frontiers | Frontiers in Signal Processing

METHODS
published: 06 July 2022
doi: 10.3389/frsip.2022.867388

Adaptive Discrete Motion Control for Mobile Relay Networks

Spilios Evmorfos[1†], *Dionysios Kalogerias*[2†] *and Athina Petropulu*[1*†]

[1]*Electrical and Computer Engineering, Rutgers, The State University of New Jersey, New Brunswick, NJ, United States,* [2]*Electrical Engineering, Yale University, New Haven, CT, United States*

OPEN ACCESS

Edited by:
Monica Bugallo,
Stony Brook University, United States

Reviewed by:
Francesco Palmieri,
University of Campania Luigi Vanvitelli, Italy
Stefania Colonnese,
Sapienza University of Rome, Italy

***Correspondence:**
Athina Petropulu
athinap@rutgers.edu

[†]*These authors have contributed equally to this work*

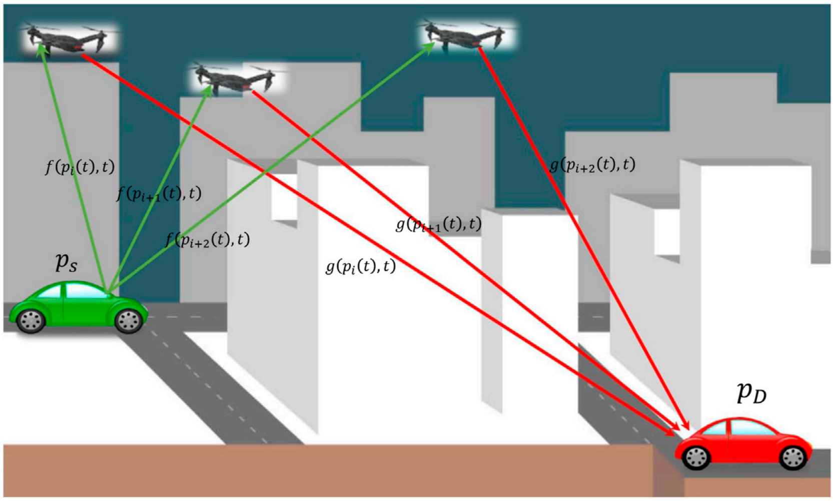
Specialty section:
This article was submitted to Signal Processing for Communications, a section of the journal Frontiers in Signal Processing

Received: *01 February 2022*
Accepted: *01 June 2022*
Published: *06 July 2022*

Citation:
Evmorfos S, Kalogerias D and Petropulu A (2022) Adaptive Discrete Motion Control for Mobile Relay Networks. Front. Sig. Proc. 2:867388. doi: 10.3389/frsip.2022.867388

We consider the problem of joint beamforming and discrete motion control for mobile relaying networks in dynamic channel environments. We assume a single source-destination communication pair. We adopt a general time slotted approach where, during each slot, every relay implements optimal beamforming and estimates its optimal position for the subsequent slot. We assume that the relays move in a 2D compact square region that has been discretized into a fine grid. The goal is to derive discrete motion policies for the relays, in an adaptive fashion, so that they accommodate the dynamic changes of the channel and, therefore, maximize the Signal-to-Interference + Noise Ratio (SINR) at the destination. We present two different approaches for constructing the motion policies. The first approach assumes that the channel evolves as a Gaussian process and exhibits correlation with respect to both time and space. A stochastic programming method is proposed for estimating the relay positions (and the beamforming weights) based on causal information. The stochastic program is equivalent to a set of simple subproblems and the exact evaluation of the objective of each subproblem is impossible. To tackle this we propose a surrogate of the original subproblem that pertains to the Sample Average Approximation method. We denote this approach as model-based because it adopts the assumption that the underlying correlation structure of the channels is completely known. The second method is denoted as model-free, because it adopts no assumption for the channel statistics. For the scope of this approach, we set the problem of discrete relay motion control in a dynamic programming framework. Finally we employ deep Q learning to derive the motion policies. We provide implementation details that are crucial for achieving good performance in terms of the collective SINR at the destination.

Keywords: relay networks, discrete motion control, stochastic programming, dynamic programming, deep reinforcement learning



GRAPHICAL ABSTRACT |

1 INTRODUCTION

In distributed relay beamforming networks, spatially distributed relays synergistically support the communication between a source and a destination (Havary-Nassab et al., 2008a; Li et al., 2011; Liu and Petropulu, 2011). The concepts of distributed beamforming hold the promise of extending the communication range and of minimizing the transmit power that is being wasted by being scattered to unwanted directions (Barriac et al., 2004).

Intelligent node mobility has been studied as a means of improving the Quality-of-Service (QoS) in communications. In (Chatzipanagiotis et al., 2014), the interplay of relay motion control and optimal transmit beamforming is considered with the goal of minimizing the relay transmit power, subject to a QoS-related constraint. In (Kalogerias et al., 2013), optimal relay positioning in the presence of an eavesdropper is considered, aiming to maximize the secrecy rate. In the context of communication-aware robotics, motion has been controlled with the goal of maintaining in-network connectivity (Yan and Mostofi, 2012; Yan and Mostofi, 2013; Muralidharan and Mostofi, 2017).

In this work, we examine the problem of optimizing the sequence of relay positions (relay trajectory) and the beamforming weights so that some SINR-based metric is maximized at the destination. The assumption that we adopt is that the channel evolves as a stochastic process that exhibits spatiotemporal correlations. Intrinsically, optimal relay positioning requires the knowledge of the Channel State Information (CSI) in all candidate positions at a future time instance. This is almost impossible to achieve since the channel varies with respect to time and space. Nonetheless, since the channel exhibits spatiotemporal correlations (induced by the shadowing propagation effect (Goldsmith, 2005; MacCartney et al., 2013) that is prominent in urban environments), it can be, explicitly or implicitly, predicted. We follow two different directions, when it comes to the discrete relay motion control.

The first direction (Kalogerias and Petropulu, 2018; Kalogerias and Petropulu, 2016) (we call it model-based) pertains to the formulation of a stochastic program that computes the beamforming weights and the subsequent relay positions, so that some SINR-based metric at the destination is maximized, subject to a total relay power budget, assuming the availability of causal CSI information. This 2-stage problem is equivalent to a set of 2-stage subproblems that can be solved in distributed fashion, one by each relay. The objective of each subproblem is impossible to be analytically evaluated, so an efficient approximation is proposed. This approximation acts as a surrogate to the initial objective. The surrogate relies on the Sample Average Approximation (SAA) (Shapiro et al., 2009). The term "model-based" is not to be confused with model-based reinforcement learning. We just use it because this method (or direction rather) assumes complete knowledge of the underlying correlation structure of the channels, so it is helpful formalism to distinguish this method from the second approach that makes no particular assumption for the channel statistics.

The second direction (Evmorfos et al., 2021a; Evmorfos et al., 2021b; Evmorfos et al., 2022) tackles the problem of discrete relay motion control from a dynamic programming viewpoint. We formulate the Markov Decision Process (MDP), that is induced by the problem of controlling the motion. Finally, we employ deep Q learning (Mnih et al., 2015) to find relay motion policies that maximize the sum of SINRs at the destination over time. We propose a pipeline for adapting deep Q learning for the problem at hand. We experimentally show that Multilayer Perceptron Neural Networks (MLPs) cannot capture high frequency components in natural signals (in low-dimensional domains). This phenomenon, referred to as *"Spectral Bias"* (Jacot et al., 2018) has been observed in several contexts, and also arises as an issue in the adaptation of deep Q learning for the relay motion control. We present an approach to tackle spectral bias, by parameterizing the Q function with a Sinusoidal Representation Network (SIREN) (Sitzmann et al., 2020).

Our intentions for this work lie in two directions. First, we attempt to compare two methods for relay motion control in urban communication environments. The two methods constitute two different viewpoints in terms of tackling the problem. The first method assumes complete knowledge on the underlying statistics of the channels (model-based) Kalogerias and Petropulu, (2018). The second method is completely model-free in the sense that it drops all assumptions for knowledge of the channel statistics and employs deep reinforcement learning to control the relay motion Evmorfos et al. (2022). In addition to the head-to-head comparison, we propose a slight variation of the model-free method that deviates from the one in Evmorfos et al. (2022) by augmenting the state with the addition of the timestep as an extra feature. This variation is more robust than the previous one, especially when the shadowing component of the urban environment is particularly strong.

Notation: We denote the matrices and vectors by bold uppercase and bold lowercase letters, respectively. The operators $(\cdot)^{\boldsymbol{T}}$ and $(\cdot)^{\boldsymbol{H}}$ denote transposition and conjugate transposition respectively. Caligraphic letters will be used to denote sets and formal script letters will be used to denote σ-algebras. The ℓ_p-norm of $\boldsymbol{x} \in \mathbb{R}^n$ is $\|\boldsymbol{x}\|_p \triangleq (\sum_{i=1}^n |x(i)|^p)^{1/p}$, for all $\mathbb{N} \ni p \geq 1$. For $\mathbb{N} \ni N \geq 1$, $\mathbb{S}^N$, $\mathbb{S}^N_{+(+)}$ will denote the sets of symmetric and symmetric positive (semidefinite) matrices, respectively. The finite N-dimensional identity operator will be denoted as $\mathbf{I}_N$. Additionally, we define $\mathfrak{J} \triangleq \sqrt{-1}$, $\mathbb{N}^+ \triangleq \{1, 2, \ldots\}$, $\mathbb{N}^+_n \triangleq \{1, 2, \ldots, n\}$, $\mathbb{N}_n \triangleq \{0\} \cup \mathbb{N}^+_n$ and $\mathbb{N}^m_n \triangleq \mathbb{N}^+_n \backslash \mathbb{N}^+_{m-1}$, for positive naturals $n > m$.

2 PROBLEM FORMULATION

2.1 System Model

Consider a scenario where source S, located at position $\mathbf{p}_{\mathsf{S}} \in \mathbb{R}^2$, wishes to communicate with user D, located at $\mathbf{p}_{\mathsf{D}} \in \mathbb{R}^2$ but does not have enough power to do so, or due to the topography, cannot communicate in a line-of-sight (LoS) fashion. Therefore, R single-antenna, trusted mobile relays are enlisted to support the communication. The relays are deployed over a two-dimensional space, which is partitioned into $M \times M$ imaginary grid cells. Time evolves in a time-slotted fashion, where T is the slot duration, and t denotes the current time slot. In every time slot, a grid cell can be occupied by at most one relay.

