Aksenova, 2016; Eliseyev et al., 2017; Choi et al., 2018). Tensor-based analyses, which present features in matrix form using tensor unfolding, are reported to be beneficial for BCI decoding as they preserve the natural structure of data (Zhao et al., 2013; Cichocki et al., 2015; Zhang et al., 2015; Eliseyev et al., 2017).

The tensors of data in BCI often benefit from tensor decomposition techniques. Slice-wise data representation can be obtained through sparsity-promoting penalization of tensor decomposition. Regularized PARAFAC and Tucker decomposition are algorithms designed for group-wise tensor penalization. Slice-wise sparsity-promoting penalization is added to the N-way Partial Least Square (NPLS) in van Gerven et al. (2009), Eliseyev et al. (2012), Motrenko and Strijov (2018), and Wu et al. (2019). These techniques have been used in a few offline BCI studies (Martínez-Montes et al., 2008; Eliseyev et al., 2012) and in other fields (Giordani and Rocci, 2013; Kim et al., 2013, 2014; Hervás et al., 2019). In Eliseyev et al. (2012), the L_1 -Regularized N-PLS algorithm was shown to be superior to its non-penalized version by suppressing noisy/irrelevant electrodes. However, these algorithms were not adapted for online decoder training in closed-loop use.

Most of the presented feature-dimensional reduction algorithms were tested offline. Additionally, feature selection performed in an offline preliminary study (Brunner et al., 2006; Huang et al., 2009; Spüler et al., 2012; Marathe and Taylor, 2013; Bousseta et al., 2018; Cantillo-Negrete et al., 2018; Kim et al., 2018; Nagel and Spüler, 2019) may not be optimal when using a CLDA strategy (Schlögl et al., 2010; Clerc et al., 2016). CLDA involves training the decoder online on data acquired during

closed-loop BCI control sessions. Decoders trained in this manner have been reported to outperform decoders trained offline using data from open-loop BCI experiments (Jarosiewicz et al., 2013). Adaptive/incremental learning algorithms are particularly suited for the CLDA strategy. These algorithms continuously update the model, using only the latest data block and relevant statistics on the older signals, while not retaining the whole signals in memory (Schlögl et al., 2010; Brandman et al., 2018; Lotte et al., 2018). Adaptive/incremental learning algorithms are beneficial for the CLDA training strategy as they allow higher decoder update rates and are compatible with long decoder learning periods, which are generally necessary for high-dimensional control. However, only a few adaptive dimensional reduction algorithms were proposed. The adaptive dimensional reduction algorithms applied in the BCI (Zhao et al., 2008; Ang et al., 2011; Song and Yoon, 2015; Woehrle et al., 2015; Hsu et al., 2016; Mobaien and Boostani, 2016; Sannelli et al., 2016; Chen and Fang, 2017; Lotte et al., 2018) and other (Dagher, 2010) fields are primarily based on projection strategies (adaptive CSP, PCA, ICA, and xDAWN algorithms) and were only tested offline. Similarly, a few adaptive feature selection algorithms were proposed. Filter methods were tested on BCI simulations using mutual information (Oliver et al., 2013) or during online BCI experiments based on the Fisher score (Faller et al., 2012). The wrapper-based strategy was optimized using parallel computation for the online BCI classifier (Mend and Kullmann, 2012), whereas embedded methods using semi-supervised feature selection (Long et al., 2011) and a weighting feature algorithm (Andreu-Perez et al., 2018) have been designed and used during online BCI applications. An adaptive genetic algorithm was proposed for adaptive channel selection in Moro et al. (2017). However, these methods have been applied to simple online binary classification only.

In BCI studies, adaptive regularization algorithms were tested offline (Roijsdijk et al., 2016; Mishra et al., 2018; Sharghian et al., 2019). Adaptive algorithms with an L_1 norm regularization strategy were reported in an adaptive logistic regression (Sheikhhattar et al., 2015), a kernel least squares (Yang et al., 2019), and recursive least squares algorithms (Chen et al., 2012) in other fields. Only a few feature selection methods were integrated into adaptive algorithms for online incremental calibration during real-time BCI experiments and were generally restricted to binary classification and EEG-based experiments (Long et al., 2011; Faller et al., 2012; Mend and Kullmann, 2012; Moro et al., 2017; Andreu-Perez et al., 2018). Computational complexity and difficulty integrating dimensional reduction methods into real-time algorithms may explain the lack of proposed solutions.

In the study, an adaptive algorithm promoting group-wise model sparsity, L_p -penalized recursive exponentially weighted N-way Partial Least Square (PREW-NPLS), is proposed. L_p , $p = 0, 0.5, 1$ norm/pseudo-norm penalty is applied to feature groups corresponding to the slices of data tensor related to the mode of analysis (e.g., spatial, frequency, temporal). The algorithm was tested with data recorded during BCI sessions of left/right arm 3D translations of a virtual avatar by a tetraplegic patient during the clinical trial NCT02550522 (ClinicalTrials.gov) conducted at the Grenoble Alpes University Hospital (CHUGA) (Benabid et al., 2019; Moly et al., 2022). The datasets were recorded during online

closed-loop experiments using the REW-NPLS decoder that was previously integrated into the BCI system. To reproduce online experimental conditions and to ensure that the proposed algorithm is compatible with real-time CLDA, a pseudo-online simulation was conducted using the same parameters (buffer size, batch training, and so on) and the same model application procedure as it was used in real time. The comparison study was restricted to penalization according to spatial modality, which is the most critical at BCIs due to data recording/transfer limitations. For each type of penalization, a set of models/penalization parameters were evaluated. The PREW-NPLS decoders highlighted equivalent or better decoding performance compared to the generic REW-NPLS algorithm for the majority of the penalization parameters. The sparsest solutions allowed the removal of up to 75% of the electrodes without decreasing performance. L_0 -PREW-NPLS and L_1 -PREW-NPLS are sufficiently computationally efficient for online closed-loop decoder adaptation at a high-frequency rate.

2. Methods

2.1. Generic partial least squares and family

Algorithms of the PLS family are widely used in BCI studies due to their stability in the case of high-dimensional data and in the presence of correlated and/or irrelevant variables. In motor BCI, the algorithms of the PLS family were applied in both continuous and discrete BCIs. Offline hand/fingers trajectory decoding (Chao et al., 2010; Chen et al., 2013; Eliseyev and Aksenova, 2014; Bundy et al., 2016; Schaeffer and Aksenova, 2016; Schaeffer, 2017; Choi et al., 2018), real-time hand translation/wrist rotation control (Benabid et al., 2019; Moly et al., 2022), error potential (ERP) detection (Rouanne et al., 2021) from ECoG, and EEG/MEG-based classification (Trejo et al., 2006; Eliseyev et al., 2017; Maleki et al., 2018) using the PLS algorithm were reported in preclinical and clinical studies. In Eliseyev and Aksenova (2014), GAM-PLS (generalized additive model—the partial least square) is reported to outperform the generic PLS in the presence of artifacts for 3D hand trajectory decoding from ECoG data in non-human primates (NHP). In Schaeffer (2017), PLS outperformed principle component regression and demonstrated comparable results with Lasso regression for hand trajectory and 1D finger trajectory decoding from ECoG in preclinical and clinical experiments. In Rouanne et al. (2021), NPLS demonstrated comparable results with Logistic Regression, SVM, MLP, and CNN for ERP detection from ECoG recordings in the sensory-motor cortex of tetraplegics, etc.

The generic PLS is a linear regression algorithm based on the iterative projection of input and output variables into the latent variable spaces of dimension f . The hyperparameter f is generally estimated through cross-validation in the preliminary study. Projectors are set to maximize the covariance between the input and latent output variables. The generic PLS is an offline algorithm. For online data stream modeling, recursive PLS (RPLS) and recursive exponentially weighted PLS (REW PLS) (Helland et al., 1992; Dayal and MacGregor, 1997; Qin, 1998) were developed. All the aforementioned PLS algorithms are vector-input-vector-output algorithms. N-way Partial Least Square (NPLS) is a generalization of the conventional PLS for tensor data (Bro, 1996, 1998). The

NPLS algorithm projects the input and output tensors into a low-dimensional space of latent variables using a low-rank tensor decomposition. The recursive N-way PLS (RNPLS) (Helland et al., 1992) and recursive exponentially weighted N-way PLS (REW-NPLS) (Dayal and MacGregor, 1997) are generalizations of the adaptive RPLS and REW PLS algorithms to tensor variables and allow online tensor data stream learning of the regression model. RNPLS still requires fixing the hyperparameter f from the offline preliminary study, whereas REW-NPLS proposes a recursive validation procedure for the online optimization of the hyperparameter, enabling a fully adaptive algorithm (Dayal and MacGregor, 1997). In addition, the REW-NPLS algorithm is more computationally effective than the RPLS algorithm (Dayal and MacGregor, 1997).

The adaptive REW-NPLS algorithm has been tested offline in BCI studies for trajectory decoding from ECoG signals and for classification from MEG data, demonstrating similar or better results compared to other algorithms from the PLS family designed for offline use (Eliseyev et al., 2017). It was applied in real time for closed-loop adaptation of 3D hand translation/wrist rotation decoders in tetraplegics (Benabid et al., 2019; Moly et al., 2022). Finally, REW-NPLS was tested in the simulation of auto-adaptive continuous (bi-directional cursor control) and discrete multiclass motor imagery (MI) BCI in a tetraplegic patient (Rouanne et al., 2022). In the (Sliwowski et al., 2022) offline study, ANN algorithms were reported to outperform the REW-NPLS decoder. However, these algorithms cannot be applied in real time under CLDA.

Fully adaptive REW-NPLS is compatible with CLDA. However, it may be further improved by integrating real-time adaptive dimension reduction and promoting group-wise decoder sparsity using regularization. Group-wise sparsity, e.g., in the spatial dimension, may allow the elimination of irrelevant or highly correlated electrodes, decreasing computational time and memory consumption at the BCI use stage. This may be critical for portable BCI systems. A sparse solution may be advantageous for small training data sets, preventing overtraining. As BCI decoders require regular updates due to neuronal signal non-stationarity, reducing the decoder training time is desirable for real-life scenarios.

2.2. REW-NPLS

NPLS (Bro, 1996, 1998) estimates a linear relationship between a tensor of independent (input) and a tensor of dependent (output) variables. Given $\underline{\mathbf{X}}_t \in \mathbb{R}^{I_1 \times \dots \times I_m}$ and $\underline{\mathbf{Y}}_t \in \mathbb{R}^{I_1 \times \dots \times I_n}$ the m and n order tensors of the input and output variables at time t , $\underline{\mathbf{Y}}_t = \underline{\mathbf{Beta}}_t + \underline{\mathbf{bias}} + \underline{\mathbf{D}}_t$, where $\underline{\mathbf{Beta}}$ and $\underline{\mathbf{bias}}$ are the tensors of parameters and their associated bias, $\underline{\mathbf{D}}_t \in \mathbb{R}^{I_1 \times \dots \times I_n}$ is the tensor of noise. The parameters are estimated from the training dataset $\{\underline{\mathbf{X}}, \underline{\mathbf{Y}}\}$, $\underline{\mathbf{X}} \in \mathbb{R}^{L \times I_1 \times \dots \times I_m}$, $\underline{\mathbf{Y}} \in \mathbb{R}^{L \times I_1 \times \dots \times I_n}$, L is the training dataset size. NPLS constructs the linear regression iteratively by projecting tensors of observation $\underline{\mathbf{X}}$ and $\underline{\mathbf{Y}}$ to the space of latent variables

using tensor decomposition: $\underline{\mathbf{X}} = \sum_{f_i=1}^f \mathbf{r}_{f_i} \circ \mathbf{w}_{f_i}^1 \circ \dots \circ \mathbf{w}_{f_i}^m + \underline{\mathbf{E}}_x$,

$\underline{Y} = \sum_{f_i=1}^f \mathbf{u}_{f_i} \circ \mathbf{q}_{f_i}^1 \circ \dots \circ \mathbf{q}_{f_i}^n + \underline{E}_y$. Here, “ \circ ” is the outer product operator (Eliseyev et al., 2017), $\mathbf{R}_f = [\mathbf{r}_1, \dots, \mathbf{r}_f] \in \mathbb{R}^{L \times f}$ and $\mathbf{U}_f = [\mathbf{u}_1, \dots, \mathbf{u}_f] \in \mathbb{R}^{L \times f}$ are matrices of the latent variables, $W^{f_i} = \left\{ \mathbf{w}_{f_i}^i \right\}_{i=1}^m$, $\mathbf{w}_{f_i}^i \in \mathbb{R}^{I_i}$ and $Q^{f_i} = \left\{ \mathbf{q}_{f_i}^i \right\}_{i=1}^n$, $\mathbf{q}_{f_i}^i \in \mathbb{R}^{J_i}$, $f_i = 1, \dots, f$ are the projection vectors constructed at iteration f_i , and $\underline{E}_x, \underline{E}_y$ are the residual tensors. The set of projectors is constructed iteratively, increasing latent variable space dimensions.