Source S transmits symbol $s(t) \in \mathbb{C}$, where $\mathbb{E}[|s(t)|^2] = 1$, using power $\sqrt{P_S} > 0$. Let us drop for notational simplicity the relay position dependence on t. The signal received by relay R_r, located at $\mathbf{p}_r(t)$, $r = 1, \ldots, R$, equals

$$x_r(t) = \sqrt{P_S} f_r(\mathbf{p}_r, t) s(t) + n_r(t),$$

where f_r denotes the flat fading channel from S to relay R_r, and $n_r(t)$ denotes reception noise at relay R_r, with $\mathbb{E}[|n_r(t)|^2] = \sigma^2$, $r = 1, \ldots, R$.

Each relay operates in an Amplify-and-Forward (AF) fashion, i.e., it transmits received signal, $x_r(t)$, multiplied by weight $w_r(t) \in \mathbb{C}$. Due to the relays' simultaneous transmissions, the destination D receives

$$y(t) = \sum_{r=1}^{R} g_r(\mathbf{p}_{\mathsf{D}}, t) w_r(t) x_r(t) + n_{\mathsf{D}}(t),$$

where g_r denotes the flat fading channel from relay R_r to destination D, and $n_{\mathsf{D}}(t)$ denotes reception noise at D. We assume here that $\mathbb{E}[|n_{\mathsf{D}}(t)|^2] = \sigma_D^2$ $y(t)$ can be rewritten as

$$\begin{aligned} y(t) &= \underbrace{\sum_{r=1}^{R} g_r(\mathbf{p}_{\mathsf{D}}, t) w_r(t) \sqrt{P_S} f_r(\mathbf{p}_r, t) s(t)}_{\text{desired signal}} + \underbrace{\sum_{r=1}^{R} g_r(\mathbf{p}_{\mathsf{D}}, t) w_r(t) n_r(t) + n_{\mathsf{D}}(t)}_{\text{noise}} \\ &\triangleq y_{signal}(t) + y_{noise}(t), \end{aligned}$$

where $y_{signal}(t)$ is the received signal component and $n_{\mathsf{D}}(t)$ represents noise at the destination.

In the following, we will use the vector $\mathbf{p}(t) \triangleq [\mathbf{p}_1^{\boldsymbol{T}}(t)\, \mathbf{p}_2^{\boldsymbol{T}}(t) \ldots \mathbf{p}_R^{\boldsymbol{T}}(t)]^{\boldsymbol{T}}$, to collect the positions of all relays at time t.

2.2 Channel Model

The channel evolves in time and space and can be described in statistical terms. In particular, during time slot t, the channel between the source and a relay positioned at $\mathbf{p}_r \in \mathbb{R}^2$ can be modeled as the product of four components (Heath, 2017), i.e.,

$$f_r(\mathbf{p}_r, t) \triangleq f_r^{PL}(\mathbf{p}_r) f_r^{SH}(\mathbf{p}_r, t) f_r^{MF}(\mathbf{p}_r, t) e^{j2\pi\phi(t)}, \qquad (1)$$

where $f_r^{PL}(\mathbf{p}_r) \triangleq \|\mathbf{p}_r - \mathbf{p}_{\mathsf{S}}\|_2^{-\ell/2}$ is the path-loss component with path-loss exponent ℓ; $f_r^{SH}(\mathbf{p}_r, t)$ the shadow fading component; $f_r^{MF}(\mathbf{p}_r, t)$ the multi-path fading component; and $e^{j2\pi\phi(t)}$, with ϕ uniformly distributed in [0, 1], a phase term. A similar model holds for the relay-destination channel g_r ($\mathbf{p}_r$, t).

The logarithm of the squared channel magnitude of **Eq. 1** converts the multiplicative channel model into an additive one, i.e.,

$$\begin{aligned} F_r(\mathbf{p}_r, t) &\triangleq 10\log_{10}(|f_r(\mathbf{p}_r, t)|^2) \\ &\triangleq \alpha_r^f(\mathbf{p}_r) + \beta_r^f(\mathbf{p}_r, t) + \xi_r^f(\mathbf{p}_r, t), \end{aligned}$$

with

$$\begin{aligned} \alpha_r^f(\mathbf{p}_r) &\triangleq -\ell\, 10\log_{10}(\|\mathbf{p}_r - \mathbf{p}_{\mathsf{S}}\|_2), \\ \beta_r^f(\mathbf{p}_r, t) &\triangleq 10\log_{10}\left(|f_r^{SH}(\mathbf{p}_r, t)|^2\right) \sim \mathcal{N}(0, \eta^2), \quad \text{and} \\ \xi_r^f(\mathbf{p}_r, t) &\triangleq 10\log_{10}\left(|f_r^{MF}(\mathbf{p}_r, t)|^2\right) \sim \mathcal{N}\left(\rho, \sigma_\xi^2\right), \end{aligned}$$

where η^2 is the shadowing power, and ρ, σ_ξ^2 are the mean and variance of multipath fading component, respectively.

The multipath fading component, $\xi_r^f(\mathbf{p}_r, t)$, varies fast in time and space, and is typically modeled as is i. i.d. between different positions and times. On the other hand, the shadowing component, $\beta_r^f(\mathbf{p}_r, t)$, induced by relatively large and slowly moving objects in the path of the signal, exhibits correlation between any two positions $\mathbf{p}_i$ and $\mathbf{p}_j$, and between any two time slots t_a and t_b, as (Kalogerias and Petropulu, 2018)

$$\mathbb{E}\left[\beta_r^f(\mathbf{p}_i, t_a)\beta_r^f\left(\mathbf{p}_j, t_b\right)\right] = \tilde{\boldsymbol{\Sigma}}^f\left(\mathbf{p}_i, \mathbf{p}_j\right)e^{-\frac{|t_a - t_b|}{c_2}},$$

where

$$\tilde{\boldsymbol{\Sigma}}^f\left(\mathbf{p}_i, \mathbf{p}_j\right) \triangleq \eta^2 e^{-\|\mathbf{p}_i - \mathbf{p}_j\|_2 \big/ c_1} \in \mathbb{R}^{M^2 \times M^2},$$

with c_1 denoting the correlation distance, and c_2 the correlation time. Similar correlations hold for similarly $\beta_r^g(\mathbf{p}_i, t)$.

Further, $\beta_r^f(\mathbf{p}_i, t)$ and $\beta_r^g(\mathbf{p}_i, t)$ exhibit correlations as

$$\mathbb{E}\left[\beta_r^f(\mathbf{p}_i, t_a)\beta_r^g\left(\mathbf{p}_j, t_b\right)\right] = \tilde{\boldsymbol{\Sigma}}^{fg}\left(\mathbf{p}_i, \mathbf{p}_j\right)e^{-\frac{|t_a - t_b|}{c_2}},$$

where

$$\tilde{\boldsymbol{\Sigma}}^{fg}\left(\mathbf{p}_i, \mathbf{p}_j\right) = \tilde{\boldsymbol{\Sigma}}^f\left(\mathbf{p}_i, \mathbf{p}_j\right)e^{-\frac{\|\mathbf{p}_S - \mathbf{p}_D\|_2}{c_3}}$$

and c_3 denoting the correlation distance of the source-destination channel (Kalogerias and Petropulu, 2018).

2.3 Joint Scheduling of Communications and Controls

Let us assume the same carrier for all communication tasks, and employ a basic joint communication/decision making TDMA-like protocol. At each time slot $t \in \mathbb{N}^+_{N_T}$, the following actions are taken:

1. The source broadcasts a pilot signal to all relays, based on which the relays estimate their channels to the source.
2. The destination also broadcasts pilots, which the relays use to estimate their channels relative to the destination.
3. Then, based on the estimated channels, the relays beamform in AF mode. Here we assume perfect CSI estimation.
4. Based on the CSI that has been received up to that point, a decision is made on where the relays need to go to, and relay motion controllers are determined to steer the relays to those positions.

The above steps are repeated for N_T time slots. Let us assume that the relays pass their estimated CSI to the destination via a dedicated low-rate channel. This simplifies information decoding at the destination (Gao et al., 2008; Proakis and Salehi, 2008).

Concerning relay motion, we assume that the relays obey the differential equation (Kalogerias and Petropulu, 2018)

$$\dot{\mathbf{p}}(\tau) \equiv \mathbf{u}(\tau), \quad \forall \tau \in [0, T],$$

where $\mathbf{u} \triangleq [\mathbf{u}_1 \ldots \mathbf{u}_R]^T$, with $\mathbf{u}_i$: $[0, T]$ being the motion controller of relay $i \in \mathbb{N}^+_R$. Assuming the relays may move only after their controls have been determined and their movement must be completed before the start of the next time slot, we can write (Kalogerias and Petropulu, 2018)

$$\mathbf{p}(t) \equiv \mathbf{p}(t-1) + \int_{\Delta\tau_{t-1}} \mathbf{u}_{t-1}(\tau)\mathrm{d}\tau, \quad \forall t \in \mathbb{N}^2_{N_T},$$

with $\mathbf{p}(1) \equiv \mathbf{p}_{init}$, and where $\Delta\tau_t \subset \mathbb{R}$ and $\mathbf{u}_t$ denote the time interval that the relays are allowed to move in, and the respective relay controller, in each time slot $t \in \mathbb{N}^+_{N_T-1}$. It holds that $\mathbf{u}(\tau) \equiv \sum_{t \in \mathbb{N}^+_{N_T-1}} \mathbf{u}_t(\tau) 1_{\Delta\tau_t}(\tau)$, where τ belongs in the first $N_T - 1$ time slots. In each time slot t, the length of $\Delta\tau_t$, $|\Delta\tau_t|$, must be small enough, so that the shadowing correlation at adjacent time slots is strong enough. These correlations are controlled by parameter γ, which can be function of the slot width. Thus, relay velocity must be of the order of $(|\Delta\tau_t|)^{-1}$. For simplicity, here we assume that the relays are not resource constrained when they move and they are only limited by their transmission power.

To determine the relay motion controller $\mathbf{u}_{t-1}(\tau), \tau \in \Delta\tau_{t-1}$, given a goal position vector at time slot t, $\mathbf{p}^o(t)$, it suffices to decide on a path in $\mathcal{S}^R$, such that the points $\mathbf{p}^o(t)$ and $\mathbf{p}(t-1)$ are connected in at most time $|\Delta\tau_{t-1}|$. Assuming the simplest path, i.e., a straight line between $\mathbf{p}_i^o(t)$ and $\mathbf{p}_i(t-1)$, for all $i \in \mathbb{N}^+_R$, the relay controllers at time slot $t-1 \in \mathbb{N}^+_{N_T-1}$ is

$$\mathbf{u}^o_{t-1}(\tau) \triangleq \frac{1}{|\Delta\tau_{t-1}|}\left(\mathbf{p}^o(t) - \mathbf{p}(t-1)\right), \ \forall \tau \in \Delta\tau_{t-1}.$$

Based on the above, the motion control problem can be formulated in terms of specifying the relay positions at the next time slot, given the relay positions at the current time slot and the estimated CSI. We assume here for simplicity that there exists some path planning and collision avoidance mechanism, the derivation of which is out of the scope of this paper.

For simplicity and tractability, we are assuming that the channel is the same for every position *within* each grid cell, and for the duration of each time slot. In other words, we are essentially adopting a *time-space block fading model*, at least for motion control purposes. This is a valid approximation of reality as the grid cell size and the time slot duration become smaller, at the expense of more stringent resource constraints at the relays, and faster channel sensing capability. Under this setting, *communication and relay control can indeed happen simultaneously* within each time slot, with the understanding that at the start of the next time slot, each relay must have completed their motion (starting at the previous time slot–also see our discussion earlier in this section–). In this way, our approach is valid in a practical setting where communication needs to be continuous and uninterrupted.

Additionally, we are assuming that the relays move sufficiently slowly, such that the local spatial and temporal changes of the wireless channel due to relay motion itself are negligible, e.g., Doppler shift effects. Then, spatial and temporal variations in channel quality are only due to changes in the physical environment, which happen at a much slower rate than that of actual communication. Note that this is a standard requirement for achieving a high communication rate, whatsoever.

We see that there is a natural interplay between relay velocity and the relative rate of change of the communication channel Kalogerias and Petropulu (2018). The challenge here is to identify a fair tradeoff between a reasonable relay velocity, grid size and a time slot, which would enable simultaneously faithful channel prediction and feasible and effective motion control (adherring to potential relay motion constraints). The width of the communication time slot depends on the spatial characteristics

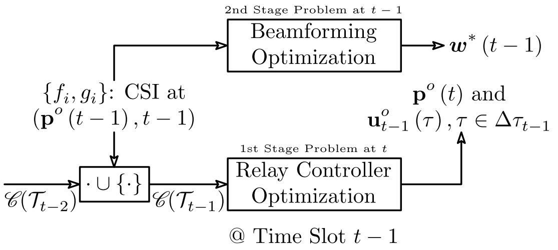

2nd Stage Problem at $t-1$
Beamforming Optimization
$\boldsymbol{w}^*(t-1)$
$\{f_i, g_i\}$: CSI at $(\mathbf{p}^o(t-1), t-1)$
$\mathbf{p}^o(t)$ and $\mathbf{u}^o_{t-1}(\tau), \tau \in \Delta\tau_{t-1}$
1st Stage Problem at t
$\cdot \cup \{\cdot\}$
$\mathscr{C}(\mathcal{T}_{t-2})$
$\mathscr{C}(\mathcal{T}_{t-1})$
Relay Controller Optimization
@ Time Slot $t-1$

FIGURE 1 | 2-Stage optimization of beamforming weights and relay motion controls. The variable $\boldsymbol{w}^*(t-1)$ denotes the optimal beamforming weights, selected at time slot $t-1$.

of the terrain, which varies with each application. This also determines the sampling rate employed for identifying the parameters of the adopted channel model. In theory, for a given relay velocity, the relays could move to any position up to which the channel remains correlated. However, as the per time slot rate of communications depends on the relay velocity (characterizing system throughput), the relays should move to much smaller distances within the slot.

In the following we use $\mathscr{C}(\mathcal{T}_t)$ to denote the set of channel gains observed by the relays, *along their trajectories* $\mathcal{T}_t \triangleq \{\mathbf{p}(1)\ldots\mathbf{p}(t)\}$, $t \in \mathbb{N}^+_{N_T}$. Then, $\mathcal{T}_t$ may be recursively updated as $\mathcal{T}_t \equiv \mathcal{T}_{t-1} \cup \{\mathbf{p}(t)\}$, for all $t \in \mathbb{N}^+_{N_T}$, with $\mathcal{T}_0 \triangleq \varnothing$. In a more precise sense, $\{\mathcal{C}(\mathcal{T}_t)\}_{t\in\mathbb{N}^+_{N_T}}$ will also denote the filtration generated by the CSI observed at the relays, *along* $\mathcal{T}_t$, interchangeably. In other words, $\mathscr{C}(\mathcal{T}_t)$ denotes the information (i.e., the σ-algebra) generated by the CSI observed up to and including time slot t and $\mathbf{p}(1)\ldots\mathbf{p}(t)$, for all $t \in \mathbb{N}^+_{N_T}$. By convention, we define $\mathscr{C}(\mathcal{T}_0) \equiv \mathscr{C}(\{\varnothing\})$ (i.e., as the trivial σ-algebra $\mathscr{C}(\mathcal{T}_0) \triangleq \{\varnothing, \Omega\}$), and we refer to time $t \equiv 0$, as a *dummy time slot.*

2.4 Spatially Controlled SINR Maximization at the Destination

Next, we present the first stage of the 2-stage generic formulation. The 2-stage approach optimizes network QoS by optimally selecting beamforming weights *and* relay positions, on a *per time slot* basis. In this subsection, we focus on the calculation of the beamforming weights. The calculation of the weights at each step remains the same both for the stochastic programming (model-based) method and the dynamic programming (model-free) method.

Optimization of Beamforming Weights: At time slot $t \in \mathbb{N}^+_{N_T}$, *given* CSI in $\mathscr{C}(\mathcal{T}_t)$, we formulate the problem (Havary-Nassab et al., 2008b; Zheng et al., 2009)

$$\begin{array}{ll} \underset{\boldsymbol{w}(t)\triangleq[w_1(t),\ldots,w_R(t)]^T}{\text{maximize}} & \dfrac{\mathbb{E}\{P_S(t)\,|\,\mathscr{C}(\mathcal{T}_t)\}}{\mathbb{E}\{P_{I+N}(t)\,|\,\mathscr{C}(\mathcal{T}_t)\}}, \\ \text{subject to} & \mathbb{E}\{P_R(t)\,|\,\mathscr{C}(\mathcal{T}_t)\} \le P_c \end{array} \tag{2}$$

where $P_R(t)$, $P_S(t)$ and $P_{I+N}(t)$ denote the random instantaneous power at the relays, the power of the signal component and the power of the interference plus noise at the destination, respectively, and where $P_c > 0$ denotes the total relay transmission power budget. Based on the mutual independence of source and destination CSI, (**Eq. 2**) can be expressed as (Havary-Nassab et al., 2008b)

$$\begin{array}{ll} \underset{\boldsymbol{w}(t)}{\text{maximize}} & \dfrac{\boldsymbol{w}^H(t)\mathbf{R}(\mathbf{p}(t),t)\boldsymbol{w}(t)}{\sigma_D^2 + \boldsymbol{w}^H(t)\mathbf{Q}(\mathbf{p}(t),t)\boldsymbol{w}(t)}, \\ \text{subject to} & \boldsymbol{w}^H(t)\mathbf{D}(\mathbf{p}(t),t)\boldsymbol{w}(t) \le P_c \end{array} \tag{3}$$

where, dropping the dependence on $(\mathbf{p}(t), t)$ or t for brevity,

$$\mathbf{D} \triangleq P_0 \text{diag}\Big(\big[|f_1|^2\, |f_2|^2 \,\ldots\, |f_R|^2\big]^T\Big) + \sigma^2 \mathbf{I}_R \in \mathbb{S}^R_{++},$$
$$\mathbf{R} \triangleq P_0 \mathbf{h}\mathbf{h}^H \in \mathbb{S}^R_{+}, \text{ with } \mathbf{h} \triangleq \big[f_1 g_1\; f_2 g_2 \,\ldots\, f_R g_R\big]^T \text{and}$$
$$\mathbf{Q} \triangleq \sigma^2 \text{diag}\Big(\big[|g_1|^2\, |g_2|^2 \,\ldots\, |g_R|^2\big]^T\Big) \in \mathbb{S}^R_{++}.$$

The optimization problem of **Eq. 3** is *always feasible, as long as* P_c *is nonnegative*, and the optimal value of **Eq. 3** can be expressed in closed form as (Havary-Nassab et al., 2008b)

$$\begin{aligned} V_t &\equiv V(\mathbf{p}(t), t) \\ &\triangleq P_c \lambda_{max}\Big(\big(\sigma_D^2 \mathbf{I}_R + P_c \mathbf{D}^{-1/2}\mathbf{Q}\mathbf{D}^{-1/2}\big)^{-1} \mathbf{D}^{-1/2}\mathbf{R}\mathbf{D}^{-1/2}\Big), \end{aligned}$$

for all $t \in \mathbb{N}^+_{N_T}$, which can be further written as (Zheng et al., 2009)

$$\begin{aligned} V_t &\equiv \sum_{i\in\mathbb{N}^+_R} \frac{P_c P_0 |f(\mathbf{p}_i(t),t)|^2 |g(\mathbf{p}_i(t),t)|^2}{P_0 \sigma_D^2 |f(\mathbf{p}_i(t),t)|^2 + P_c \sigma^2 |g(\mathbf{p}_i(t),t)|^2 + \sigma^2 \sigma_D^2} \\ &\triangleq \sum_{i\in\mathbb{N}^+_R} V_I(\mathbf{p}_i(t),t), \quad \forall t \in \mathbb{N}^+_{N_T}. \end{aligned}$$

The above analytical expression of the optimal value V_t in terms of relay positions and their corresponding channel magnitudes will be key in our subsequent development.

3 STOCHASTIC PROGRAMMING FOR MYOPIC RELAY CONTROL

During time slot $t-1$, we need to determine the relay positions for time slot t, so that we achieve the maximum V_t. However, at time slot $t-1$, we only know $\mathcal{C}(\mathcal{T}_{t-1})$, which does not include information on the CSI that will be experienced during time slot t. Therefore, exactly optimizing the relay positions at the next time slot seems to be an impossible task.

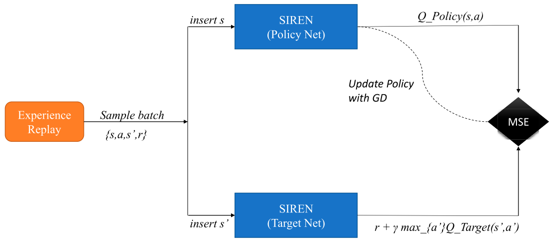


FIGURE 2 | Figure for visualizing the Pipeline of the deep Q learning with SIRENs approach.