REW-NPLS (Eliseyev et al., 2017) update a set of F ($F \in \mathbb{N}^*$ is the fixed upper bound latent space dimension) models $\left\{ \underline{\mathbf{Beta}}_{UI}^f, \underline{\mathbf{bias}}_{UI}^f \right\}_{f=1}^F$ using current blocks of tensors of observation $\left\{ \underline{\mathbf{X}}_{UI}, \underline{\mathbf{Y}}_{UI} \right\}$ and previously computed models. Here, $UI \in \mathbb{N}$ is the updated iteration number, $\underline{\mathbf{Beta}}_{UI}^f \in \mathbb{R}^{(I_1 \times \dots \times I_M) \times (J_1 \times \dots \times J_N)}$, $\underline{\mathbf{bias}}_{UI}^f \in \mathbb{R}^{I_1 \times \dots \times I_N}$ are current update of model coefficients, and $\underline{\mathbf{X}}_{UI} \in \mathbb{R}^{\Delta L \times I_1 \times \dots \times I_M}$, $\underline{\mathbf{Y}}_{UI} \in \mathbb{R}^{\Delta L \times J_1 \times \dots \times J_N}$ are the current input and output tensors of observations. $\Delta L \in \mathbb{N}^*$ is the number of samples recorded between the update blocks $UI - 1$ and UI .

The REW-NPLS algorithm is a generalization of the REW-PLS algorithm to tensor variables and belongs to the family of kernel PLS algorithms (Eliseyev et al., 2017). It evaluates a set of projectors and model coefficients from covariance tensors $\underline{\mathbf{XY}} = \underline{\mathbf{X}} \times_1 \underline{\mathbf{Y}}$, $\underline{\mathbf{XY}} \in \mathbb{R}^{(I_1 \times \dots \times I_M) \times (J_1 \times \dots \times J_N)}$, and $\underline{\mathbf{XX}} = \underline{\mathbf{X}} \times_1 \underline{\mathbf{X}}$, $\underline{\mathbf{XX}} \in \mathbb{R}^{(I_1 \times \dots \times I_M) \times (I_1 \times \dots \times I_M)}$. Here, “ \times_k ” is the k -mode tensor product (Eliseyev et al., 2017). First, a set of input variable projectors W are evaluated from the covariance tensor $\underline{\mathbf{XY}}$. The projectors are estimated using a rank one decomposition of the tensor $\underline{\mathbf{V}} = \underline{\mathbf{XY}} \in \mathbb{R}^{I_1 \times \dots \times I_M}$ in the case of single output. For higher dimensions, the eigenvector with the largest eigenvalue is computed from the covariance tensor $\underline{\mathbf{XY}}$ to decrease the dimension of the tensor to decompose: $\mathbf{e} = \text{eig}(\underline{\mathbf{XY}}^T \underline{\mathbf{XY}})$, $\underline{\mathbf{V}} = \text{reshape}(\underline{\mathbf{XY}} \cdot \mathbf{e})$, $\underline{\mathbf{V}} \in I_1 \times \dots \times I_M$. Here, $\underline{\mathbf{XY}} \in \mathbb{R}^{(I_1 \times \dots \times I_M) \times (J_1 \times \dots \times J_N)}$ is the unfolded tensor $\underline{\mathbf{XY}}$, and $\mathbf{e} = \text{eig}()$ is an eigenvector with the largest eigenvalue. Output projectors and the model parameters $\underline{\mathbf{Beta}}$ and $\underline{\mathbf{bias}}$ are computed using W and covariance tensors $\underline{\mathbf{XX}}$ and $\underline{\mathbf{XY}}$. The projectors sets and model parameters are evaluated sequentially, increasing the latent variables’ space dimension in the internal REW-NPLS iterations f .

Finally, at the UI th update, the covariance tensors $\underline{\mathbf{XX}}_{UI}$ and $\underline{\mathbf{XY}}_{UI}$ are computed from the previous $\underline{\mathbf{XX}}_{UI-1}$ and $\underline{\mathbf{XY}}_{UI-1}$ tensors, and the current block of observations $\left\{ \underline{\mathbf{X}}_{UI}, \underline{\mathbf{Y}}_{UI} \right\}$:

$$\underline{\mathbf{XX}}_{UI} = \mu_1 \underline{\mathbf{XX}}_{UI-1} + \underline{\mathbf{X}}_{UI} \times_1 \underline{\mathbf{X}}_{UI},$$

$$\underline{\mathbf{XY}}_{UI} = \mu_1 \underline{\mathbf{XY}}_{UI-1} + \underline{\mathbf{X}}_{UI} \times_1 \underline{\mathbf{Y}}_{UI}$$

μ_1 is a forgetting factor. A set of projectors is evaluated using a rank-one tensor decomposition (Eliseyev et al., 2017) of the current tensor $\underline{\mathbf{V}}_{UI}$.

In the REW-NPLS algorithm, only the covariance tensors, the normalization coefficients, and the current model are stored together with the current block of observations collected since the previous update.

2.3. PARAFAC procedure in REW-NPLS

Several tensor decomposition strategies were designed: the Parallel factor analysis (PARAFAC), Tucker, multilinear SVD decomposition, and so forth (Cichocki et al., 2015). Similar to generic NPLS, the REW-NPLS algorithm employs PARAFAC tensor decomposition (Eliseyev et al., 2017). PARAFAC or CANDECOMP/PARAFAC (CP), also known as polyadic decomposition (PD), can be considered the generalization of principal component analysis (PCA) and singular value decomposition (SVD) to the tensor case (Sheikhhattar et al., 2015; Sharghian et al., 2019). This method represents a M -order tensor $\underline{\mathbf{V}} \in \mathbb{R}^{I_1 \times \dots \times I_m}$ as the linear combination of vectors’ outer products (rank-one tensors) such as follows:

$$\underline{\mathbf{V}} = \sum_{r=1}^R \rho_r \mathbf{w}_r^1 \circ \mathbf{w}_r^2 \circ \dots \circ \mathbf{w}_r^m + \underline{\mathbf{E}}$$

$$\|\mathbf{w}_r^i\| = 1, r = 1, \dots, R; i = 1, \dots, m.$$

Here, $1 \leq i \leq m$ corresponds to the i^{th} mode/dimension of the tensor variable, “ \circ ” is the (vector) outer product of the decomposition factors (projectors) $\mathbf{w}_r^i \in \mathbb{R}^{I_i}$, $R \in \mathbb{N}$ is the number of rank-one tensors used for decomposition, ρ_r is the weight associated with each rank-one tensor of the decomposition and $\underline{\mathbf{E}} \in \mathbb{R}^{I_1 \times \dots \times I_m}$ is the residual tensor (Kolda and Bader, 2009). PARAFAC evaluates the projectors, minimizing the residuals.

Similar to generic NPLS, only one step of PARAFAC ($R = 1$) is applied to the current tensor decomposition at each internal iteration $f_i = 1, \dots, f$ of REW-NPLS:

$$\underline{\mathbf{V}} = \rho_r \mathbf{w}^1 \circ \mathbf{w}^2 \circ \dots \circ \mathbf{w}^m + \underline{\mathbf{E}}$$

$$\|\mathbf{w}^i\| = 1, i = 1, \dots, m.$$

To solve the optimization problem, the alternating least squares (ALS) algorithm is employed in REW-NPLS. ALS optimizes one projector iteratively at a time and fixes others, reducing, at each iteration of ALS, the optimization problem to a least-squares linear regression (Kolda and Bader, 2009; Cichocki et al., 2015; Pereira Da Silva et al., 2015).

REW-NPLS includes several iterations inside other iterative procedures: ALS iterations for PARAFAC ($R = 1$) decomposition, internal REW-NPLS iterations increasing latent variables space dimension, and, finally, the update iterations UI .

2.4. L_p -Penalized REW-NPLS (PREW-NPLS)

Sparse input variables projectors may result in a sparse model. Variable excluded from all the projectors W^{f_i} , $f_i = 1, \dots, f$ is excluded from consideration. As for the tensor data, the projection is made according to the mode of analysis (e.g., special, frequency,

or temporal). The sparsity of the projectors may allow excluding the slices of data from the model (e.g., exclude non-informative or redundant electrodes). To achieve sparse projectors, the proposed PREW-NPLS algorithm employed a penalized version of the PARAFAC ($R = 1$), introducing sparsity-promoting penalization. $L_p, p = 0, 0.5, 1$ penalization, being the classic lasso regularization (L_1) or less conventional L_0 and $L_{0.5}$ penalization is studied. This section describes L_p -Penalization in PARAFAC ($R = 1$) and its integration into the REW-NPLS algorithm to build a new sparsity-promoting adaptive PREW-NPLS algorithm.

2.4.1. L_p -penalized PARAFAC ($R = 1$)

To simplify the notations without losing generality, a case of a three-order tensor is considered in the section. Given a three-order tensor, one step of PARAFAC ($R = 1$) applied at each iteration of REW-NPLS solves the optimization problem:

$$\min_{\mathbf{w}^1, \mathbf{w}^2, \mathbf{w}^3} \left\| \underline{\mathbf{V}} - \rho \mathbf{w}^1 \circ \mathbf{w}^2 \circ \mathbf{w}^3 \right\|^2, \quad (1)$$

$$\|\mathbf{w}^1\| = \|\mathbf{w}^2\| = \|\mathbf{w}^3\| = 1,$$

$\underline{\mathbf{V}} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ and $\mathbf{w}^i \in \mathbb{R}^{I_i}, i = 1, 2, 3$. As the norms of the projectors are arbitrary values in (1), the decomposition vectors are evaluated, minimizing $\underline{\mathbf{V}} - \mathbf{w}^1 \circ \mathbf{w}^2 \circ \mathbf{w}^3$ before being normalized. The optimization problem (1) is solved using the ALS procedure. At each step, ALS fixes two of the three vectors $\mathbf{w}^1, \mathbf{w}^2, \mathbf{w}^3$ reducing the problem to a linear least-squares optimization, e.g. $\mathbf{w}^2, \mathbf{w}^3$ are fixes to approximate \mathbf{w}^1 and then \mathbf{w}^2 is evaluated fixing $\mathbf{w}^1, \mathbf{w}^3$ etc. until convergence (Uschmajew, 2015):

$$\min_{\mathbf{w}^1} \left\| \underline{\mathbf{V}}_{(1)} - \mathbf{w}^1 (\mathbf{w}^3 \otimes \mathbf{w}^2)^T \right\|^2,$$

$$\min_{\mathbf{w}^2} \left\| \underline{\mathbf{V}}_{(2)} - \mathbf{w}^2 (\mathbf{w}^3 \otimes \mathbf{w}^1)^T \right\|^2,$$

$$\min_{\mathbf{w}^3} \left\| \underline{\mathbf{V}}_{(3)} - \mathbf{w}^3 (\mathbf{w}^2 \otimes \mathbf{w}^1)^T \right\|^2.$$

Here, $\underline{\mathbf{V}}_{(i)} = (\mathbf{v}_1^1 \dots \mathbf{v}_1^{I_1}) \in \mathbb{R}^{I_1 \times I_2 I_3}$ is the tensor $\underline{\mathbf{V}}$ unfolded according to i -the direction and \otimes is the Kronecker product. Taking into account that $(\mathbf{w}^2 \otimes \mathbf{w}^1)^T \in \mathbb{R}^{I_1 I_2}, (\mathbf{w}^3 \otimes \mathbf{w}^1)^T \in \mathbb{R}^{I_1 I_3}$ and $(\mathbf{w}^3 \otimes \mathbf{w}^2)^T \in \mathbb{R}^{I_2 I_3}$ are vectors, the optimization tasks are separated into element-wise optimizations:

$$\min_{\mathbf{w}_1^j} \left\| \mathbf{v}_1^j - \mathbf{w}_1^j (\mathbf{w}^3 \otimes \mathbf{w}^2)^T \right\|^2 \quad j = 1, \dots, I_1, \quad (2)$$

$$\min_{\mathbf{w}_2^j} \left\| \mathbf{v}_2^j - \mathbf{w}_2^j (\mathbf{w}^3 \otimes \mathbf{w}^1)^T \right\|^2 \quad j = 1, \dots, I_2, \quad (3)$$

$$\min_{\mathbf{w}_3^j} \left\| \mathbf{v}_3^j - \mathbf{w}_3^j (\mathbf{w}^2 \otimes \mathbf{w}^1)^T \right\|^2 \quad j = 1, \dots, I_3. \quad (4)$$

Here, $\mathbf{w}^1 = (\mathbf{w}_1^1, \dots, \mathbf{w}_1^{I_1})^T \in \mathbb{R}^{I_1}, \mathbf{w}^2 = (\mathbf{w}_1^2, \dots, \mathbf{w}_1^{I_2})^T \in \mathbb{R}^{I_2}$, and $\mathbf{w}^3 = (\mathbf{w}_1^3, \dots, \mathbf{w}_1^{I_3})^T \in \mathbb{R}^{I_3}$. The least square (LS) solutions of (2)–(4) are:

$$\left(\mathbf{w}_1^j \right)_{LS} = \frac{\mathbf{v}_1^j (\mathbf{w}^3 \otimes \mathbf{w}^2)}{\|\mathbf{w}^3 \otimes \mathbf{w}^2\|^2}, \quad j = 1, \dots, I_1, \quad (5)$$

$$\left(\mathbf{w}_2^j \right)_{LS} = \frac{\mathbf{v}_2^j (\mathbf{w}^3 \otimes \mathbf{w}^1)}{\|\mathbf{w}^3 \otimes \mathbf{w}^1\|^2}, \quad j = 1, \dots, I_2, \quad (6)$$

$$\left(\mathbf{w}_3^j \right)_{LS} = \frac{\mathbf{v}_3^j (\mathbf{w}^2 \otimes \mathbf{w}^1)}{\|\mathbf{w}^2 \otimes \mathbf{w}^1\|^2}, \quad j = 1, \dots, I_3. \quad (7)$$