Since deterministic optimization of V_t with respect to $\mathbf{p}(t)$ is not possible to be carried out during time slot $t-1$, we can alternatively optimize a projection of V_t onto the space of all measurable functions of $\mathscr{C}(\mathcal{T}_{t-1})$ Kalogerias and Petropulu, (2018). Since, for every $\mathbf{p}(t) \in \mathcal{S}^R$, V_t is of finite variance, we can consider orthogonal projections. In other words, we can consider the Minimum Mean-Square Error (MMSE) predictor of V_t given the available information $\mathscr{C}(\mathcal{T}_{t-1})$. We can then optimize the $\mathbb{E}\{V_t \,|\, \mathscr{C}(\mathcal{T}_{t-1})\}$ with respect to the point $\mathbf{p}(t)$, which results in the 2-stage stochastic program (Shapiro et al., 2009)

$$\begin{array}{ll} \underset{\mathbf{p}(t)}{\text{maximize}} & \mathbb{E}\left\{ V_t \equiv \sum_{i\in\mathbb{N}_R^+} V_I(\mathbf{p}_i(t), t) \,\middle|\, \mathscr{C}(\mathcal{T}_{t-1}) \right\}, \\ \text{subject to} & \mathbf{p}(t) \in \mathcal{C}(\mathbf{p}^o(t-1)) \end{array} \tag{4}$$

to be solved at time slot $t-1 \in \mathbb{N}_{N_T-1}^+$, where $\mathbf{p}^o(1) \in \mathcal{S}^R$ is the initial positions of the relays and $\mathcal{C}(\mathbf{p}^o(t-1)) \subseteq \mathcal{S}^R$ denotes spatially feasible neighborhood around point $\mathbf{p}^o(t-1) \in \mathcal{S}^R$, which is the optimal decision vector determined at time slot $t-2 \in \mathbb{N}_{N_T-2}$ For example, $\mathcal{C}$ may be such that it does not allow the relays to collide with each other, or with other obstacles in space at their next slot positions. In general, $\mathcal{C}$ depends on t, but here, for simplicity that dependence is not shown.

The map $\mathcal{C}(\cdot)$ is typically referred to as *finite-valued multifunction*, and we write $\mathcal{C}\colon \mathcal{S}^R \rightrightarrows \mathcal{S}^R$ (Shapiro et al., 2009). Additionally, problems (4) and (3) are referred to as the *first-stage problem* and the *second-stage problem*, respectively (Shapiro et al., 2009). The block diagram of the above described process is shown in **Figure 1**.

As compared to traditional AF beamforming for a static case, our spatially controlled system described above, uses the same CSI as in the stationary case, to predict the optimal beamforming performance in its vicinity in the MMSE sense, and moves to the optimally selected location. The prediction here relies on the aforementioned spatiotemporal channel model. Of course, this requires a sufficiently slowly varying channel relatively to relay motion, which can be guaranteed if the motion is constrained within small steps.

3.1 Motion Policies & the Interchangeability Principle

To assist in the process of understanding the techniques to solve **Eq. 4**, we make note of an important *variational* property of **Eq. 4**, related to the *long-term performance* of the proposed spatially controlled beamforming system. Our discussion pertains to the employment of the so-called *Interchangeability Principle (IP)* (Bertsekas and Shreve, 1978; Bertsekas, 1995; Rockafellar and Wets, 2004; Shapiro et al., 2009; Kalogerias and Petropulu, 2017), also known as the *Fundamental Lemma of Stochastic Control (FLSC)* (Astrom, 1970; Speyer and Chung, 2008) Kalogerias and Petropulu, (2018). The IP refers conditions that allow the interchange of expectation and maximization or minimization in general stochastic programs.

A version of the IP for the first-stage problem of **(4)** is established in (Kalogerias and Petropulu, 2017) Specifically, the IP implies that **(4)** is exchangeable by the variational problem (Kalogerias and Petropulu, 2017)

$$\begin{array}{ll} \underset{\mathbf{p}(t)}{\text{maximize}} & \mathbb{E}\{V_t\} \\ \text{subject to} & \mathbf{p}(t) \in \mathcal{C}(\mathbf{p}^o(t-1)) \\ & \mathbf{p}(t)\,\text{is}\,\mathscr{C}(\mathcal{T}_{t-1}) - \text{measurable} \end{array}, \tag{5}$$

to be solved at each $t-1 \in \mathbb{N}_{N_T-1}^+$. Upon comparing **Eq. 5** and the original problem **Eq. 4** one can see that, the former problem includes optimization of the *unconditional expectation* of V_t over all (measurable) mappings of the variables generating $\mathcal{C}(\mathcal{T}_{t-1})$ to $\mathcal{C}(\mathbf{p}^o(t-1))$. "This implies that, in **Eq. 5**, $\mathbf{p}(t)$ is a function of all CSI and motion controls up to and including time slot $t-1$, whereas, in **Eq. 4**, $\mathbf{p}(t)$ is a *point*, since all variables generating $\mathcal{C}(\mathcal{T}_{t-1})$ are *fixed before decision making*. Aligned with the literature, any feasible decision $\mathbf{p}(t)$ in **Eq. 5** will be called an (*admissible*) *policy*, or a *decision rule*. *Exchangeability* of **Eqs. 4**, **5** is understood in the sense that the optimal value of **Eq. 5**, which is a number, coincides with the *expectation* of the optimal value of **Eq. 4**, which is a measurable function of $\mathcal{C}(\mathcal{T}_{t-1})$ (and fixed for every realization of the variables generating $\mathcal{C}(\mathcal{T}_{t-1})$). In other words, maximization is *interchangeable* with integration, in the sense that" (Kalogerias and Petropulu, 2017)

$$\sup_{\mathbf{p}(t)\in\mathcal{D}_t} \mathbb{E}\{V_t\} \equiv \mathbb{E}\left\{ \sup_{\mathbf{p}(t)\in\mathcal{C}(\mathbf{p}^o(t-1))} \mathbb{E}\{V_t \,|\, \mathscr{C}(\mathcal{T}_{t-1})\} \right\},$$

for all $t \in \mathbb{N}_{N_T}^2$, where $\mathcal{D}_t$ denotes the set of feasible decisions for (**Eq. 5**). Furthermore, due to our assumption that the control space $\mathcal{S}$ is finite, the IP guarantees that an optimal solution to the original stochastic program (**Eq. 4**) is also feasible and thus, optimal, for (**Eq. 5**).

$$\boldsymbol{m}_{1:\,t-1} \triangleq \left[\boldsymbol{F}^T(1)\,\boldsymbol{G}^T(1)\,\ldots\,\boldsymbol{F}^T(t-1)\,\boldsymbol{G}^T(t-1)\right]^{\boldsymbol{T}} \in \mathbb{R}^{2R(t-1)\times 1} \tag{6}$$

$$\boldsymbol{\mu}_{1:\,t-1} \triangleq \left[\boldsymbol{\alpha}_S(\mathbf{p}(1))\,\boldsymbol{\alpha}_D(\mathbf{p}(1))\,\ldots\,\boldsymbol{\alpha}_S(\mathbf{p}(t-1))\,\boldsymbol{\alpha}_D(\mathbf{p}(t-1))\right]^{\boldsymbol{T}} \ell \in \mathbb{R}^{2R(t-1)\times 1} \tag{7}$$

$$\boldsymbol{c}_{1:\,t-1}^{F(G)}(\mathbf{p}) \triangleq \left[\boldsymbol{c}_1^{F(G)}(\mathbf{p})\,\ldots\,\boldsymbol{c}_{t-1}^{F(G)}(\mathbf{p})\right] \in \mathbb{R}^{1\times 2R(t-1)} \tag{8}$$

$$\boldsymbol{c}_k^{F(G)}(\mathbf{p}) \triangleq \left[\left\{\mathbb{E}\left\{\sigma_{S(D)}(\mathbf{p},t)\sigma_S^j(k)\right\}\right\}_{j\in\mathbb{N}_R^+}\left\{\mathbb{E}\left\{\sigma_{S(D)}(\mathbf{p},t)\sigma_D^j(k)\right\}\right\}_{j\in\mathbb{N}_R^+}\right] \in \mathbb{R}^{1\times 2R},\ \forall k\in\mathbb{N}_{t-1}^+ \tag{9}$$

$$\boldsymbol{\Sigma}_{1:\,t-1} \triangleq \begin{bmatrix} \boldsymbol{\Sigma}(1,1) & \cdots & \boldsymbol{\Sigma}(1,t-1) \\ \vdots & \ddots & \vdots \\ \boldsymbol{\Sigma}(t-1,1) & \cdots & \boldsymbol{\Sigma}(t-1,t-1) \end{bmatrix} \in \mathbb{S}_{++}^{2R(t-1)} \tag{10}$$

3.2 Near-Optimal Beamformer Motion Control

One can readily observe that the problem of **(4)** is separable. Given that, for each $t \in \mathbb{N}_{N_T-1}^+$, decisions taken and CSI collected so far are available to all relays, **(4)** can be solved in a distributed fashion at the relays, with the ith relay being responsible for solving the problem (Kalogerias and Petropulu, 2018)

$$\begin{array}{ll} \underset{\mathbf{p}}{\text{maximize}} & \mathbb{E}\left\{V_I(\mathbf{p},t)\,\middle|\,\mathscr{C}(\mathcal{T}_{t-1})\right\} \\ \text{subject to} & \mathbf{p} \in \mathcal{C}_i(\mathbf{p}^o(t-1)) \end{array}, \tag{11}$$

at each $t-1 \in \mathbb{N}^+_{N_T-1}$, where $\mathcal{C}_i\colon \mathbb{R}^2 \rightrightarrows \mathbb{R}^2$ denotes the corresponding section of $\mathcal{C}$, for each $i \in \mathbb{N}^+_R$. Note that no local exchange of intermediate results is required among relays; given the available information, each relay independently solves its own subproblem. It is also evident that apart from the obvious difference in the feasible set, the optimization problems at each of the relays are identical.

However, the objective of problem **Eq. 11** is impossible to obtain analytically, and it is necessary to resort to some well behaved and computationally efficient *surrogates*. Next, we present *a near-optimal* such approach. The said approach relies on *global* function approximation techniques, and achieves excellent empirical performance.

The proposed approximation to the stochastic program (11) will be based on the following technical, though simple, result.

Lemma 1 (Big Expectations) (Kalogerias and Petropulu, 2018) *Under the assumptions of the wireless channel model, it is true that, at any* $\mathbf{p} \in \mathcal{S}$,

$$\begin{bmatrix} F(\mathbf{p},t) \\ G(\mathbf{p},t) \end{bmatrix} \Bigg| \mathscr{C}(\mathcal{T}_{t-1}) \sim \mathcal{N}\left(\boldsymbol{\mu}^{F,G}_{t\,|\,t-1}(\mathbf{p}), \boldsymbol{\Sigma}^{F,G}_{t\,|\,t-1}(\mathbf{p})\right),$$

for all $t \in \mathbb{N}^2_{N_T}$, *and where we define*

$$\boldsymbol{\mu}^{F,G}_{t|t-1}(\mathbf{p}) \triangleq \left[\mu^F_{t\,|\,t-1}(\mathbf{p})\,\mu^G_{t\,|\,t-1}(\mathbf{p})\right]^{\boldsymbol{T}},$$
$$\mu^F_{t|t-1}(\mathbf{p}) \triangleq \alpha_S(\mathbf{p})\ell + \boldsymbol{c}^F_{1:\,t-1}(\mathbf{p})\boldsymbol{\Sigma}^{-1}_{1:\,t-1}(\boldsymbol{m}_{1:\,t-1} - \boldsymbol{\mu}_{1:\,t-1}) \in \mathbb{R},$$
$$\mu^G_{t|t-1}(\mathbf{p}) \triangleq \alpha_D(\mathbf{p})\ell + \boldsymbol{c}^G_{1:\,t-1}(\mathbf{p})\boldsymbol{\Sigma}^{-1}_{1:\,t-1}(\boldsymbol{m}_{1:\,t-1} - \boldsymbol{\mu}_{1:\,t-1}) \in \mathbb{R} \quad \textit{and}$$

$$\boldsymbol{\Sigma}^{F,G}_{t|t-1}(\mathbf{p}) \triangleq \begin{bmatrix} \eta^2 + \sigma^2_\xi & \eta^2 e^{-\frac{\|\mathbf{p}_S - \mathbf{p}_D\|_2}{\delta}} \\ \eta^2 e^{-\frac{\|\mathbf{p}_S - \mathbf{p}_D\|_2}{\delta}} & \eta^2 + \sigma^2_\xi \end{bmatrix} - \begin{bmatrix} \boldsymbol{c}^F_{1:\,t-1}(\mathbf{p}) \\ \boldsymbol{c}^G_{1:\,t-1}(\mathbf{p}) \end{bmatrix} \boldsymbol{\Sigma}^{-1}_{1:\,t-1} \begin{bmatrix} \boldsymbol{c}^F_{1:\,t-1}(\mathbf{p}) \\ \boldsymbol{c}^G_{1:\,t-1}(\mathbf{p}) \end{bmatrix}^{\boldsymbol{T}} \in \mathbb{S}^2_{++},$$

with $\boldsymbol{m}_{1:t-1}$, $\boldsymbol{\mu}_{1:t-1}$, $\boldsymbol{c}^F_{1:\,t-1}(\mathbf{p})$, $\boldsymbol{c}^G_{1:\,t-1}(\mathbf{p})$, $\boldsymbol{c}^F_k(\mathbf{p})$, $\boldsymbol{c}^G_k(\mathbf{p})$ *and* $\boldsymbol{\Sigma}_{1:t-1}$ *defined as in* (6), (7), (8), (9), *and* (10) *respectively, for all* $(\mathbf{p},t) \in \mathcal{S} \times \mathbb{N}^2_{N_T}$. *Further, for every choice of* $(m,n) \in \mathbb{Z} \times \mathbb{Z}$, *the conditional correlation of the fields* $|f(\mathbf{p},t)|^m$ *and* $|g(\mathbf{p},t)|^n$ *relative to* $\mathcal{C}(\mathcal{T}_{t-1})$ *may be expressed in closed form as*

$$\mathbb{E}\left\{|f(\mathbf{p},t)|^m |g(\mathbf{p},t)|^n \,\middle|\, \mathscr{C}(\mathcal{T}_{t-1})\right\} \equiv 10^{(m+n)\rho/20} \exp\left(\frac{\log(10)}{20}\begin{bmatrix} m \\ n \end{bmatrix}^{\boldsymbol{T}} \boldsymbol{\mu}^{F,G}_{t\,|\,t-1}(\mathbf{p}) + \left(\frac{\log(10)}{20}\right)^2 \begin{bmatrix} m \\ n \end{bmatrix}^{\boldsymbol{T}} \boldsymbol{\Sigma}^{F,G}_{t|t-1}(\mathbf{p}) \begin{bmatrix} m \\ n \end{bmatrix}\right),$$

at any $\mathbf{p} \in \mathcal{S}$ *and for all* $t \in \mathbb{N}^2_{N_T}$.

The detailed description of the proposed technique for efficiently approximating our base problem **(11)** now follows.

Sample Average Approximation (SAA): This is a direct Monte Carlo approach, where, *at worst*, existence of a *sampling, or pseudosampling mechanism at each relay* is assumed, capable of generating samples from a bivariate Gaussian measure. We may then observe that the objective of **Eq. 11** can be represented, for all $t \in \mathbb{N}^2_{N_T}$, via a Lebesgue integral as

$$\mathbb{E}\left\{V_I(\mathbf{p},t)\,\middle|\,\mathscr{C}(\mathcal{T}_{t-1})\right\} = \int_{\mathbb{R}^2} r(\boldsymbol{x})\mathcal{N}\left(\boldsymbol{x}; \boldsymbol{\mu}^{F,G}_{t\,|\,t-1}(\mathbf{p}), \boldsymbol{\Sigma}^{F,G}_{t\,|\,t-1}(\mathbf{p})\right)\mathrm{d}\boldsymbol{x},$$

for any choice of $\mathbf{p} \in \mathcal{S}$, where $\mathcal{N}(\cdot; \boldsymbol{\mu}, \boldsymbol{\Sigma})\colon \mathbb{R}^2 \to \mathbb{R}_{++}$ denotes the bivariate Gaussian density, with mean $\boldsymbol{\mu} \in \mathbb{R}^{2\times1}$ and covariance $\boldsymbol{\Sigma} \in \mathbb{S}^{2\times2}_+$, and the function $r\colon \mathbb{R}^2 \to \mathbb{R}_{++}$ is defined as

$$r(\boldsymbol{x}) \triangleq \frac{P_c P_0 10^{\rho/10}\left[\exp(x_1 + x_2)\right]^\varsigma}{P_0\sigma^2_D\left[\exp(x_1)\right]^\varsigma + P_c\sigma^2\left[\exp(x_2)\right]^\varsigma + 10^{-\frac{\rho}{10}}\sigma^2\sigma^2_D},$$

for all $\boldsymbol{x} \equiv (x_1, x_2) \in \mathbb{R}^2$, where $\varsigma \triangleq \log(10)/10$. By a simple change of variables, it is also true that

$$\mathbb{E}\left\{V_I(\mathbf{p},t)\,\middle|\,\mathscr{C}(\mathcal{T}_{t-1})\right\} = \int_{\mathbb{R}^2} r\left(\sqrt{\boldsymbol{\Sigma}^{F,G}_{t\,|\,t-1}(\mathbf{p})}\,\boldsymbol{x} + \boldsymbol{\mu}^{F,G}_{t\,|\,t-1}(\mathbf{p})\right)\mathcal{N}(\boldsymbol{x}; \mathbf{0}, \mathbf{I}_2)\mathrm{d}\boldsymbol{x},$$

for all $\mathbf{p} \in \mathcal{S}$ and $t \in \mathbb{N}^2_{N_T}$.

Now, for each relay $i \in \mathbb{N}^+_R$, at each $t \in \mathbb{N}^+_{N_T-1}$ and for some $S \in \mathbb{N}^+$, let $\{\boldsymbol{x}^j_{i,t}\}_{j\in\mathbb{N}^+_S}$ be a sequence of independent random elements in $\mathbb{R}^2$, such that $\boldsymbol{x}^j_{i,t} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_2)$, for all $j \in \mathbb{N}^+_S$. We also assume that all such sequences are mutually independent of the channel fields F and G. Then, by defining the sample average estimate

$$\mathsf{S}_S(\mathbf{p},t) \triangleq \frac{1}{S}\sum_{j\in\mathbb{N}^+_S} r\left(\sqrt{\boldsymbol{\Sigma}^{F,G}_{t\,|\,t-1}(\mathbf{p})}\,\boldsymbol{x}^i_{j,t-1} + \boldsymbol{\mu}^{F,G}_{t\,|\,t-1}(\mathbf{p})\right),$$

the SAA of our initial problem **Eq. 11** is formulated as

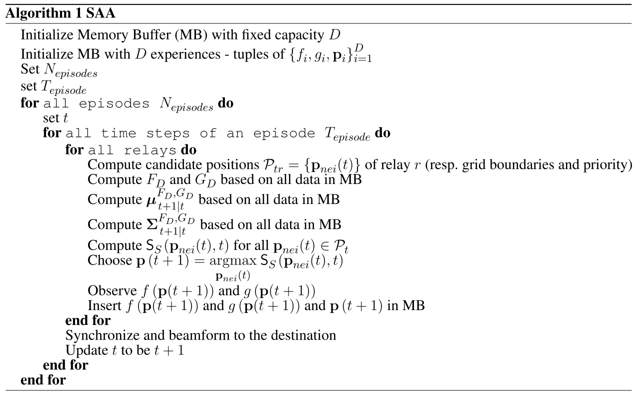
$$\begin{array}{ll} \underset{\mathbf{p}}{\text{maximize}} & \mathsf{S}_S(\mathbf{p},t) \\ \text{subject to} & \mathbf{p} \in \mathcal{C}_i(\mathbf{p}^o(t-1)) \end{array}, \tag{12}$$

at relay $i \in \mathbb{N}^+_R$, solved at each $t-1 \in \mathbb{N}^+_{N_T-1}$. A detailed analysis of the SAA problem **Eq. 12** is out of the scope of our discussion herein. Still, it is worth mentioning that the feasible of set of **Eq. 12** is finite, and therefore its optimal solution possesses various strong asymptotic guarantees in terms of convergence to the optimal solution of the original problem, as $S \to \infty$. For further details, see (Shapiro et al. (2009), Chapter 5).

On the downside, computing the objective of the SAA problem **Eq. 12** assumes availability of Monte Carlo samples, which could be restrictive in certain scenarios. Nevertheless, assuming mutual independence of the sequences $\{\boldsymbol{x}^j_{i,t}\}_j$, for each i and each t is not required. In fact, one could generate one sequence for all relays, per time slot, or even better, one sequence for all relays, for all time slots altogether. Such sampling schemes are legitimate, for two reasons. First, all SAAs of the form **Eq. 12** are solved independently for each relay and at each time slot. Second, Monte Carlo sampling is by construction statistically independent from the spatiotemporal channel fields F and G. As a result, such sampling schemes relax (in fact, eliminate) the need for (pseudo)random sampling at each *individual* relay. This makes them particularly attractive for practical purposes.

We denote this approach as SAA for the rest of the paper. The control flow of the SAA is presented in Algorithm 1.