In this study, sparse promoting penalization using $L_p, p = 0, 0.5, 1$ norm/pseudo norm is proposed to be integrated into the cost function of PARAFAC ($R = 1$) to provide a slice-wise sparsity to the solution. The optimization task (1) is replaced by:

$$\left\| \underline{\mathbf{V}} - \hat{\underline{\mathbf{V}}} \right\|^2 + P(\mathbf{w}^1, \mathbf{w}^2, \mathbf{w}^3) \rightarrow \min, \quad (8)$$

$$\hat{\underline{\mathbf{V}}} = \rho \mathbf{w}^1 \circ \mathbf{w}^2 \circ \mathbf{w}^3,$$

$$P(\mathbf{w}^1, \mathbf{w}^2, \mathbf{w}^3) = \lambda_1 \|\mathbf{w}^1\|_{q, \mathcal{L}_1} + \lambda_2 \|\mathbf{w}^2\|_{q, \mathcal{L}_2} + \lambda_3 \|\mathbf{w}^3\|_{q, \mathcal{L}_3},$$

$$\|\mathbf{w}^1\| = \|\mathbf{w}^2\| = \|\mathbf{w}^3\| = 1.$$

Here, $\|\mathbf{w}^i\|_{p, \mathcal{L}_i}$ for $p = 0, 0.5, 1$ and $i = 1, 2, 3$ is denoted as

$$\|\mathbf{w}^i\|_{0, \mathcal{L}_i} = \sum_{k \in \mathcal{L}_i} (1 - \delta_{0, \mathbf{w}_k^i}),$$

$$\|\mathbf{w}^i\|_{1, \mathcal{L}_i} = \sum_{k \in \mathcal{L}_i} |\mathbf{w}_k^i|,$$

and

$$\|\mathbf{w}^i\|_{\frac{1}{2}, \mathcal{L}_i} = \sum_{k \in \mathcal{L}_i} \sqrt{|\mathbf{w}_k^i|}.$$

$0 < \lambda_i \leq 1$ are regularization coefficients, the Kronecker delta $\delta_{0, \mathbf{w}_k^i} = 1$ if $\mathbf{w}_k^i = 0$, and $\delta_{0, \mathbf{w}_k^i} = 0$ otherwise. Notably, only

a part of the indices defined by a set $\mathcal{L}_i \subset \{1, 2, \dots, I_i\}$, $i = 1, 2, 3$ are penalized, resulting in selective penalization (Lutay and Khusainov, 2021). A set \mathcal{L}_i may vary depending on the REW-NPLS iteration. Selective penalization is introduced for the integration of L_p -Penalized PARAFAC ($R = 1$) into the REW-NPLS algorithm.

Similar sparse promoting penalization in PARAFAC ($R = 1$) was considered in Eliseyev et al. (2012). However, this study was limited to L_1 -norm penalization. Integrated with the conventional non-adaptive NPLS algorithm, the optimization problem was solved using time consuming numerical optimization, which was then applied offline. In the current manuscript, a more general case of L_p , $p = 0, 0.5, 1$ norm/pseudo norm penalization is considered. In addition, selective penalization is applied for efficient integration of penalized PARAFAC into the REW-NPLS algorithm. Finally, an efficient optimization procedure compatible with real-time online applications is proposed.

The same ALS strategy is applied to complete the optimization task (8). ALS fixed all projectors at each step except one, leading to three successive optimization tasks. The least square (LS) solution w_{LS}^j of the non-regularized problem is used as an initial approximation. Notably, unlike in non-regularized optimization, due to the penalization terms, the norms of the projectors are not arbitrary values anymore. Therefore, the normalization of a current estimate is added to ALS optimization iterations:

$$\min_{\tilde{w}^1} \left(\left\| \mathbf{V}_{(1)} - \tilde{w}^1 (\mathbf{w}^3 \otimes \mathbf{w}^2) \right\|^2 + \lambda_1 \left\| \tilde{w}^1 \right\|_{q, \mathcal{L}_1} \right), \mathbf{w}^1 = \tilde{w}^1 / \left\| \tilde{w}^1 \right\|, \quad (9)$$

$$\min_{\tilde{w}^2} \left(\left\| \mathbf{V}_{(2)} - \tilde{w}^2 (\mathbf{w}^3 \otimes \mathbf{w}^1) \right\|^2 + \lambda_2 \left\| \tilde{w}^2 \right\|_{q, \mathcal{L}_2} \right), \mathbf{w}^2 = \tilde{w}^2 / \left\| \tilde{w}^2 \right\|, \quad (10)$$

$$\min_{\tilde{w}^3} \left(\left\| \mathbf{V}_{(3)} - \tilde{w}^3 (\mathbf{w}^2 \otimes \mathbf{w}^1) \right\|^2 + \lambda_3 \left\| \tilde{w}^3 \right\|_{q, \mathcal{L}_3} \right), \mathbf{w}^3 = \tilde{w}^3 / \left\| \tilde{w}^3 \right\|. \quad (11)$$

It should be noted that all considered regularization functions are decomposed as a sum of element-wise functions. Consequently, similarly to (3)–(5) optimization tasks, (9)–(11) are split into element-wise optimizations:

$$\min_{w_j^1} \left(\left\| \mathbf{v}_1^j - w_j^1 (\mathbf{w}^3 \otimes \mathbf{w}^2) \right\|^2 + \lambda_1 \times g_p(w_j^1) \right), j = 1, \dots, I_1, \quad (12)$$

$$\min_{w_j^2} \left(\left\| \mathbf{v}_2^j - w_j^2 (\mathbf{w}^3 \otimes \mathbf{w}^1) \right\|^2 + \lambda_2 \times g_p(w_j^2) \right), j = 1, \dots, I_2, \quad (13)$$

$$\min_{w_j^3} \left(\left\| \mathbf{v}_3^j - w_j^3 (\mathbf{w}^2 \otimes \mathbf{w}^1) \right\|^2 + \lambda_3 \times g_p(w_j^3) \right), j = 1, \dots, I_3, \quad (14)$$

$$g_p(w_j^i) = \begin{cases} 1 - \delta_{0, w_j^i}, & \text{if } p = 0 \text{ and } w_j^i \in \mathcal{L}_i \\ \left| w_j^i \right|, & \text{if } p = 1 \text{ and } w_j^i \in \mathcal{L}_i \\ \sqrt{\left| w_j^i \right|}, & \text{if } p = 1/2 \text{ and } w_j^i \in \mathcal{L}_i \\ 0, & \text{otherwise} \end{cases}. \quad (15)$$

The particular cases of $L_0, L_{0.5}, L_1$ penalization are given below. Details are presented in the Appendix.

L_0 -penalization. In the case of L_0 -penalization, the penalization term reflects the number of non-zero coefficients. Considering one of the optimization iterations of ALS, e.g. (12), the solution is an element-wise hard thresholding of the least square solution $(w_j^1)_{LS}$, $j = 1, \dots, I_1$ (see Appendix):

$$(w_j^1)_{L_0} = \begin{cases} 0 & \text{if } j \in \mathcal{L}_1 \text{ and } (w_j^1)_{LS} \leq \text{Threshold}_{L_0} \\ (w_j^1)_{LS} & \text{otherwise} \end{cases},$$

where

$$\text{Threshold}_{L_0} = \frac{\sqrt{\lambda_1}}{\|\mathbf{w}^3 \otimes \mathbf{w}^2\|}.$$

$L_{0.5}$ -penalization. In the case of $L_{0.5}$ penalization and considering one of the optimization steps of ALS, e.g., (12), the cost function $CostF_{L_{0.5}}$ to minimize takes the following form:

$$CostF_{L_{0.5}}(w_j^1) = \left\| \mathbf{v}_1^j - w_j^1 (\mathbf{w}^3 \otimes \mathbf{w}^2) \right\|^2 + \lambda_1 \sqrt{\left| w_j^1 \right|} \quad (16)$$

or, equivalently,

$$CostF_{L_{0.5}}(w_j^1) = \left\| \mathbf{w}^3 \otimes \mathbf{w}^2 \right\|^2 \left((w_j^1)_{LS} - w_j^1 \right)^2 + \lambda_1 \sqrt{\left| w_j^1 \right|}, \quad (17)$$

with solution,

$$(w_j^1)_{L_{0.5}} = \begin{cases} 0, & \text{if } j \in \mathcal{L}_1 \text{ and } (w_j^1)_{LS} \leq \text{Threshold}_{L_{0.5}} \\ \text{argmin} \left(CostF_{L_{0.5}}(0), CostF_{L_{0.5}}(\mathcal{B} \cdot (w_j^1)_{LS}) \right), & \text{if } j \in \mathcal{L}_1 \text{ and } (w_j^1)_{LS} > \text{Threshold}_{L_{0.5}} \\ (w_j^1)_{LS}, & \text{otherwise,} \end{cases}$$

where

$$\text{Threshold}_{L_{0.5}} = \frac{3}{4} \left(\frac{\lambda_1}{\left\| \mathbf{w}^3 \otimes \mathbf{w}^2 \right\|^2} \right)^{2/3},$$

and \mathcal{B} is the biggest root of the cubic equation,

$$x(1-x)^2 = C,$$

$$C = \frac{\lambda_1^2}{\left\| 16\mathbf{w}^3 \otimes \mathbf{w}^2 \right\|^4 \left((w_j^1)_{LS} \right)^3},$$

in the interval $[0; 1]$ (see Appendix).

L_1 -penalization. Finally, in the case of L_1 penalization, considering one optimization step of ALS optimization, e.g., (12), the solution is an element-wise soft-thresholding of the least square

solution $(w_j^1)_{LS}$, $j = 1, \dots, I_1$ (see Appendix):

$$(w_j^1)_{L_1} = \begin{cases} 0 & , \text{if } j \in \mathcal{L}_1 \text{ and } (w_j^1)_{LS} \leq \text{Threshold}_{L_1} \\ \text{sign}((w_j^1)_{LS}) \left(|(w_j^1)_{LS}| - \text{Threshold}_{L_1} \right) & , \\ (w_j^1)_{LS} & \text{if } j \in \mathcal{L}_1 \text{ and } (w_j^1)_{LS} > \text{Threshold}_{L_1} \\ & \text{otherwise} \end{cases}$$

where

$$\text{Threshold}_{L_1} = \frac{\lambda_1}{\|w^3 \otimes w^2\|^2}.$$

2.4.2. Integration to PREW-NPLS

A Penalized PARAFAC ($R = 1$) is used in the REW-NPLS algorithm to extract a set of projectors $\{w_f^1 \in \mathbb{R}^{I_1}, w_f^2 \in \mathbb{R}^{I_2}, \dots, w_f^m \in \mathbb{R}^{I_m}\}_{f=1}^F$. For $f = 1$, all projector elements can be potentially penalized: $\mathcal{L}_{j,1} = \{1, 2, \dots, I_j\}$, $j = 1, \dots, m$. After the first set of projectors' extraction, non-zero elements of the projectors correspond to tensor slices already included in the decoding model with non-zero coefficients. For the next iterations, corresponding indexes were removed from a set of indexes to be penalized, resulting in a sequence $\mathcal{L}_{j,2} \subset \mathcal{L}_{j,1}$. A scheme representing the PREW-NPLS algorithm in the case of penalizing one data tensor direction is shown in Figure 1.

2.5. Experiments

This study relies on the neural signal dataset recorded during the online closed-loop BCI clinical experiments. The « BCI and Tetraplegia » clinical trial (NCT02550522, [ClinicalTrials.gov](https://clinicaltrials.gov)) (University Hospital, Grenoble, 2015) was approved by French authorities: National Agency for the Safety of Medicines and Health Products (Agence nationale de sécurité du médicament et des produits de santé, ANSM) with the registration number 2015-A00650-49 and the Ethic Committee for the Protection of Individuals (Comité de Protection des Personnes, CPP) with the registration number 15-CHUG-19. The research activities were carried out in accordance with the guidelines and regulations of the ANSM and the CPP. The patient provided informed consent for the clinical trial and publication as well as to publish the information/image(s) in an online open-access publication. Details of the clinical trial protocol are available in Benabid et al. (2019).