Algorithm 1. SAA



```
Algorithm 1 SAA
Initialize Memory Buffer (MB) with fixed capacity $D$
Initialize MB with $D$ experiences - tuples of $\{f_i, g_i, \mathbf{p}_i\}_{i=1}^{D}$
Set $N_{episodes}$
set $T_{episode}$
for all episodes $N_{episodes}$ do
    set $t$
    for all time steps of an episode $T_{episode}$ do
        for all relays do
            Compute candidate positions $\mathcal{P}_{tr} = \{\mathbf{p}_{nei}(t)\}$ of relay $r$ (resp. grid boundaries and priority)
            Compute $F_D$ and $G_D$ based on all data in MB
            Compute $\boldsymbol{\mu}_{t+1|t}^{F_D,G_D}$ based on all data in MB
            Compute $\boldsymbol{\Sigma}_{t+1|t}^{F_D,G_D}$ based on all data in MB
            Compute $\mathsf{S}_S(\mathbf{p}_{nei}(t), t)$ for all $\mathbf{p}_{nei}(t) \in \mathcal{P}_t$
            Choose $\mathbf{p}(t+1) = \underset{\mathbf{p}_{nei}(t)}{\operatorname{argmax}} \mathsf{S}_S(\mathbf{p}_{nei}(t), t)$
            Observe $f(\mathbf{p}(t+1))$ and $g(\mathbf{p}(t+1))$
            Insert $f(\mathbf{p}(t+1))$ and $g(\mathbf{p}(t+1))$ and $\mathbf{p}(t+1)$ in MB
        end for
        Synchronize and beamform to the destination
        Update $t$ to be $t+1$
    end for
end for
```

4 DEEP REINFORCEMENT LEARNING FOR ADAPTIVE DISCRETE RELAY MOTION CONTROL

4.1 Dynamic Programming for Relay Motion Control

The previously mentioned approach tackles the problem of relay motion control from a myopic perspective in the sense that the stochastic program is formulated so as to select the relay positions for the subsequent time slot with the goal of maximizing the collective SINR at the destination only for that particular slot.

The employment of reinforcement learning for the problem of discrete relay motion control entails that we reformulate the problem as a dynamic program. In this set up we want, at time slot $t-1$, to derive a motion policy (a methodology for choosing the relays' displacement) so as to maximize the discounted sum of V_Is (in expectation) from the subsequent time step t to the infinite horizon.

To formally pose that program we need to introduce a Markov Decision Process (MDP). The MDP is a tuple defined as $\{\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma\}$ (Sutton and Barto, 2018):

The formulation of the dynamic program is as follows:

If γ is a discount factor, we can formulate the infinite horizon relay control problem as:

$$\begin{aligned} \underset{\mathbf{u}(t), t\geq 0}{\text{maximize}} \quad & \mathbb{E}\left\{\sum_{t=1}^{\infty} \gamma^{t-1} \sum_{r=1}^{R} V_I(\mathbf{p}(t), t)\right\} \\ \text{subject to} \quad & \begin{bmatrix} \mathbf{C}(t) \\ \mathbf{p}(t) \end{bmatrix} = \begin{bmatrix} e^{-1/c_2}\mathbf{C}(t-1) + \mathbf{W}(t) \\ \mathbf{p}(t-1) + \mathbf{u}(t) \end{bmatrix}, \\ & \mathbf{u}(t) \in \mathcal{A} \text{ is a function of } \mathscr{C}(\mathcal{T}_{t-1}) \end{aligned} \tag{13}$$

where $\mathbf{u}(t)$ is the control at time t (essentially determining the relay displacement), and the driving noise $\mathbf{W}(t)$ is distributed as $\mathcal{N}(0, (1-e^{-2/c_2})\boldsymbol{\Sigma}_{\mathbf{C}})$ and $\mathbf{C}(0) \sim \mathcal{N}(0, \boldsymbol{\Sigma}_{\mathbf{C}})$. $\boldsymbol{\Sigma}_{\mathbf{C}}$ is the covariance matrix for all channels (source and destination) for all the cells in the grid. The said covariance matrix is explicitly defined in (Kalogerias and Petropulu, 2017) and admits a particular form if the channels evolve according to the spatiotemporal Gaussian process defined in 2.2.

Now, either the above problem defines a MDP or POMDP is dependent on the history $\mathcal{C}(\mathcal{T}_t)$. In particular, if $\mathcal{C}(\mathcal{T}_t)$ is generated by the whole state vector at each time slot then it is easy to see that problem **Eq. 13** is fully observable, since all CSI generated by the environment is available for the relays to exploit for deciding upon the subsequent displacement.

On the other hand, if $\mathcal{C}(\mathcal{T}_t)$ is generated by the relay decisions together with only their local observations by their trajectories, then problem **Eq. 13** becomes partially observable. Specifically, partial observability may be thought of as a dynamic observation selection process, which only reveals CSI pertaining to the trajectory of each relay, keeping the rest of the CSI hidden from the decision making process.

4.2 Deep Q Learning for Discrete Relay Motion Control

The employment of deep Q learning for relay motion control expels the need for making particular assumption for the underlying correlation structure of the channels.

Taking into account the (12) one can infer that we can construct a single policy that is learned by the collective experience of all the agents/relays and it constitutes the single policy that the movement of all relays strictly adhere to. In that spirit, we instantiate one neural network to parameterize the state-action value function (Q) and it is being trained on the experiences of all the relay. The motion policy is ϵ-greedy with respect to the estimation of the Q function.

Initially, we adopt the deep Q learning algorithm as described in (Mnih et al., 2015) and illustrated in **Figure 2**. Even though, as we pointed out in the previous subsection, the state of the MDP is the concatenation of the relay position $\mathbf{p} = \boldsymbol{s}$ and the channels $f(\mathbf{p}, t)$ and g $(\mathbf{p}, t)$, we follow a slightly different approach in the adoption of deep Q learning. In particular, the input to the neural network is the concatenation of the position $\mathbf{p} = [x, y]$ and the time step t. We should note at this point that augmenting the neural network input with the timestamp of the transition is a differentiation between the algorithm presented in this current work and the solution proposed in Evmorfos et al. (2022). This alternative, even though does not affect the implementation much, provides measurable improvements in cases where the power of the shadowing is strong. The reward r is the contribution of the relay to the SINR at the destination during the respective time step (V_I). At each time slot the relay selects an action $a \in \mathcal{A}_{\text{full}}$.

In general, Q learning with rich function approximators such as neural networks requires some heuristics for stability. The first such heuristic is the *Experience Replay* (Mnih et al., 2015). Each tuple of experience for a relay, namely $\{state, action, next\ state, reward\} \equiv \boldsymbol{s}, a, \boldsymbol{s}', r\}$, is stored in a memory This memory we denote as Experience Replay. For the neural network updates, we sample uniformly a batch of experiences from the Experience Replay and use that batch to perform gradient descent to estimate the Q function (and subsequently the decision-making policy).

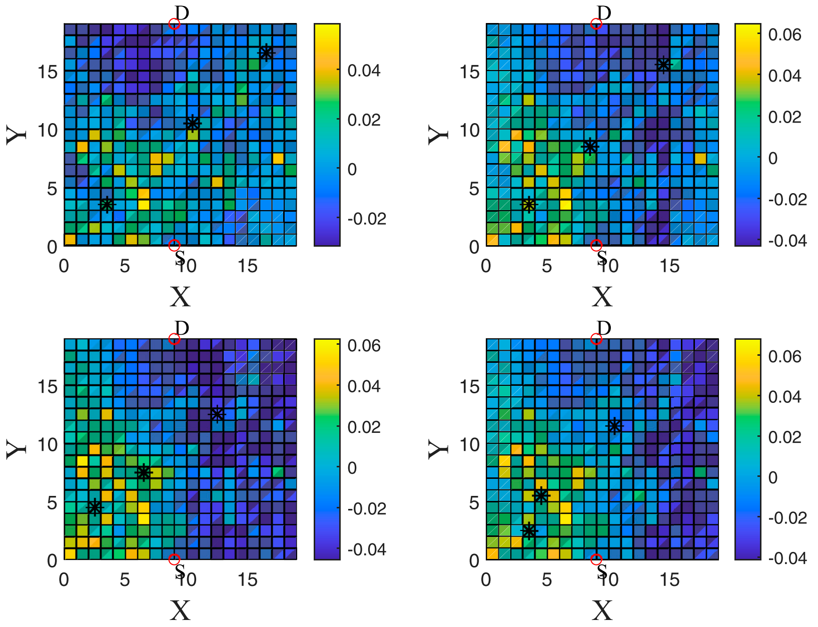


FIGURE 3 | This is a heatmap for visualizing a trajectory of the relays. We can see the V_l for all grid cells for four different time steps (each time step has a 2-time-slot difference with the previous and the next). One can see the positions of the relays for every time slot. The relays are moving towards better and better positions (larger V_ls).

The second heuristic is the *Target Network* (Mnih et al., 2015). The Target Network (Q_{target} ($\boldsymbol{s}'$, a'; $\boldsymbol{\theta}^-$)) provides the estimation for the targets (labels) for the updates of the *Policy Network* (Q_{policy} ($\boldsymbol{s}'$, a'; $\boldsymbol{\theta}^+$)), i.e., the network used for estimating the Q function. The two networks share (typically) the same architecture. We do not update the Target Network's weights with any optimization scheme, but, after a predefined number of training steps, the weights of the Policy Network are copied to the Target Network. This provides stationary targets for the weight updates and brings the task of the Q function approximation closer to a supervised learning paradigm.

Therefore, at each update step we sample a batch of experiences from the Experience Replay and use the batch to perform gradient descent on the loss:

$$\mathcal{L} = \left(Q_{policy}\left(\boldsymbol{s}, a; \boldsymbol{\theta}^+\right) - \left(r + \gamma \max_{a'} Q_{target}\left(\boldsymbol{s}', a'; \boldsymbol{\theta}^-\right)\right)\right)^2.$$

At each step, the Policy Network's weights are updated according to:

$$\boldsymbol{\theta}^+_{t+1} = \boldsymbol{\theta}^+_t + \lambda\left(Y_t - Q_{policy}\left(\boldsymbol{s}, a; \boldsymbol{\theta}^+_t\right)\right)\nabla_{\boldsymbol{\theta}^+_t} Q\left(\boldsymbol{s}, a; \boldsymbol{\theta}^+_t\right).$$

where,

$$Y_t = r + \gamma \max_{a'} Q_{target}\left(\boldsymbol{s}', a'; \boldsymbol{\theta}^-_t\right)$$

The parameter λ is the learning rate. The parameter γ is a scalar called the *discount factor* and γin (0, 1). The choice for the discount factor pertains to a trade off between the importance assigned to long term rewards and the importance assigned to short term rewards. The parameters a, $a' \in \mathcal{A}_{\text{full}}$ correspond to the action chosen during the current state and the action chosen for the next state (the state during the next time slot). Also, $\boldsymbol{s}$ and $\boldsymbol{s}'$ correspond to the current state and the next state respectively. The general pipeline of the deep Q learning algorithm is defined in **Figure 3**.