The participant was a 29-year-old right-handed male with traumatic sensorimotor tetraplegia caused by a complete C4–C5 spinal cord injury two years prior to the study. He underwent bilateral implantation of two chronic wireless WIMAGINE implants (Benabid et al., 2019) for ECoG signal recording on 21 June 2017. Two WIMAGINE recording systems were surgically implanted into the skull near the sensory-motor cortex (SMC) through a 25-mm radial craniotomy. Before the surgery, the patients' SMC was clearly localized using functional imaging. Details are provided in Benabid et al. (2019). The WIMAGINE device is made up of an active implantable medical device composed of 64-plane platinum-iridium 90/10 electrodes with a

2.3 mm diameter and a 4–4.5 mm inter-electrode distance (Sauter-Starace et al., 2019). The recorded signals were low- and high-pass filtered, with a bandwidth range of 0.5 Hz to 300 Hz, using analog low-pass filters as well as a digital low-pass FIR filter embedded into the implant hardware. The digitized ECoG data from 32 electrodes from each implant (Figure 2) were radio transmitted to a custom-designed base station at a 586 Hz sampling rate.

Since the implantation, the patient had been trained to control multiple real and virtual effectors, such as games that were specially created for BCI training, a wheelchair, an exoskeleton, an avatar, and so on, using a custom-made BCI platform (Benabid et al., 2019). The database used in the study was recorded during the online closed-loop BCI experiments of upper limbs BCI control. The BCI sessions included alternative active states (AS) of the 3D reaching task for each hand, 1D wrist rotation of each hand, and the idle state (IS) (Benabid et al., 2019). The patient aimed to reach the proposed targets or rotate the wrist to specific angles following pursuit tasks. A pursuit task session was composed of successive tasks, e.g., a left-hand 3D reaching task, a right-hand 3D reaching task, IS, and so on. Each task is composed of several trials in which the cursor must reach the proposed targets. The cursor position is not reset between tasks, during the task, or during the idle state. Twenty-two targets were symmetrically distributed in two cubes in front of the patient. The virtual avatar was used as visual feedback (Benabid et al., 2019). In this study, only the left- and right-hand 3D translation trials were used for algorithm evaluation. The experimental paradigm is illustrated in Figure 2.

Data were recorded in the period of 468–666 days after implantation and included 43 sessions. The average session duration was 29 ± 8 min. During the experiments, the first six experimental sessions were used for incremental real-time decoder updates. REW NPLS decoders for each active state task were integrated into the BCI clinical trial platform. The decoders were initialized with a zero. The total training time of the decoding models was 3 h and 37 min, including a total of 189 and 194 trials of the left- and right-hand translation, respectively. The decoders were calibrated during experimental sessions in late September 2018 and were tested without re-calibration in experiments from early October 2018 to mid-March 2019, with 37 experimental sessions distributed over 5–203 days after the last model update session (468–666 days after implantation; Figure 2).

2.6. Application, feature extraction

During the experimental sessions, neural signal epochs of 1 s for 64 recording channels $X_t \in \mathbb{R}^{586 \times 64}$ with sliding steps of 100 ms used for feature extraction. The epochs were mapped into the feature space using a complex continuous wavelet transform (CCWT) (Morlet) with a frequency range of 10–150 Hz and a 10-Hz step. CCWT is a common feature extraction strategy that is widely used in the field of BCIs (Chao et al., 2010; Shimoda et al., 2012; Schaeffer and Aksenova, 2016; Eliseyev et al., 2017; Choi et al., 2018). The absolute value of CCWT was decimated by averaging along the temporal modality to obtain a 10-point description of 1 s time epoch for each frequency band and for each channel, resulting in the temporal-frequency-spatial neural

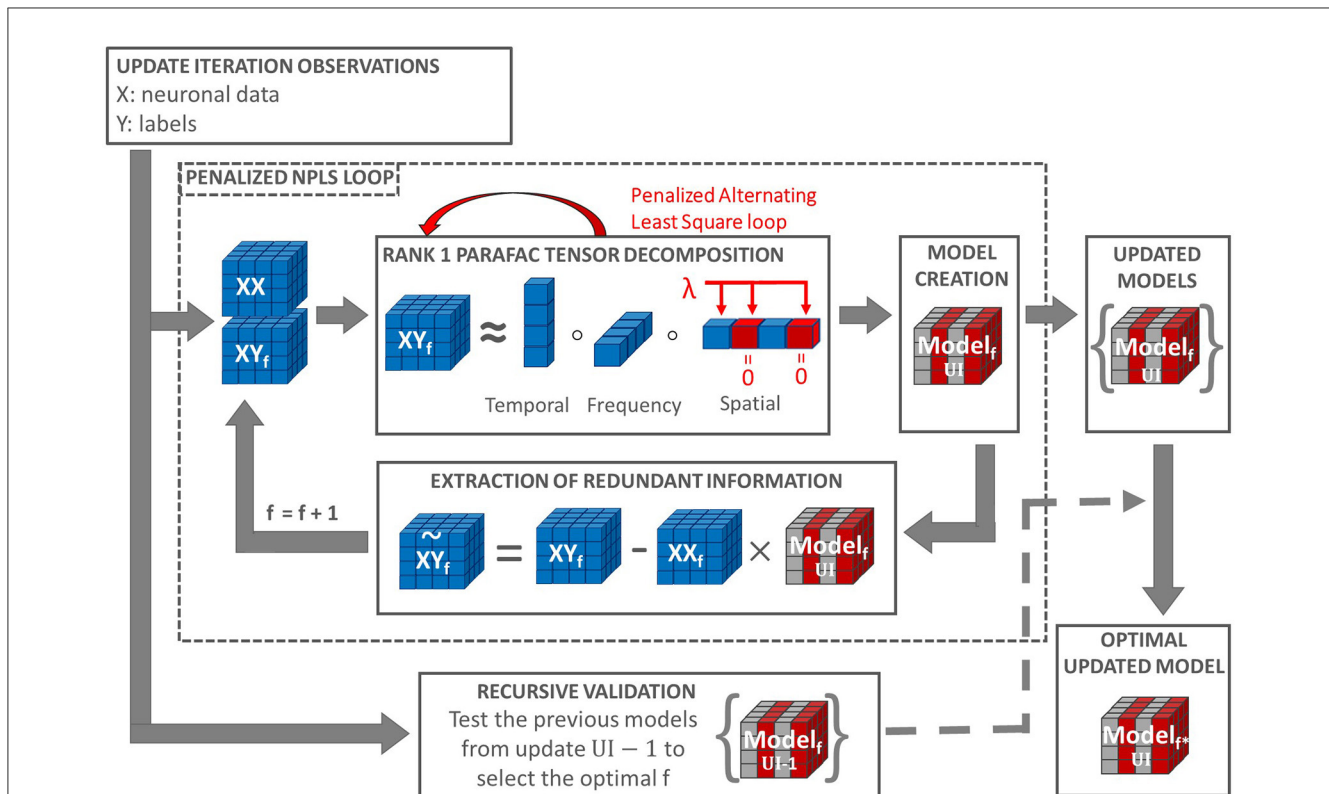


FIGURE 1 Penalized REW-NPLS (PREW-NPLS) algorithm. For the model update iteration U_i , an updated training data set is collected. The update loop includes the recursive validation procedure for the evaluation of optimal latent variable space dimension f (number of factors) and penalized NPLS loop, which includes the sequential tensors XY and XX decomposition. Each step of decomposition uses Rank 1 PARAFAC, which employs penalized alternating least square procedure. Selective sparsity-promoting penalization with penalization parameter λ is applied, e.g., to the spatial dimension. Part of the weights is set to zero due to penalization (shown in red). This results in zero slices in the decoding model (shown in red). A set of updated models for all parameters f are stored up to the maximum f value, 100 in the current setting.

feature tensor $\underline{X}_t \in \mathbb{R}^{10 \times 15 \times 64}$. The observations of neural features \underline{X}_t and the movement features \underline{y}_t recorded during the decoder update/calibration sessions were used for the decoding model identification. The movement features (optimal prediction) at time t is defined as the 3D Cartesian vector between the current effector position and the target position (Benabid et al., 2019). The control command $\hat{\underline{y}}_t$ for 3D hand translation is defined as the cartesian increments of the current position and is sent to the effector at 10 Hz.

2.7. Comparison study

To compare the proposed PREW-NPLS and the generic, non-penalized REW-NPLS algorithms, the decoding of the 3D reaching task for each hand is considered. The comparison study was restricted to penalization according to spatial modality. Three types of penalization $L_p, p = 0, 0.5, 1$ were tested. For $L_p, p = 0.5, 1$ penalties, a set of models were evaluated with the penalization parameter λ going from 0 to 0.5 with 0.02 steps. In the case of the L_0 -PREW-NPLS, preliminary results highlighted that the studied λ range was not relevant. Therefore, the models with the penalization

parameter λ going from 0 to 0.05 with 0.002 steps were estimated. The cases with $\lambda = 0$ correspond to the generic, non-penalized REW-NPLS algorithm.

The performance of algorithms was evaluated offline in a pseudo-online manner. Pseudo-online simulation uses the procedure and parameters of online stream data processing. The dataset was recorded during the online closed-loop BCI experiments using the REW-NPLS decoder previously integrated into the BCI system. As the recorded data integrate the neuronal feedback into real-time decoding, the presented offline comparison study is not fully generalizable to the online case. Nevertheless, it allows the characterization of, to some extent, the studied algorithms.

To be as close as possible to the settings of the online experiments, in the comparison study, the penalized models were calibrated on the same experiments that were used for decoder training during the online closed-loop experiments.

The predicted trajectories performed during the online closed-loop experiments are related to the decoding model used during the experiments. Consequently, sample-based indicators were used to compare the predictions of the tested algorithms in the offline study. The cosine similarity indicator was based on the comparison between the predicted directions $\hat{\underline{y}}_t$ and the optimal direction \underline{y}_t

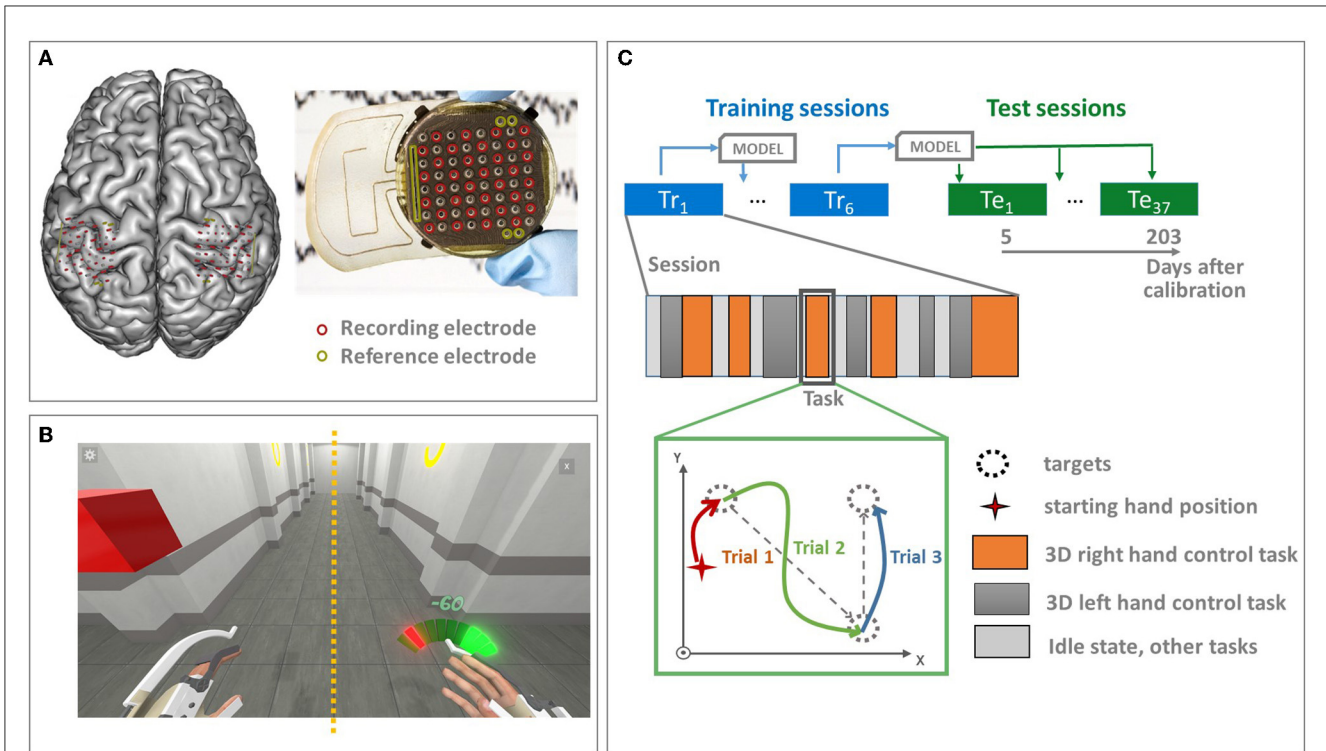


FIGURE 2 Experimental design. **(A)** The patient underwent bilateral implantation of two chronic wireless WIMAGINE implants. The recording systems were epidurally implanted within a 25 mm radial craniotomy in front of the sensory-motor cortex. The WIMAGINE implant is composed of 64 plane electrodes with a 2.3 mm diameter and a 4–4.5 mm inter-electrode distance. The digitized ECoG data from 32 electrodes from each implant (shown in red) are radio transmitted to a base station at the 586 Hz sampling rate. Reference electrodes are shown in green. **(B)** The database was recorded during the online closed-loop BCI experiments of alternative upper limbs BCI control. The virtual avatar was used as visual feedback. **(C)** The data sets were recorded in the period of 468 to 666 days after implantation and includes 43 sessions in general. The first six experimental sessions were used for incremental real-time decoder training. Separate decoders were identified to control different tasks: alternative 3D reaching tasks, and wrists rotation. They were mixed together using the Markov Mixture of expert approach (Moly et al., 2022). The decoders were tested without re-calibration with 37 experimental sessions distributed over 5–203 days after the last model update. Only 3D reaching tasks of both hands are included in the current study. The 3D reaching tasks are composed of a series of trials in which the sequentially proposed targets must be reached (pursuit tasks). The cursor position is not reset during the task, between tasks, and during the idle state.

was employed:

$$CosSim_t = \frac{\mathbf{y}_t \cdot \hat{\mathbf{y}}_t}{\|\mathbf{y}_t\| \|\hat{\mathbf{y}}_t\|}$$

Here, “.” defined the dot product \mathbf{y}_t and $\hat{\mathbf{y}}_t$, and they are the optimal and predicted output. The optimal 0 output is defined as the 3D Cartesian vector between the current position and the target. Notably, $CosSim_t \in [-1, 1]$ represent how direct the movement is to the final destination. A mean cosine similarity of 1 corresponds to a direct and short trajectory. $CosSim$ performance criterion is evaluated as the median, 25th (Q_1) and 75th (Q_3) percentiles of the $CosSim_t$ over samples and is notated as $CosSim = \text{median}(Q_1 - Q_3)$.