When the relays move (the do not stay in the same grid cell for two consecutive slots), they require additional energy consumption. i some cases though, the diplacement to a neighboring grid cell does not correspond to significant improvement in terms of the cumulative SINR at the destination. Therefore, to account for the energy used for the application, we choose to not perform the ϵ-greedy policy directly on the estimates Q_{policy} ($\boldsymbol{s}$, a; $\boldsymbol{\theta}^+$) of the Q function, but we decrease the estimates for all actions a, except for the action $stay$, by a small percentage μ. In that way we prohibit the relay displacement if this action does not correspond to a significant increase in the expectation of the cumulative sum of rewards (SINR). How significant this displacement action should be for it to be performed pertains to the choice of μ. For our simulations, in the subsequent sections, we choose μ to be 1%.

4.3 Sinusoidal Representation Networks for Q Function Parameterization

There have been many recent works which convincingly claim that coordinate-based Multilayer Perceptron Neural Networks (MLPs), i.e., MLPs that map a vector of coordinates to a low-dimensional natural signal, fail to learn high frequency components of the said signal. This constitutes a phenomenon that is called the spectral bias in machine learning literature (Jacot et al., 2018; Cao et al., 2019). The work in (Sitzmann et al., 2020)

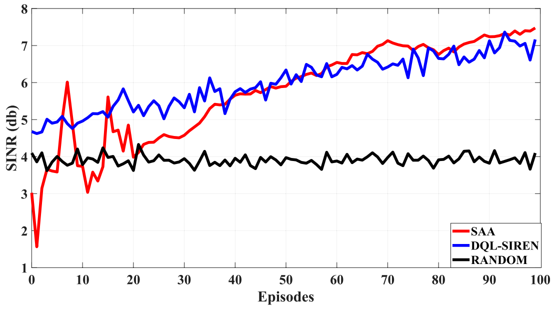


FIGURE 4 | Comparison of the SAA, the DQL-SIREN and the Random policy.

examines the amelioration of spectral bias for MLPs. The inadequacy of MLPs for such inductive biases is bypassed by introducing a variation of the conventional MLP architecture with sinusoid (sin (·)) as activation function between layers. Tis MLP alternative was termed *Sinusoidal Representation Networks* (SIRENs), and was shown, both theoretically and experimentally, to effectively tackle the spectral bias.

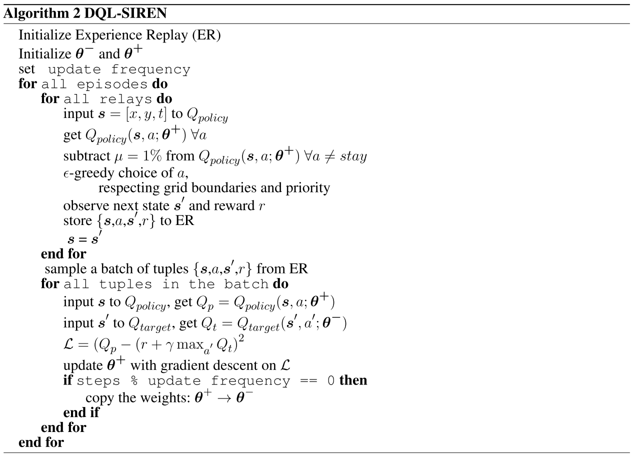
The sinusoid is a periodic function which is quite atypical as a choice for activation function in neural networks. The authors in (Sitzmann et al., 2020) propose the employment of weight initialization framework so that the distribution of activations is retained during training and convergence is achieved without the network oscillating.

In particular, if we assume an intermediate layer of the neural network with input $\boldsymbol{x} \in \mathbb{R}^n$, then the output is an affine transformation using the weights $\boldsymbol{w}$ passed through the sinusoid activation, therefore the output is $\sin(\boldsymbol{w}^\mathsf{T}\boldsymbol{x} + b)$. Since the layer is not the first layer of the network, the input $\boldsymbol{x}$ is arcsine distributed. With these assumptions it was shown in (Sitzmann et al., 2020) that, if the elements of $\boldsymbol{w}$, namely w_i, are initialized from a uniform distribution $w_i \sim \mathcal{U}(-\sqrt{\frac{6}{n}}, \sqrt{\frac{6}{n}})$, then $\boldsymbol{w}^\mathsf{T}\boldsymbol{x} \sim \mathcal{N}(0,1)$ as n grows. Therefore one should initialize the weights of all intermediate layers with $w_i \sim \mathcal{U}(-\sqrt{\frac{6}{n}}, \sqrt{\frac{6}{n}})$. The neurons of the first layer are initialized with the use of a scalar hyperparameter ω_0, so that the output of the first layer, $\sin(\omega_0 \boldsymbol{Wx} + b)$ spans multiple periods over $[-1, 1]$. $\boldsymbol{W}$ is a matrix whose elements correspond to the weights of the first layer.

When we adopt the deep Q learning approach for discrete relay motion control, we basically train a neural network (MLP) to learn a low-dimensional natural signal from coordinates, namely the state-action value function Q ($\boldsymbol{s}$, a). The Q function, Q ($\boldsymbol{s}$, a), represents the sum of SINR at the destination that the relays are expected to achieve for an infinite time horizon, starting from the respective position $\boldsymbol{s}$ and performing action a. The Policy Network, being a coordinate MLP may not be able to converge for the high frequency components of the underlying Q function that arise from the fact that the channels exhibit very abrupt spatiotemporal variations.

Therefore we propose that both the Policy and the Target Networks are SIRENs. The control flow of the algorithm we propose is given in Algorithm 2. We denote this as DQL-SIREN, which stands for *Deep Q Learning with Sinusoidal Representation Networks*.

Algorithm 2. DQL-SIREN



```
Algorithm 2 DQL-SIREN
Initialize Experience Replay (ER)
Initialize $\boldsymbol{\theta}^-$ and $\boldsymbol{\theta}^+$
set update frequency
for all episodes do
    for all relays do
        input $\boldsymbol{s} = [x, y, t]$ to $Q_{policy}$
        get $Q_{policy}(\boldsymbol{s}, a; \boldsymbol{\theta}^+)\ \forall a$
        subtract $\mu = 1\%$ from $Q_{policy}(\boldsymbol{s}, a; \boldsymbol{\theta}^+)\ \forall a \neq stay$
        $\epsilon$-greedy choice of $a$,
            respecting grid boundaries and priority
        observe next state $\boldsymbol{s}'$ and reward $r$
        store $\{\boldsymbol{s}, a, \boldsymbol{s}', r\}$ to ER
        $\boldsymbol{s} = \boldsymbol{s}'$
    end for
    sample a batch of tuples $\{\boldsymbol{s}, a, \boldsymbol{s}', r\}$ from ER
    for all tuples in the batch do
        input $\boldsymbol{s}$ to $Q_{policy}$, get $Q_p = Q_{policy}(\boldsymbol{s}, a; \boldsymbol{\theta}^+)$
        input $\boldsymbol{s}'$ to $Q_{target}$, get $Q_t = Q_{target}(\boldsymbol{s}', a'; \boldsymbol{\theta}^-)$
        $\mathcal{L} = (Q_p - (r + \gamma \max_{a'} Q_t))^2$
        update $\boldsymbol{\theta}^+$ with gradient descent on $\mathcal{L}$
        if steps % update frequency == 0 then
            copy the weights: $\boldsymbol{\theta}^+ \rightarrow \boldsymbol{\theta}^-$
        end if
    end for
end for
```

5 SIMULATIONS

We test our proposed schemes by simulating a 20, ×, 20 m grid. All the grid cells are $1m \times 1m$. The number of agents/relays that assist the single source destination communication pair is $R = 3$. For every time slot the position of each relay is constrained within the boundaries of the gridded region and also constrained to adhere to a predetermined relay movement priority. Only one relay can occupy a grid cell per time slot. The center of the relay/agent and the center of the respective grid cell coincide.

When it comes to the shadowing part of our assumed channel model, we define a threshold θ which quantifies the distance in time and space where the shadowing component is important and can be taken into account for the construction of the motion policy. We assume that the shadowing power $\eta^2 = 15$ and the autocorrelation distance is $c_1 = 10m$ and the autocorrelation time is $c_2 = 20sec$. The variances of noises at the relays and destination are fixed as $\sigma^2 \equiv \sigma_D^2 \equiv 1$.The source and destination are fixed at $\mathbf{p}_S \equiv [10\,0]^T$ and $\mathbf{p}_D \equiv [10\,20]^T$.

Each one of the relays can move 1 grid cell/time slot and the size of each cell is $1m \times 1m$ (as mentioned before). The time slot length is set to be $0.6sec$. Therefore the calculation of the channel and the decision of the movement for each relay should take up an amount of time that is strictly less than the duration of the time interval.

5.1 Specifications for the DQL-SIREN and the SAA

Regarding the DQL-SIREN, we employ SIRENs for both the Policy and the Target Networks. Each SIREN is comprised by three dense layers (350 neurons for each layer) and the learning rate is $1e-4$.

TABLE 1 | Table of comparison between the two methods regarding key features.

Features	SAA	DQL-SIREN
Channel statistics	known (model-based)	unknown (model-free)
Robustness w.r.t seeds	extremely robust	slight variation bt seeds
Memory size	150 transitions	3,000 transitions
Horizon	myopic policies	long horizon policies (for γ close to 1)
Exploration	not required	required (ϵ-greedy)
Best SINR achieved	7.4 *db*	7.2 *db*

The Experience Replay size is 3,000 tuples and we begin every experiment with 300 transitions derived by a completely random policy before the start of training for all the deep Q learning approaches. The ϵ of the ϵ-greedy policy is initialized to be 1 but it is steadily decreased until it gets to 0.1 This is a very typical regime in RL. It is a very simple way to handle the dilemma between exploration and exploitation in RL, where we begin by giving emphasis to exploration first and then gradually exploration is traded for exploitation. We copy the weights of the Policy Network to the weights of the Target Network every 100 steps of training. The batch size is chosen to be 128 (even though the methods work reliably for different batch sizes ranging from 64 to 512) and the discount factor γ is chosen to be 0.99. We want to mention that small values for γ translate to a more myopic agent (an agent that assigns significance to short term rewards at the expense of long term/delayed rewards). On the other hand, values of γ closer to 1 correspond to agents that assign almost equal value to long term rewards and short term rewards. For the deep Q learning methods that we have proposed, we noticed that for low values of γ converence and performance is impeded, something that we attribute to the interplay of Q learning and neural network employment rather than to the nature of the underlying MDP.

We set the ω_0 for the DQL-SIREN to 5 (the performance of the algorithm is robust for different values of the said parameter). Finally, we use the Adam optimizer for updating the network weights.

When it comes to the SAA, the sample size is set to 150 for the experiments.

5.2 Synthesized Data and Simulations

We create synthetic CSI data that adhere to the channel statistics described in 2.2.

In **Figure 4**, we plot the average SINR at the destination (in dB scale) achieved by the cooperation of all three relays, per episode, for 100 episodes, where every episode is comprised by 30 steps. The transmission power of the source is $P_S = 57dbm$ and the relay transmission power budget is $P_R = 57dBm$. The assumed channel parameters are set as $\ell = 2.3, \rho = 3, \eta^2 = 15, \sigma_\xi^2 = 3, c_1 = 10, c_2 = 20, c_3 = 0.5$. The variance of the noise at the relays and destination are $\sigma_D^2 = \sigma^2 = 0.5$.