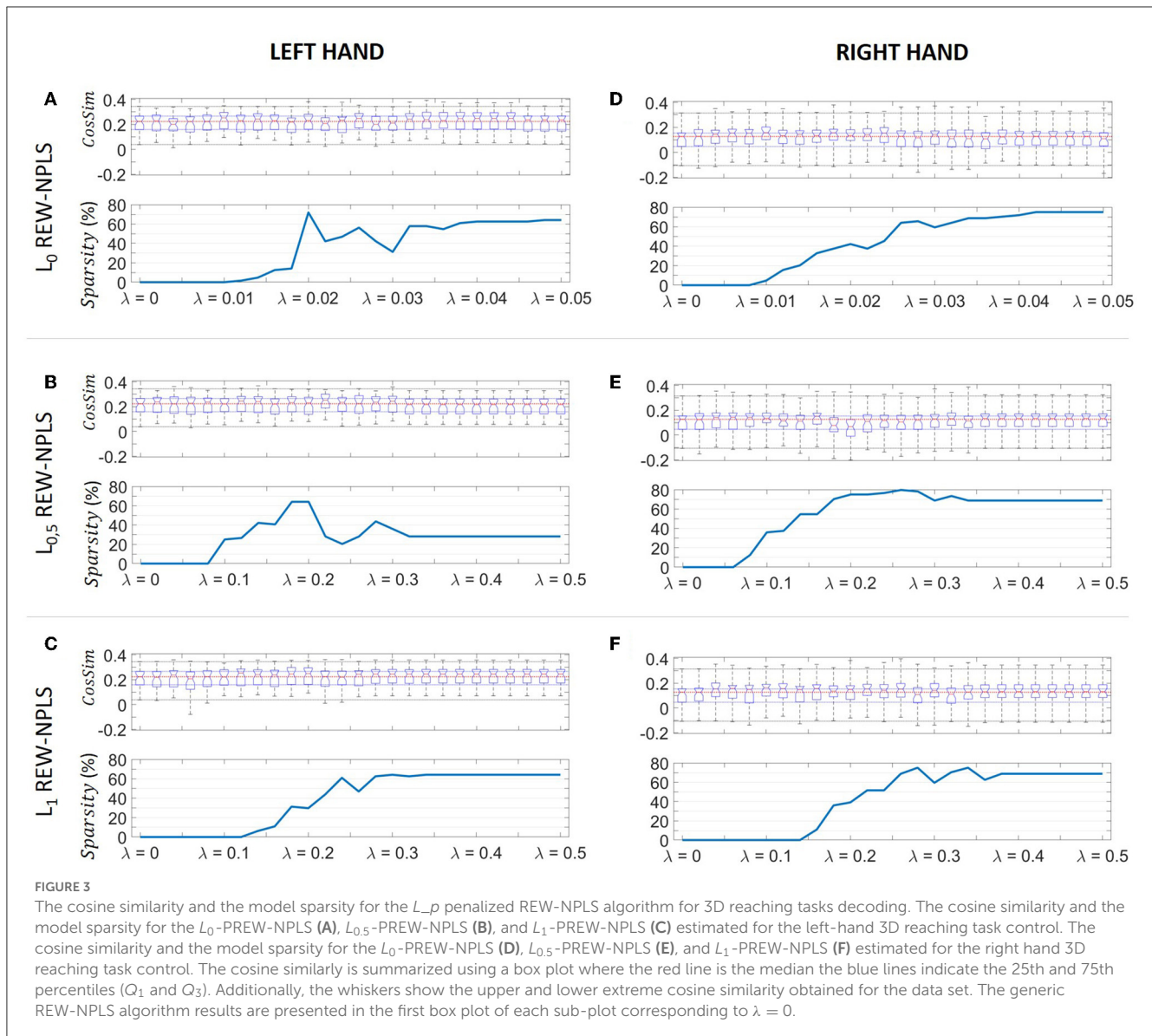
The PREW-NPLS algorithms converge to sparse solutions, fixing non-relevant model coefficients to exactly zero. The decoding performance, therefore, is not the only relevant indicator. Considering a penalized model with the penalization restricted to the i th dimension, the model *Sparsity* index for i th dimension is introduced as the percentage of slices according to i th dimension of the tensor of model coefficients fixed fully to zero. For example, if the model is penalized according to the spatial dimension,

minimizing the number of electrodes involved, *Sparsity* index for the spatial dimension corresponds to the percentage of electrodes fully excluded from the decoder (zero model coefficient for these electrodes for all frequencies and all time delays).

The significance of the differences in the cosine similarity between the REW-NPLS and PREW-NPLS algorithms was computed for the left and right-hand 3D translation studies and for each penalization parameter λ . The statistical analysis was performed with the non-parametric paired Wilcoxon signed rank test with ($\alpha_{multi-class} = 0.00161$) and without ($\alpha = 0.05$) the Bonferroni correction.

3. Results

The cosine similarity performance and the *Sparsity* of models with $L_p, p = 0, 0.5, 1$ penalization according to spatial modality for the left and right-hand 3D movement tasks are presented depending on the penalization coefficient λ in Figure 3. The generic REW-NPLS ($\lambda = 0$) performance is presented in the first position of each sub-figure.



The generic REW-NPLS ($\lambda = 0$) performance highlighted a $CosSim = 0.223$ ($0.158 - 0.266$) (median = 0.223, $Q_1 = 0.158$, $Q_3 = 0.266$), for the left hand, and $CosSim = 0.127$ ($0.047 - 0.155$) (median = 0.127, $Q_1 = 0.047$, $Q_3 = 0.155$) for the right-hand 3D translation decoding.

The penalized decoders showed relevant performance for different penalization parameters λ . *Sparsity* indexes generally increase with higher penalization coefficients. Decoding performance for all penalty types and penalization parameters were highly variable between sessions (high inter-session variability). Decoding of the right-hand translation stresses worse cosine similarity than decoding of the left-hand, but PREW-NPLS allowed for better decoding performance than generic REW-NPLS.

L_0 PREW-NPLS algorithm demonstrated improved performance compared to the generic REW-NPLS for different parameters λ in both hand translation and decoding tasks (Figures 3A, D). For a minor penalization, $\lambda = 0.01$, low sparsity was observed: *Sparsity* = 0% for the left hand, and

Sparsity = 4.68% corresponding to only three electrodes removed for the right hand. Additionally, the performance reached $CosSim = 0.252$ ($0.165 - 0.296$) for the left hand and $CosSim = 0.157$ ($0.1018 - 0.203$) for the right hand translation, resulting in a median movement of the cosine similarity of 13 and 24%, respectively.

Performance improvement was demonstrated for sparser solutions using different penalization parameters. For a moderate penalty, $\lambda = 0.026$, cosine similarity $CosSim = 0.248$ ($0.173 - 0.288$) was achieved with a *Sparsity* = 56.25% for the left hand. For the right hand, with $\lambda = 0.018$, cosine similarity $CosSim = 0.157$ ($0.0989 - 0.185$) was achieved with *Sparsity* = 37.5%. A sparser solution with *Sparsity* = 45.31% was obtained for the right hand for $\lambda = 0.024$, with $CosSim = 0.153$ ($0.0786 - 0.198$).

For a strong penalty ($0.04 \leq \lambda \leq 0.046$), L_0 PREW-NPLS demonstrated for the left hand a similar decoding performance $CosSim = 0.248$ ($0.162 - 0.294$) with 40 electrode parameters set to the 0 value, *Sparsity* = 62.5%. For the right hand, L_0

PREW-NPLS models converged to a highly sparse solution with $\lambda > 0.046$ with 48 electrodes over 64 removed (*Sparsity* = 75%), which highlighted decoding performance similar to the generic REW-NPLS [*CosSim* = 0.128 (0.058 – 0.168)].

Similarly, the ***L_0.5*** PREW-NPLS decoder demonstrated better decoding performance with sparser solutions for several penalization parameters compared to the generic REW-NPLS decoder (Figures 3B, E).

In particular, for a moderate penalty, $\lambda = 0.22$, for the left hand, 18 electrode parameter weights were set to zero (*Sparsity* = 28.13%), achieving higher performance *CosSim* = 0.253 (0.189 – 0.301) compared to the generic REW-NPLS model. Similarly, for the right hand, for penalization $\lambda = 0.16$, a sparse model (*Sparsity* = 54.69%) was identified, increasing the decoding performance *CosSim* = 0.150 (0.0881 – 0.176). Sparsest models were obtained for $\lambda = 0.26, 0.28$ with *Sparsity* = 79.69% and *Sparsity* = 78.13%, respectively.

With higher penalization, for the left hand, the performance *CosSim* = 0.245 (0.156 – 0.2838) was achieved with *Sparsity* = 35.94% for $\lambda = 0.3$. For $\lambda \geq 0.32$, the sparsity converged to *Sparsity* = 28.13% with decoding performance similar to the generic REW-NPLS model: *CosSim* = 0.217 (0.143 – 0.261). For the right hand, for $\lambda \geq 0.36$, the models converged to the same solutions with *Sparsity* = 68.75% (44 electrodes removed) and a cosine similarity *CosSim* = 0.131 (0.0835 – 0.186) slightly superior compared to the generic REW-NPLS decoder.

L_1 PREW-NPLS decoder demonstrated similar results (Figures 3C, F). For low penalty, $\lambda = 0.12$, *Sparsity* = 0%, the *L_1* PREW-NPLS model highlighted *CosSim* = 0.253 (0.151 – 0.286) for the left hand. For the right hand, the models with small penalization parameters $\lambda = 0.04, 0.06$ and 0.1 with zero sparsity (*Sparsity* = 0%) highlighted a cosine similarity *CosSim* = 0.154 (0.0915 – 0.202), *CosSim* = 0.158 (0.0791 – 0.184) and *CosSim* = 0.164 (0.0959 – 0.191) representing an improvements of 21, 24, and 29%, respectively.

For higher penalization parameters, *Sparsity* = 29.69% was reached for $\lambda = 0.20$ with a decoding performance of *CosSim* = 0.249 (0.162 – 0.295) for the left hand. Similarly, for the right hand, the decoding performance *CosSim* = 0.154 (0.101 – 0.192) and *CosSim* = 0.152 (0.0872 – 0.197) with 33 (51.56%) and 44 (68.75%) electrode parameters weights set to zero was achieved for $\lambda = 0.22, 0.26$, respectively.

Finally, for the strong penalization, $\lambda \geq 0.34$, 41 electrode parameter weights were set to zero leading to *CosSim* = 0.245 (0.173 – 0.283) for the left-hand. For penalization parameters $\lambda \geq 0.38$, the models stabilized to a solution with *Sparsity* = 68.75% with a decoding performance *CosSim* = 0.131 (0.0835 – 0.186) for the right hand.

The REW-NPLS and the *L_p* PREW-NPLS model parameter weights were shown in the temporal, frequency, and spatial domains in Supplementary Figures 1, 2 for the left- and right-hand translation models, respectively. The presented models were the ones with the maximal penalization coefficient $\lambda = 0.06, 0.4$ and 0.4 for the *L₀*-, *L_{0.5}*-, and *L₁* PREW-NPLS algorithms, respectively. The spatial parameter weights are presented in Supplementary Figures 3, 4 on a map with the electrode locations relative to the sensory (SS) and motor (MS) sulci.

The results of the statistical analysis performed with the non-parametric paired Wilcoxon signed rank test with and without the Bonferroni correction are presented in Supplementary Tables 1, 2 for the left- and right-hand translation and decoding for each of the PREW-NPLS algorithms. Numerous models highlighted the statistically significant performance improvement of the proposed PREW-NPLS algorithms compared to the generic REW-NPLS for all the penalization types.

4. Discussion

Among the various potential applications, the functional compensation/restoration of individuals suffering from severe motor disabilities has always been a focus of BCI research. The primary challenge of motor BCIs is the high-dimensional control of complex effectors. To achieve this objective, high-resolution neuronal activity recording is generally required, which results in the high-dimensional data flow being processed in real time with a high decision rate of at least (8–10 Hz). Moreover, closed-loop decoder training is reported to be more effective (Jarosiewicz et al., 2013) compared to decoders trained classically using training data recorded in an open-loop setting. CLDA BCIs use incremental learning/adaptive algorithms capable of updating a decoder in real time during BCI sessions, increasing the computational load. In addition to high computing time and power requirements, high-dimensional feature space may lead to numerous issues, complicating decoder training. Efficient feature selection decreases the feature space dimension and may allow for improving the generalization ability of the decoder by excluding irrelevant, redundant, or noisy variables. As neuronal activity features are naturally grouped (e.g., by the sensors and the frequency bands), group-wise feature selection may be beneficial, which can help fully exclude non-informative or redundant electrodes or/and frequency bands from consideration. A group-wise feature selection procedure embedded in adaptive/incremental learning of the decoder is proposed in the study. It was based on sparsity-promoting penalization of the data tensor decomposition integration to the REW-NPLS algorithm. *L_p*, *p* = 0, 0.5, 1 norm/pseudo-norm penalties were studied. Sparsity-promoting penalties were applied to feature groups corresponding to slices of the data tensor related to the mode of analysis (e.g., spatial).

The proposed algorithms were tested with a dataset of left/right arms 3D translations of a virtual avatar controlled by the tetraplegic subject. The studied models were trained on the same data (the first six sessions), which were used for decoder training during the online experiments. The number of training sessions was relatively small (14%) and focused at the beginning of the experiments. This may partially explain the high inter-session variability of the decoding performance for all the algorithms (depicted in Supplementary Figure 5). However, *L_p*-PREW-NPLS algorithms highlighted equal or better decoding performance compared to the generic REW-NPLS decoder with sparse solutions, with up to two-thirds of the electrode parameter weights set to zero for both tasks of the left and the right-hand translation. Computational load measurements demonstrated the expected decrease (close to linear) in memory consumption and computational time during

the model execution stage, depending on the model sparsity for 3D hand translation. 25, 50, and 75% of sparsity in the spatial dimension results in an average of 25, 50, and 75% decreases in memory consumption and a 23, 45, and 67% decrease in computational time.

The PREW-NPLS decoders highlighted equal or better decoding performance compared to the generic REW-NPLS algorithm for the majority of the penalization parameters. For example, for the right-hand decoding, the L_1 -PREW-NPLS decoder with a moderate penalization parameter $\lambda = 0.26$, highlighted a significant (p -value = 10^{-6}) cosine similarity improvement of 21, 116, and $[[\text{Mathtype-mtef1-eqn-317.mtf}]]$ for the median and the 25th and 75th percentiles, respectively, with fewer than half of the electrodes, maintained at a non-zero value (33 electrodes set to zero). For the small penalization parameters ($\lambda = 0.1$), the model converged to a non-sparse solution with a significant cosine similarity improvement (p -value = 10^{-6}), leading to a median, 25th and 75th percentile enhancement of 24, 104, and 23%, respectively. The sparsest solution with the L_1 -PREW-NPLS algorithm removed 75% of the electrodes without decreasing the cosine similarity indicator and reducing the features space from $10 \times 15 \times 64 = 9,600$ features to $10 \times 15 \times 16 = 2,400$ features.

Decoding performance improvements were more evident for the 3D right-hand translation models than the left-hand 3D translation. As the 3D right-hand translation models demonstrated lower decoding performance than left-hand translation models, a sparse decoder is more effective for a less performant neural signal decoder in this study.

An increase in the training dataset size may potentially improve decoder performance and limit overtraining. **Supplementary Figure 6** shows the cosine similarity and model sparsity using L_1 -PREW-NPLS with larger training datasets: 30 and 50% of the whole data (13 and 22 sessions, respectively). It results in fewer sparse models and an improvement in the decoder's performance (**Supplementary Figure 6**). However, in BCI research, the recording of data is highly expensive, limiting access to big datasets. The decoder training method with shorter recordings is highly profitable for real-life BCI applications.

The relative efficiency of penalization types is unclear in the current study. The maximum improvement in performance was achieved using L_1 penalization, but the results are comparable to other penalization types. Application $L_{0.5}$ penalty is computationally heavier compared to L_0 or L_1 penalties without evident advantages. A complimentary comparative study may allow clarification of the relative efficiency of the proposed penalizations.

L_0 -PREW-NPLS and L_1 -PREW-NPLS algorithms are sufficiently computationally effective to be applied for the online closed-loop decoder adaptation. To reproduce the online experimental conditions, a pseudo-online simulation was conducted using the same parameters (buffer size, batch training, and so on) as used in real time to ensure the compatibility of the proposed algorithm with the real-time application.

Finally, reducing the feature space dimension may lead to more interpretable models. Several activation patterns were discernible. In the spatial domains, the electrodes close to the motor and sensory sulci exhibit important parameter weights.

The current study is limited to offline simulations using datasets recorded from previous online closed-loop experiments with the REW-NPLS decoder integrated into the BCI system. As the neuronal signal integrates the neuronal feedback to the decoder, which was applied in real time, the offline comparison study is not fully generalizable. However, conducting real experiments to compare all algorithms is too costly. In addition, as the neuronal feedback is related to the real-time decoder output, the comparison results are likely to be biased in favor of the decoder applied during real-time experiments rather than the newly tested algorithms.

The study presented in the study is limited to penalization in the spatial domain, which is crucial for decreasing data transfer. In order to evaluate the influence on the decoding performance of penalization applied to other modalities, further research needs to be conducted to include decoders that promote sparsity in the frequency and temporal domains.

This study reports a case study. The results are obtained from the dataset, which was based on a single-subject clinical trial. Further tests with more subjects are required to support the conclusions. The authors plan to contribute to conducting clinical trials to assess the proposed algorithm in real-time experiments.

Data availability statement

The data analyzed in this study is subject to the following licenses/restrictions. Data used for this study are from recordings from the clinical trial ([ClinicalTrials.gov](https://clinicaltrials.gov/ct2/show/study/NCT02550522), NCT02550522) and are not publicly available. Part of the dataset may be provided upon reasonable request. Requests to access these datasets should be directed at: tetiana.aksenova@cea.fr.

Ethics statement

The studies involving human participants were reviewed and approved by National Agency for the Safety of Medicines and Health Products (Agence nationale de sécurité du médicament et des produits de santé: ANSM), registration number 2015-A00650-49, and the Committee for the Protection of Individuals (Comité de Protection des Personnes—CPP), registration number 15-CHUG-19. The patients/participants provided their written informed consent to participate in this study.

Author contributions

AM, AA, and TA: designed the algorithm. AM and AA: helped with using software. AM and FM: analyzed the data. AM, AA, FM, and TA: wrote the manuscript. All authors contributed to the article and approved the submitted version.

Funding

CLINATEC[®] is a Laboratory of CEA-Grenoble and has statutory links with the University Hospital of Grenoble (CHUGA) and with the University Grenoble Alpes (UGA). This study was funded by CEA (recurrent funding) and the French Ministry of

Health (Grant PHRC-15-15-0124), Institut Carnot, and Fonds de Dotation CLINATEC[®]. The Foundation Philanthropique Edmond J. Safra is a major funding institution for the CLINATEC[®] Edmond J. Safra Biomedical Research Center.

Acknowledgments

We thank Thomas Costecalde, Serpil Karakas, Olivier Faivre, Alexandre Verney, Benoit Milville, Guillaume Charvet, Stéphan Chabardes, Alim-Louis Benabid, and all the team members working at CLINATEC[®] for the help in the experimental setup and clinical trial experiments.

Conflict of interest

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.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Supplementary material

The Supplementary Material for this article can be found online at: <https://www.frontiersin.org/articles/10.3389/fnhum.2023.1075666/full#supplementary-material>

SUPPLEMENTARY TABLE 1

Significance of the performance (*CosSim median*) differences between the REW-NPLS and the L_0, L_0.5, and L_1 PREW-NPLS algorithms for the left hand 3D translation study using the non-parametric paired Wilcoxon signed rank test with and without the Bonferroni correction ($\alpha = 0.05$ and $\alpha_{Bonferroni} = 0.00161$).

SUPPLEMENTARY TABLE 2

Significance of the performance (*CosSim median*) differences between the REW-NPLS and the L_0, L_0.5, and L_1 PREW-NPLS algorithms for the right hand 3D translation study using the non-parametric paired Wilcoxon signed rank test with and without the Bonferroni correction ($\alpha = 0.05$ and $\alpha_{Bonferroni} = 0.00161$).

SUPPLEMENTARY FIGURE 1

Parameter weights of the generic REW-NPLS and L_p , $p = 0, 0.5, 1$ PREW-NPLS for the left-hand 3D translation. Absolute values of model coefficients are projected into spatial, frequency, or temporal domains. The parameter weights related to the three controlled axes (y_1 , y_2 , and y_3) are represented using blue, orange, and yellow lines, respectively.

SUPPLEMENTARY FIGURE 2

Parameter weights of the generic REW-NPLS and L_p , $p = 0, 0.5, 1$, PREW-NPLS for the right-hand 3D translation. Absolute values of model coefficients are projected into spatial, frequency, or temporal domains. The parameter weights related to the three controlled axes (y_1 , y_2 , and y_3) are represented using blue, orange, and yellow lines, respectively.

SUPPLEMENTARY FIGURE 3

Parameter weights of the models identified using generic REW-NPLS and L_p , $p = 0, 0.5, 1$, PREW-NPLS algorithms (the left-hand 3D translation study) projected into the spatial domain depending on the electrode locations in the implant. The sensory sulcus (SS) and motor sulcus (MS) are shown by yellow and red curves respectively.

SUPPLEMENTARY FIGURE 4

Parameter weights of the models identified using generic REW-NPLS and L_p , $p = 0, 0.5, 1$, PREW-NPLS algorithms (the right-hand 3D translation study) projected into the spatial domain depending on the electrode locations in the implant. The sensory sulcus (SS) and motor sulcus (MS) are shown by yellow and red curves respectively.

SUPPLEMENTARY FIGURE 5

The cosine similarity for the L_p penalized REW-NPLS algorithms ($p = 0, 0.5, 1$), for different λ for 37 test sessions. Cosine similarity of decoding of left-hand 3D translation is depicted in subplots (A–C). Cosine similarity of decoding of right-hand 3D translation is depicted in subplots (D–F).

SUPPLEMENTARY FIGURE 6

The cosine similarity and the model sparsity for the L_1 penalized REW-NPLS algorithm for 3D reaching tasks using 30% (A, C) and 50% (B, D) of the data set for the decoder training. The cosine similarity is summarized using a box plot where the red line is the median the blue lines indicate the 25th and 75th percentiles (Q_1 and Q_3). Additionally, the whiskers show the upper and lower extreme cosine similarity obtained for the data set. The generic REW-NPLS algorithm results are presented in the first box plot of each sub-plot corresponding to $\lambda = 0$.