We generate 3,000 = 100, ×, 30 instances of the source-relay and relay destination channels for the whole grid (20, ×, 20). Every 30 time steps we initialize the relays to random positions in the grid and let them move. We plot the average SINR for every 30 steps of the algorithms.

5.3 Simulation Results and Discussion

We present the results of our simulations in **Figure 4**. As we stated before, the results correspond to the average SINR at the destination for 100 episodes. Each episode consists of 30 time steps. The runs correspond to the average over six different seeds.

We compare three different policies. The first one is the Random policy, where each relay chooses the displacement for the next step at random. The second policy is the DQL-SIREN that solves the dynamic program (maximization of the discounted sum of V_Is for every relay from the current time step to the infinite horizon). The third policy is the myopic SAA that corresponds to the stochastic program and optimizes each individual relay's V_I for the subsequent slot.

As one can see that both the SAA and the DQL-SIREN perform significantly better than the Random policy (they both achieve an average SINR of approximately 7 *db* in contrast to the Random policy that achieves about 4 *db*). **Table 1** contains a head-to-head comparison of the SAA and the DQL-SIREN approaches regarding some qualitative and some quantitative features.

The convergence of the DQL-SIREN is faster than that of SAA. This is reasonable since, when it comes to the SAA approach, for the first five episodes there have not been collected enough samples (150). Both SAA and DQL-SIREN perform approximately the same in terms of average SINR. Towards the end of the experiments there is a small gap between the two (with the SAA performing slightly better). This can be attributed to the ϵ-greedy policy of the DQL-SIREN, where ϵ never goes to zero (choosing a random action a small percentage of the time for maintaining exploration).

There are some interesting inferences that one can make, based on the simulations. First of all, even though the SAA is myopic and only attempts to maximize the SINR for the subsequent time slot, works quite well in the sense of the aggregated statistic of the average SINR. This is a clear indication that, for the formulated problem, being greedy translates to performing adequately in the sense of cumulative reward.

Of course this peculiarity stands true only when the statistics of the channels are completely known and do not change significantly during the operation time. Apparently, in such a scenario, the phenomenon of delayed rewards is not much prevalent.

6 CONCLUSION

In this paper, we examine the discrete motion control for mobile relays facilitating the communication between a source and a destination. We compare two different approaches to tackle the problem. The first approach employs stochastic programming for scheduling the relay motion. This approach is myopic meaning that it seeks to maximize the SINR at the destination, only at the subsequent time slot. In addition, the stochastic programming approach makes specific assumption for the statistics of the channel evolution. The second approach is a deep reinforcement learning approach that is not myopic meaning that its goal is to maximize the discounted sum of SINR at the destination from the subsequent slot to an infinite time horizon. Additionally, the second approach makes no particular assumptions for the channel statistics. We test our methods in synthetic channel data produced in accordance to a known model for spatiotemporally varying channels. Both methods perform similarly and achieve significant improvement in comparison to a standard random policy for relay motion. We also provide a head-to-head comparison of the two approaches regarding various key qualitative and quantitative features. As future work, we plan on extending the current methods for scenarios with multiple source-destination communication pairs and, possibly, include the existence of eavesdroppers.

DATA AVAILABILITY STATEMENT

The raw data supporting the conclusion of this article will be made available by the authors, without undue reservation.

AUTHOR CONTRIBUTIONS

All authors listed have made a substantial, direct, and intellectual contribution to the work and approved it for publication.

FUNDING

Work supported by ARO under grant W911NF2110071.

REFERENCES

Astrom, K. J. (1970). *Introduction to Stochastic Control Theory*, 70. New York: Academic Press.

Barriac, G., Mudumbai, R., and Madhow, U. (2004). "Distributed Beamforming for Information Transfer in Sensor Networks," in Third International Symposium on Information Processing in Sensor Networks, 2004 (IEEE), 81–88. doi:10.1145/984622.984635

Bertsekas, D. (1995). *Dynamic Programming & Optimal Control*. 4th edn., II. Belmont, Massachusetts: Athena Scientific.

Bertsekas, D. P., and Shreve, S. E. (1978). *Stochastic Optimal Control: The Discrete Time Case*, 23. New York: Academic Press.

Cao, Y., Fang, Z., Wu, Y., Zhou, D.-X., and Gu, Q. (2019). Towards Understanding the Spectral Bias of Deep Learning. *arXiv preprint arXiv:1912.01198*.

Chatzipanagiotis, N., Liu, Y., Petropulu, A., and Zavlanos, M. M. (2014). Distributed Cooperative Beamforming in Multi-Source Multi-Destination Clustered Systems. *IEEE Trans. Signal Process.* 62, 6105–6117. doi:10.1109/tsp.2014.2359634

Evmorfos, S., Diamantaras, K., and Petropulu, A. (2021a). "Deep Q Learning with Fourier Feature Mapping for Mobile Relay Beamforming Networks," in 2021 IEEE 22nd International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), 126–130. doi:10.1109/SPAWC51858.2021.9593138

Evmorfos, S., Diamantaras, K., and Petropulu, A. (2021b). "Double Deep Q Learning with Gradient Biasing for Mobile Relay Beamforming Networks," in 2021 55th Asilomar Conference on Signals, Systems, and Computers, 742–746. doi:10.1109/ieeeconf53345.2021.9723405

Evmorfos, S., Diamantaras, K., and Petropulu, A. (2022). Reinforcement Learning for Motion Policies in Mobile Relaying Networks. *IEEE Trans. Signal Process.* 70, 850–861. doi:10.1109/TSP.2022.3141305

Gao, F., Cui, T., and Nallanathan, A. (2008). On Channel Estimation and Optimal Training Design for Amplify and Forward Relay Networks. *IEEE Trans. Wirel. Commun.* 7, 1907–1916. doi:10.1109/TWC.2008.070118

Goldsmith, A. (2005). *Wireless Communications*. Cambridge University Press.

Havary-Nassab, V., Shahbazpanahi, S., Grami, A., and Zhi-Quan Luo, Z.-Q. (2008a). Distributed Beamforming for Relay Networks Based on Second-Order Statistics of the Channel State Information. *IEEE Trans. Signal Process.* 56, 4306–4316. doi:10.1109/tsp.2008.925945

Havary-Nassab, V., ShahbazPanahi, S., Grami, A., and Zhi-Quan Luo, Z.-Q. (2008b). Distributed Beamforming for Relay Networks Based on Second-Order Statistics of the Channel State Information. *IEEE Trans. Signal Process.* 56, 4306–4316. doi:10.1109/TSP.2008.925945

Heath, R. W. (2017). *Introduction to Wireless Digital Communication: A Signal Processing Perspective*. Prentice-Hall.

Jacot, A., Gabriel, F., and Hongler, C. (2018). "Neural Tangent Kernel: Convergence and Generalization in Neural Networks," in Advances in Neural Information Processing Systems. Editors S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (Curran Associates, Inc).31.

Kalogerias, D. S., Chatzipanagiotis, N., Zavlanos, M. M., and Petropulu, A. P. (2013). "Mobile Jammers for Secrecy Rate Maximization in Cooperative Networks," in Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference, 2901–2905. doi:10.1109/ICASSP.2013.6638188

Kalogerias, D. S., and Petropulu, A. P. (2016). "Mobile Beamforming Amp; Spatially Controlled Relay Communications," in 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 6405–6409. doi:10.1109/ICASSP.2016

Kalogerias, D. S., and Petropulu, A. P. (2017). Spatially Controlled Relay Beamforming: 2-stage Optimal Policies. *Arxiv*.

Kalogerias, D. S., and Petropulu, A. P. (2018). Spatially Controlled Relay Beamforming. *IEEE Trans. Signal Process.* 66, 6418–6433. doi:10.1109/tsp.2018.2875896

Li, J., Petropulu, A. P., and Poor, H. V. (2011). Cooperative Transmission for Relay Networks Based on Second-Order Statistics of Channel State Information. *IEEE Trans. Signal Process.* 59, 1280–1291. doi:10.1109/TSP.2010.2094614

Liu, Y., and Petropulu, A. P. (2011). On the Sumrate of Amplify-And-Forward Relay Networks with Multiple Source-Destination Pairs. *IEEE Trans. Wirel. Commun.* 10, 3732–3742. doi:10.1109/twc.2011.091411.101523

MacCartney, G. R., Zhang, J., Nie, S., and Rappaport, T. S. (2013). Path Loss Models for 5G Millimeter Wave Propagation Channels in Urban Microcells. *Globecom*, 3948–3953. doi:10.1109/glocom.2013.6831690

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., et al. (2015). Human-level Control through Deep Reinforcement Learning. *nature* 518, 529–533. doi:10.1038/nature14236

Muralidharan, A., and Mostofi, Y. (2017). "First Passage Distance to Connectivity for Mobile Robots," in Proceedings of the American Control Conference (IEEE), 1517–1523. doi:10.23919/ACC.2017.7963168

Proakis, J. G., and Salehi, M. (2008). *Digital Communications*. McGraw-Hill.

Rockafellar, R. T., and Wets, R. J.-B. (2004). *Variational Analysis*, 317. Springer Science & Business Media.

Shapiro, A., Dentcheva, D., and Ruszczyński, A. (2009). *Lectures on Stochastic Programming*. 2nd edn. Society for Industrial and Applied Mathematics.

Sitzmann, V., Martel, J., Bergman, A., Lindell, D., and Wetzstein, G. (2020). Implicit Neural Representations with Periodic Activation Functions. *Adv. Neural Inf. Process. Syst.* 33, 7462–7473.

Speyer, J. L., and Chung, W. H. (2008). *Stochastic Processes, Estimation, and Control*. Siam.

Sutton, R. S., and Barto, A. G. (2018). *Reinforcement Learning: An Introduction*. MIT press.

Yan, Y., and Mostofi, Y. (2013). Co-optimization of Communication and Motion Planning of a Robotic Operation under Resource Constraints and in Fading Environments. *IEEE Trans. Wirel. Commun.* 12, 1562–1572. doi:10.1109/twc.2013.021213.120138

Yan, Y., and Mostofi, Y. (2012). Robotic Router Formation in Realistic Communication Environments. *IEEE Trans. Robot.* 28, 810–827. doi:10.1109/TRO.2012.2188163

Zheng, G., Wong, K.-K., Paulraj, A., and Ottersten, B. (2009). Collaborative-Relay Beamforming with Perfect CSI: Optimum and Distributed Implementation. *IEEE Signal Process. Lett.* 16, 257–260. doi:10.1109/LSP.2008.2010810

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.