References

- Andreu-Perez, J., Cao, F., Hagrais, H., and Yang, G. (2018). A self-adaptive online brain-machine interface of a humanoid robot through a general type-2 fuzzy inference system. *IEEE Trans. Fuzzy Syst.* 26, 101–116. doi: 10.1109/TFUZZ.2016.2637403
- Ang, K. K., Chin, Z. Y., Zhang, H., and Guan, C. (2011). "Filter Bank Common Spatial Pattern (FBCSP) algorithm using online adaptive and semi-supervised learning," in *The 2011 International Joint Conference on Neural Networks* (San Jose, CA), 392–396. doi: 10.1109/IJCNN.2011.6033248
- Bellman, R. E. (1961). *Adaptive Control Processes: A Guided Tour*. Princeton: Princeton University Press. doi: 10.1515/9781400874668
- Benabid, A. L., Costecalde, T., Eliseyev, A., Charvet, G., Verney, A., Karakas, S., et al. (2019). An exoskeleton controlled by an epidural wireless brain-machine interface in a tetraplegic patient: a proof-of-concept demonstration. *Lancet Neurol.* 18, 1112–1122. doi: 10.1016/S1474-4422(19)30321-7
- Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. New York, NY: Springer.
- Bolón-Canedo, V., Sánchez-Marroño, N., and Alonso-Betanzos, A. (2013). A review of feature selection methods on synthetic data. *Knowl. Inf. Syst.* 34, 483–519. doi: 10.1007/s10115-012-0487-8
- Bousseta, R., El Ouakouak, I., Gharbi, M., and Rezagui, F. (2018). EEG based brain computer interface for controlling a robot arm movement through thought. *IRBM* 39, 129–35. doi: 10.1016/j.irbm.2018.02.001
- Brandman, D. M., Hosman, T., Saab, J., Burkhart, M. C., Shanahan, B. E., Ciancibello, J. G., et al. (2018). Rapid calibration of an intracortical brain-computer interface for people with tetraplegia. *J. Neural Eng.* 15, 026007. doi: 10.1088/1741-2552/aa9ee7
- Bro, R. (1996). Multiway calibration. *Multilinear PLS. J. Chemom.* 10, 47–61. doi: 10.1002/(SICI)1099-128X(199601)10:1<&t;47::AID-CEM400&t;3.0.CO;2-C
- Bro, R. (1998). *Multiway Analysis in the Food Industry: Models, Algorithms, and Applications* (Kobenhavns UniversitetKobenhavns Universitet, LUKKET: 2012 Det

- Biovidenskabelige Fakultet for Fødevarer, Veterinærmedicin og Naturressourcer Faculty of Life Sciences, LUKKET: 2012 Institut for Fødevarevidenskab Department of Food Science, LUKKET: 2012 Kvalitet og Teknologi Quality and Technology).
- Brunner, C., Scherer, R., Graimann, B., Supp, G., and Pfurtscheller, G. (2006). Online control of a brain-computer interface using phase synchronization. *IEEE Trans. Biomed. Eng.* 53, 2501–6. doi: 10.1109/TBME.2006.881775
- Bundy, D. T., Pahwa, M., Szrama, N., and Leuthardt, E. C. (2016). Decoding three-dimensional reaching movements using electrocorticographic signals in humans. *J. Neural Eng.* 13, 026021. doi: 10.1088/1741-2560/13/2/026021
- Cantillo-Negrete, J., Carino-Escobar, R. I., Carrillo-Mora, P., Elias-Vinas, D., and Gutierrez-Martinez, J. (2018). Motor imagery-based brain-computer interface coupled to a robotic hand orthosis aimed for neurorehabilitation of stroke patients. *J. Healthc. Eng.* 2018, 1624637. doi: 10.1155/2018/1624637
- Chao, Z. C., Nagasaka, Y., and Fujii, N. (2010). Long-term asynchronous decoding of arm motion using electrocorticographic signals in monkey. *Front. Neuroeng.* 3, 3. doi: 10.3389/fneng.2010.00003
- Chen, B., Zhao, S., Seth, S., and Principe, J. C. (2012). “Online efficient learning with quantized KLMS and L1 regularization,” in *The 2012 International Joint Conference on Neural Networks (IJCNN)* (Brisbane), 1–6. doi: 10.1109/IJCNN.2012.6252455
- Chen, C., Shin, D., Watanabe, H., Nakanishi, Y., Kambara, H., Yoshimura, N., et al. (2013). Prediction of hand trajectory from electrocorticography signals in primary motor cortex. *PLoS ONE* 8, e83534. doi: 10.1371/journal.pone.0083534
- Chen, C. K., and Fang, W. C. (2017). “A reliable brain-computer interface based on SSVEP using online recursive independent component analysis,” in *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* (Jeju Island), 2798–2801. doi: 10.1109/EMBC.2017.8037438
- Choi, H., Lee, J., Park, J., Lee, S., Ahn, K., Kim, I. Y., et al. (2018). Improved prediction of bimanual movements by a two-staged (effector-then-trajectory) decoder with epidural ECoG in nonhuman primates. *J. Neural Eng.* 15, 016011. doi: 10.1088/1741-2552/aa8a83
- Cichocki, A., Mandic, D., Lathauwer, L. D., Zhou, G., Zhao, Q., Caiafa, C., et al. (2015). Tensor decompositions for signal processing applications: from two-way to multiway component analysis. *IEEE Signal Process. Mag.* 32, 145–163. doi: 10.1109/MSP.2013.2297439
- Cincotti, F., Mattia, D., Aloise, F., Bufalari, S., Astolfi, L., De Vico Fallani, F., et al. (2008). High-resolution EEG techniques for brain-computer interface applications. *J. Neurosci. Methods* 167, 31–42. doi: 10.1016/j.jneumeth.2007.06.031
- Clerc, M., Dauce, E., and Mattout, J. (2016). “Adaptive methods in machine learning,” in *Brain-Computer Interfaces*, eds M. Clera, I. S. Antipolis, L. Bougrain, and F. Lotte (Hoboken, NJ: Wiley-Blackwell), 207–232. doi: 10.1002/9781119144977.ch10
- Dagher, I. (2010). “Incremental PCA-LDA algorithm,” in *2010 IEEE International Conference on Computational Intelligence for Measurement Systems and Applications*, 97–101. doi: 10.1109/CIMSA.2010.5611752
- Dayal, B. S., and MacGregor, J. F. (1997). Recursive exponentially weighted PLS and its applications to adaptive control and prediction. *J. Process. Control* 7, 169–79. doi: 10.1016/S0959-1524(97)80001-7
- Eliseyev, A., and AksenoVA, T. (2014). Stable and artifact-resistant decoding of 3D hand trajectories from ECoG signals using the generalized additive model. *J. Neural Eng.* 11, 066005. doi: 10.1088/1741-2560/11/6/066005
- Eliseyev, A., and AksenoVA, T. (2016). Penalized multi-way partial least squares for smooth trajectory decoding from electrocorticographic (ECoG) recording. *PLoS ONE* 11, e0154878. doi: 10.1371/journal.pone.0154878
- Eliseyev, A., Aubouiroux, V., Costecalde, T., Langar, L., Charvet, G., Mestais, C., et al. (2017). Recursive exponentially weighted n-way partial least squares regression with recursive-validation of hyper-parameters in brain-computer interface applications. *Sci. Rep.* 7, 16281. doi: 10.1038/s41598-017-16579-9
- Eliseyev, A., Moro, C., Faber, J., Wyss, A., Torres, N., Mestais, C., et al. (2012). L1-penalized N-way PLS for subset of electrodes selection in BCI experiments. *J. Neural Eng.* 9, 045010. doi: 10.1088/1741-2560/9/4/045010
- Faller, J., Vidaurre, C., Solis-Escalante, T., Neuper, C., and Scherer, R. (2012). Autocalibration and recurrent adaptation: towards a plug and play online ERD-BCI. *IEEE Trans. Neural Syst. Rehabil. Eng.* 20, 313–9. doi: 10.1109/TNSRE.2012.2189584
- Flamary, R., and Rakotomamonjy, A. (2012). Decoding finger movements from ecog signals using switching linear models. *Front. Neurosci.* 6, 29. doi: 10.3389/fnins.2012.00029
- Giordani, P., and Rocci, R. (2013). Constrained candecomp/parafac via the lasso. *Psychometrika* 78, 669–84. doi: 10.1007/s11336-013-9321-9
- Hastie, T., Tibshirani, R., Wainwright, M., Tibshirani, R., and Wainwright, M. (2015). *Statistical Learning with Sparsity: The Lasso and Generalizations*. Boca Raton, FL: Chapman and Hall/CRC. doi: 10.1201/b18401
- Haufe, S., Dähne, S., and Nikulin, V. V. (2014). Dimensionality reduction for the analysis of brain oscillations. *Neuroimage* 101, 583–97. doi: 10.1016/j.neuroimage.2014.06.073
- Helland, K., Berntsen, H. E., Borgen, O. S., and Martens, H. (1992). Recursive algorithm for partial least squares regression. *Chemom. Intell. Lab. Syst.* 14, 129–37. doi: 10.1016/0169-7439(92)80098-O
- Hervás, D., Prats-Montalbán, J. M., García-Cañaveras, J. C., Lahoz, A., and Ferrer, A. (2019). Sparse N-way partial least squares by L1-penalization. *Chemom. Intell. Lab. Syst.* 185, 85–91. doi: 10.1016/j.chemolab.2019.01.004
- Hsu, S. H., Mullen, T. R., Jung, T. P., and Cauwenberghs, G. (2016). Real-time adaptive EEG source separation using online recursive independent component analysis. *IEEE Trans. Neural Syst. Rehabil. Eng.* 24, 309–19. doi: 10.1109/TNSRE.2015.2508759
- Huang, D., Lin, P., Fei, D., Chen, X., and Bai, O. (2009). “EEG-based online two-dimensional cursor control,” in *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (Minneapolis), 4547–4550.
- Jafarifarmand, A., and Badamchizadeh, M. A. (2020). Real-time multiclass motor imagery brain-computer interface by modified common spatial patterns and adaptive neuro-fuzzy classifier. *Biomed. Signal Process. Control* 57, 101749. doi: 10.1016/j.bspc.2019.101749
- Jarosiewicz, B., Masse, N. Y., Bacher, D., Cash, S. S., Eskandar, E., Friebs, G., et al. (2013). Advantages of closed-loop calibration in intracortical brain-computer interfaces for people with tetraplegia. *J. Neural Eng.* 10, 046012. doi: 10.1088/1741-2560/10/4/046012
- Jiang, T., Jiang, T., Wang, T., Mei, S., Liu, Q., Li, Y., et al. (2017). Characterization and decoding the spatial patterns of hand extension/flexion using high-density ECoG. *IEEE Trans. Neural Syst. Rehabil. Eng.* 25, 370–9. doi: 10.1109/TNSRE.2016.2647255
- Khair, U. M., and Dhanalakshmi, R. (2019). Stability of feature selection algorithm: a review. *J. King Saud Univ. Comput. Inf. Sci.* 34, 1060–1073. doi: 10.1016/j.jksuci.2019.06.012
- Khan, J., Bhatti, M. H., Khan, U. G., and Iqbal, R. (2019). Multiclass EEG motor-imagery classification with sub-band common spatial patterns. *EURASIP J. Wirel. Commun. Netw.* 2019, 174. doi: 10.1186/s13638-019-1497-y
- Kim, H. J., Ollila, E., Koivunen, V., and Croux, C. (2013). “Robust and sparse estimation of tensor decompositions,” in *2013 IEEE Global Conference on Signal and Information Processing* (Austin, TX), 965–968. doi: 10.1109/GlobalSIP.2013.6737053
- Kim, H. J., Ollila, E., Koivunen, V., and Poor, H. V. (2014). “Robust iteratively reweighted Lasso for sparse tensor factorizations,” in *2014 IEEE Workshop on Statistical Signal Processing (SSP)* (Gold Coast), 420–423. doi: 10.1109/SSP.2014.6884665
- Kim, S., White, A., Scalzo, F., and Collier, D. (2018). Elastic net ensemble classifier for event-related potential based automatic spelling. *Biomed. Signal Process. Control* 46, 166–73. doi: 10.1016/j.bspc.2018.06.005
- Kim, S. P., Sanchez, J. C., Rao, Y. N., Erdogmus, D., Carmena, J. M., Lebedev, M. A., et al. (2006). A comparison of optimal MIMO linear and nonlinear models for brain-machine interfaces. *J. Neural Eng.* 3, 145–61. doi: 10.1088/1741-2560/3/2/009
- Kolda, T., and Bader, B. (2009). Tensor decompositions and applications. *SIAM Rev.* 51, 455–500. doi: 10.1137/07070111X
- Long, J., Gu, Z., Li, Y., Yu, T., Li, F., Fu, M., et al. (2011). Semi-supervised joint spatio-temporal feature selection for P300-based BCI speller. *Cogn. Neurodyn.* 5, 387. doi: 10.1007/s11571-011-9167-8
- López-Larraz, E., Montesano, L., Gil-Agudo, Á., and Minguez, J. (2014). Continuous decoding of movement intention of upper limb self-initiated analytic movements from pre-movement EEG correlates. *J. Neuroeng. Rehabil.* 11, 153. doi: 10.1186/1743-0003-11-153
- Lotte, F., Bougrain, L., Cichocki, A., Clerc, M., Congedo, M., Rakotomamonjy, A., et al. (2018). A review of classification algorithms for EEG-based brain-computer interfaces: a 10 year update. *J. Neural Eng.* 15, 031005. doi: 10.1088/1741-2552/aab2f2
- Lotte, F., and Guan, C. (2011). Regularizing Common spatial patterns to improve BCI designs: unified theory and new algorithms. *IEEE Trans. Biomed. Eng.* 58, 355–62. doi: 10.1109/TBME.2010.2082539
- Lutay, V. N., and Khusainov, N. S. (2021). The selective regularization of a linear regression model. *J. Phys. Conf. Ser.* 2099, 012024. doi: 10.1088/1742-6596/2099/1/012024
- Mak, J. N., and Wolpaw, J. R. (2009). Clinical applications of brain-computer interfaces: current state and future prospects. *IEEE Rev. Biomed. Eng.* 2, 187–99. doi: 10.1109/RBME.2009.2035356
- Maleki, M., Manshour, N., and Kayıkçıoğlu, T. (2018). “Fast and accurate classifier-based brain-computer interface system using single channel EEG data,” in *2018 26th Signal Processing and Communications Applications Conference (SIU)*, 1–4. doi: 10.1109/SIU.2018.8404376
- Marathe, A. R., and Taylor, D. M. (2013). Decoding continuous limb movements from high-density epidural electrode arrays using custom spatial filters. *J. Neural Eng.* 10, 036015. doi: 10.1088/1741-2560/10/3/036015
- Marathe, A. R., and Taylor, D. M. (2015). The impact of command signal power distribution, processing delays, and speed scaling on neurally-controlled devices. *J. Neural Eng.* 12, 046031. doi: 10.1088/1741-2560/12/4/046031

- Martínez-Montes, E., Sánchez-Bornot, J. M., and Valdés-Sosa, P. A. (2008). Penalized parafac analysis of spontaneous eeg recordings. *Stat. Sin.* 18, 1449–64.
- Mend, M., and Kullmann, W. H. (2012). Human computer interface with online brute force feature selection. *Biomed. Eng. Biomed. Tech.* 57, 659–62. doi: 10.1515/bmt-2012-4082
- Mishra, P. K., Jagadish, B., Kiran, M. P. R. S., Rajalakshmi, P., and Reddy, D. S. (2018). “A novel classification for EEG based four class motor imagery using kullback-leibler regularized riemannian manifold,” *2018 IEEE 20th International Conference on e-Health Networking, Applications and Services (Healthcom)* (Ostrava), 1–5. doi: 10.1109/HealthCom.2018.8531086
- Mobaien, A., and Boostani, R. (2016). “ACSP: adaptive CSP filter for BCI applications,” in *2016 24th Iranian Conference on Electrical Engineering (ICEE)*, 466–471. doi: 10.1109/IranianCEE.2016.7585567
- Moly, A., Costecalde, T., Martel, F., Martin, M., Larzabal, C., Karakas, S., et al. (2019). “A novel classification for EEG based four class motor imagery using kullback-leibler regularized riemannian manifold,” *2018 IEEE 20th International Conference on e-Health Networking, Applications and Services (Healthcom)* (Ostrava), 1–5. doi: 10.1109/HealthCom.2018.8531086
- Moro, R., Berger, P., and Bielikova, M. (2017). “Towards adaptive brain-computer interfaces: improving accuracy of detection of event-related potentials,” *2017 12th International Workshop on Semantic and Social Media Adaptation and Personalization (SMAP)* (Bratislava, SK), 34–39. doi: 10.1109/SMAP.2017.8022664
- Motrenko, A., and Strijov, V. (2018). Multiway feature selection for ECoG-based brain-computer interface. *Expert Syst. Appl.* 114, 402–13. doi: 10.1016/j.eswa.2018.06.054
- Nagel, S., and Spüler, M. (2019). Asynchronous non-invasive high-speed BCI speller with robust non-control state detection. *Sci. Rep.* 9, 1–9. doi: 10.1038/s41598-019-44645-x
- Nakanishi, Y., Yanagisawa, T., Shin, D., Kambara, H., Yoshimura, N., Tanaka, M., et al. (2017). Mapping ECoG channel contributions to trajectory and muscle activity prediction in human sensorimotor cortex. *Sci. Rep.* 7, 1–13. doi: 10.1038/srep45486
- Nicolas-Alonso, L. F., and Gomez-Gil, J. (2012). Brain computer interfaces, a review. *Sensors* 12, 1211–1279. doi: 10.3390/s120201211
- Oliver, G., Sunehag, P., and Gedeon, T. (2013). “Online feature selection for brain computer interfaces,” in *2013 IEEE Symposium on Computational Intelligence, Cognitive Algorithms, Mind, and Brain (CCMB)* (Singapore), 122–129. doi: 10.1109/CCMB.2013.6609175
- Palmer, J. A., and Hirata, M. (2018). Independent component analysis (ICA) features for electro-corticographic (ECoG) brain-machine interfaces (BMIs). doi: 10.1038/s41598-019-44645-x
- Pereira Da Silva, A., Comon, P., and De Almeida, A. L. F. (2015). *Rank-1 Tensor Approximation Methods and Application to Deflation*. Saint-Martin-d'Hères: GIPSA-lab.
- Peterson, V., Wyser, D., Lamberg, O., Spies, R., and Gassert, R. (2019). A penalized time-frequency band feature selection and classification procedure for improved motor intention decoding in multichannel EEG. *J. Neural Eng.* 16, 016019. doi: 10.1088/1741-2552/aaf046
- Qin, S. J. (1998). Recursive PLS algorithms for adaptive data modeling. *Comput. Chem. Eng.* 22, 503–14. doi: 10.1016/S0098-1354(97)00262-7
- Remeseiro, B., and Bolon-Canedo, V. (2019). A review of feature selection methods in medical applications. *Comput. Biol. Med.* 112, 103375. doi: 10.1016/j.combiomed.2019.103375
- Roijndijk, L., Gielen, S., and Farquhar, J. (2016). Classifying regularized sensor covariance matrices: an alternative to CSP. *IEEE Trans. Neural Syst. Rehabil. Eng.* 24, 893–900. doi: 10.1109/TNSRE.2015.2477687
- Rouanne, V., Costecalde, T., Benabid, A. L., and Aksenova, T. (2022). Unsupervised adaptation of an ECoG based brain-computer interface using neural correlates of task performance. *Sci. Rep.* 12, 1–11. doi: 10.1038/s41598-022-25049-w
- Rouanne, V., Sliwowski, M., Costecalde, T., Benabid, A. L., and Aksenova, T. (2021). “Detection of error correlates in the motor cortex in a long term clinical trial of ECoG based brain computer interface,” in *BIOSIGNALS*, 26–34. doi: 10.5220/0010227800260034
- Sannelli, C., Vidaurre, C., Müller, K.-R., and Blankertz, B. (2016). Ensembles of adaptive spatial filters increase BCI performance: an online evaluation. *J. Neural Eng.* 13, 046003. doi: 10.1088/1741-2560/13/4/046003
- Sauter-Starace, F., Ratel, D., Cretallaz, C., Foerster, M., Lambert, A., Gaude, C., et al. (2019). Long-term sheep implantation of WIMAGINE®, a Wireless 64-channel electrocorticogram recorder. *Front. Neurosci.* 13, 847. doi: 10.3389/fnins.2019.00847
- Schaeffer, M.-C., and Aksenova, T. (2016). Switching Markov decoders for asynchronous trajectory reconstruction from ECoG signals in monkeys for BCI applications. *J. Physiol.* 110, 348–60. doi: 10.1016/j.jphysparis.2017.03.002
- Schaeffer, M. C. (2017). *Traitement du Signal ECoG Pour Interface Cerveau Machine à Grand Nombre de Degrés de Liberté Pour Application Clinique* [Doctoral dissertation]. Université Grenoble Alpes (ComUE), Grenoble.
- Schlögl, A., Vidaurre, C., and Müller, K.-R. (2010). “Adaptive methods in bci research - an introductory tutorial,” in *Brain-Computer Interfaces: Revolutionizing Human-Computer Interaction. The Frontiers Collection*, eds B. Graimann, G. Pfurtscheller, and B. Allison (Berlin, Heidelberg: Springer Berlin Heidelberg), 331–355. doi: 10.1007/978-3-642-02091-9_18
- Schwartz, A. B., Cui, X. T., Weber, D. J., and Moran, D. W. (2006). Brain-controlled interfaces: movement restoration with neural prosthetics. *Neuron* 52, 205–20. doi: 10.1016/j.neuron.2006.09.019
- Seifzadeh, S., Rezaei, M., Faez, K., and Amiri, M. (2017). Fast and efficient four-class motor imagery electroencephalography signal analysis using common spatial pattern-ridge regression algorithm for the purpose of brain-computer interface. *J. Med. Signals Sens.* 7, 80–5. doi: 10.4103/2228-7477.205593
- Shanechi, M. M., Orsborn, A. L., Moorman, H. G., Gowda, S., Dangi, S., Carmena, J. M., et al. (2017). Rapid control and feedback rates enhance neuroprosthetic control. *Nat. Commun.* 8, 13825. doi: 10.1038/ncomms13825
- Sharghian, V., Rezaei, T. Y., Farzamina, A., and Tinati, M. A. (2019). “Online dictionary learning for sparse representation-based classification of motor imagery EEG,” in *2019 27th Iranian Conference on Electrical Engineering (ICEE)*, 1793–1797. doi: 10.1109/IranianCEE.2019.8786703
- Sheikhhattar, A., Fritz, J. B., Shamma, S. A., and Babadi, B. (2015). “Adaptive sparse logistic regression with application to neuronal plasticity analysis,” in *2015 49th Asilomar Conference on Signals, Systems and Computers*, 1551–1555. doi: 10.1109/ACSSC.2015.7421406
- Shimoda, K., Nagasaka, Y., Chao, Z. C., and Fujii, N. (2012). Decoding continuous three-dimensional hand trajectories from epidural electrocorticographic signals in Japanese macaques. *J. Neural Eng.* 9, 036015. doi: 10.1088/1741-2560/9/3/036015
- Sliwowski, M., Martin, M., Souloumiac, A., Blanchart, P., and Aksenova, T. (2022). Decoding ECoG signal into 3D hand translation using deep learning. *J. Neural Engin.* 19, 026023. doi: 10.1088/1741-2552/ac5d69
- Song, X., and Yoon, S.-C. (2015). Improving brain-computer interface classification using adaptive common spatial patterns. *Comput. Biol. Med.* 61, 150–60. doi: 10.1016/j.combiomed.2015.03.023
- Spüler, M., Rosenstiel, W., and Bogdan, M. (2012). “Adaptive SVM-based classification increases performance of a meg-based brain-computer interface (BCI),” in *Artificial Neural Networks and Machine Learning – ICANN. 2012. Lecture Notes in Computer Science*, eda A. E. P. Villa, W. Duch, P. Erdi, F. Masulli, and G. Palm (Berlin, Heidelberg: Springer), 669–676. doi: 10.1007/978-3-642-33269-2_84
- Sreeja, S. R., Himanshu, Samanta, D., and Sarma, M. (2019). “Weighted sparse representation for classification of motor imagery EEG signals,” in *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* (Berlin), 6180–6183. doi: 10.1109/EMBC.2019.8857496
- Sreenath, R., and Ramana, R. (2017). Classification of denoising techniques for EEG signals: a review. *Int. J. Pure Appl. Math.* 117, 967–72.
- Toda, A., Imamizu, H., Kawato, M., and Sato, M. (2011). Reconstruction of two-dimensional movement trajectories from selected magnetoencephalography cortical currents by combined sparse Bayesian methods. *NeuroImage* 54, 892–905. doi: 10.1016/j.neuroimage.2010.09.057
- Trejo, L. J., Rosipal, R., and Matthews, B. (2006). Brain-computer interfaces for 1-D and 2-D cursor control: designs using volitional control of the EEG spectrum or steady-state visual evoked potentials. *IEEE Trans. Neural Syst. Rehabil. Eng.* 14, 225–9. doi: 10.1109/TNSRE.2006.875578
- University Hospital, Grenoble. (2015). *Brain Computer Interface: Neuroprosthetic Control of a Motorized Exoskeleton, NCT02550522, ICTRP Clinical Trial*. Available online at: <https://clinicaltrials.gov/ct2/show/NCT02550522?cond=NCT02550522&draw=2&rank=1>
- Uschmajew, A. (2015). A new convergence proof for the higher-order power method and generalizations. *ArXiv*. doi: 10.48550/arXiv.1407.4586
- van Gerven, M., Hesse, C., Jensen, O., and Heskes, T. (2009). Interpreting single trial data using groupwise regularization. *NeuroImage* 46, 665–76. doi: 10.1016/j.neuroimage.2009.02.041
- Woehrle, H., Krell, M. M., Straube, S., Kim, S. K., Kirchner, E. A., Kirchner, F., et al. (2015). An adaptive spatial filter for user-independent single trial detection of event-related potentials. *IEEE Trans. Biomed. Eng.* 62, 1696–705. doi: 10.1109/TBME.2015.2402252
- Wu, Q., Zhang, Y., Liu, J., Sun, J., Cichocki, A., Gao, F., et al. (2019). Regularized group sparse discriminant analysis for p300-based brain-computer interface. *Int. J. Neural Syst.* 29, 1950002. doi: 10.1142/S0129065719500023
- Yang, C., Qiao, J., Ahmad, Z., Nie, K., and Wang, L. (2019). Online sequential echo state network with sparse RLS algorithm for time series prediction. *Neural Netw.* 118, 32–42. doi: 10.1016/j.neunet.2019.05.006

Zhang, Y., Zhou, G., Jin, J., Wang, M., Wang, X., Cichocki, A., et al. (2013). L1-regularized multiway canonical correlation analysis for SSVEP-based BCI. *IEEE Trans. Neural Syst. Rehabil. Eng.* 21, 887–96. doi: 10.1109/TNSRE.2013.2279680

Zhang, Y., Zhou, G., Jin, J., Wang, X., and Cichocki, A. (2015). Optimizing spatial patterns with sparse filter bands for motor-imagery based brain-computer interface. *J. Neurosci. Methods* 255, 85–91. doi: 10.1016/j.jneumeth.2015.08.004

Zhao, Q., Caiafa, C. F., Mandic, D. P., Chao, Z. C., Nagasaka, Y., Fujii, N., et al. (2013). Higher order partial least squares (HOPLS): a generalized multilinear regression method. *IEEE Trans. Pattern Anal. Mach. Intell.* 35, 1660–73. doi: 10.1109/TPAMI.2012.254

Zhao, Q., Liqing, Z., Cichocki, A., and Jie, L. (2008). “Incremental common spatial pattern algorithm for BCI” in *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)* (Hong Kong), p. 2656–2659.