DEEP ACTOR-CRITIC FOR CONTINUOUS 3D MOTION CONTROL IN MOBILE RELAY BEAMFORMING NETWORKS

Spilios Evmorfos and Athina P. Petropulu

Rutgers, The State University of New Jersey, Piscataway, NJ

ABSTRACT

The paper studies the motion control for mobile relays implementing cooperative beamforming to aid the communication between a source-destination pair. We consider an urban communication scenario, where the channels exhibit spatiotemporal correlations and thus can be learned. The relays move in a time-slotted fashion within a three-dimensional cube. During every slot, the relays beamform optimally to maximize the Signal-to-Interference+Noise Ratio (SINR) at the destination and decide their positions for the next slot. Unlike prior works that assume knowledge of channel statistics, our proposed approach is model-free. Also, typically, prior approaches assume discrete motion on the two-dimensional plane. However, as discretization introduces the curse of dimensionality, those methods do not easily extend to three-dimensional motion. We propose a model-free, continuous control actor-critic approach that can be easily applied to 2D and 3D motion with the same complexity . To address the random nature of the channel, we propose to use Sinusoidal Representation Networks (SIRENs) for value function approximation. Our approach outperforms the direct application of the State-of-the-Art continuous control algorithms for both 2D and 3D cases.

Index Terms— Mobile Relay Beamforming, Actor-Critic, Continuous Motion Control, Reinforcement Learning.

1. INTRODUCTION

We consider a scenario in which users need to communicate but due to significant attenuation the communication range is limited. Introducing relays that implement beamforming to the destination can increase the communication range. We are interested in mobile relays, because, when operating in a time varying environment, they can improve the quality of communication by optimally selecting their beamforming positions. An application scenario would be a swarm of drones, deployed over a busy street, performing relay beamforming and aiding vehicle-vehicle or vehicle-infrastructure communications. Such data sharing would require much larger bandwidth than what is available at the 5.9GHz band currently used for vehicular communications. Researchers are looking into the mm-wave band, where more bandwidth is available [1], but higher frequencies experience high attenuation. Relay beamforming is a good approach to extend the communication range in urban mm-wave communications [2, 3, 4].

In [2], a time-slotted, mobile relay scenario is considered. In each slot, the relays beamform to the destination, and then optimally position themselves for the next slot. Beamforming weights and motion policies are obtained so that the SINR at the destination is maximized, subject to a total relay transmission power constraint. For the construction of relay motion policies, one should implicitly or explicitly, exploit channel patterns in time and space. The shadowing propagation effect in urban environments gives rise to spatiotemporal channel correlations that can be leveraged for learning the evolution of the channels in time and space. Along that direction, [2] devises a methodology to exploit the said correlations, which are assumed to be fully known, to estimate the relay motion policies in a predictive fashion. In [2], the motion of the relays is discrete and confined within a 2D rectangular area, discretized into a fine grid. In [5], a Reinforcement Learning (RL) approach for relay motion control is proposed, where again the motion is discrete and constrained in a 2D grid. Specific statistical models for the channels are assumed and the parameters of the said models are learned from observed data. In [6], a discrete 2D motion control of unmanned aerial vehicles (UAVs) is considered, and motion policies are derived based on tabular Q learning. In [7], the motion control of a single autonomous flying base station is considered. Again, the motion is discrete and constrained in 2D space, and a Q learning approach is proposed, where the Q function is parameterized by neural networks. In [8], the authors examine the 3D motion of UAVs, where the motion is discrete. They consider the action space to be comprised by 7 actions and they propose tabular Q learning to derive the motion policies (storing the Q values in a table and updating them). Finally, in [9], the authors propose a model-free deep Q learning algorithm where the motion is discrete and restrained in 2D.

Many of the previously mentioned approaches are model-based, meaning that they make specific assumptions for the channel statistics and are sensitive to deviations of the channels from the assumed statistical structure. We focus on model-free methods that are generally more robust. Also, [2, 5] are on-policy meaning that they use data from the whole

Work supported in part by ARO grant W911NF2110071

Code can be found in: https://github.com/SpiliosEv/SoftActorCriticSIREN3D relay trajectory to update the policy. This introduces sample complexity that is prohibitive for real world deployment. We focus on off-policy RL, where only a subset of past experiences is kept and used for policy updates. All the previously mentioned approaches consider the motion to be discrete, meaning that the state and action spaces are discrete. When the area of motion is large enough or better performance is needed, the curse of dimensionality arises. The size of the state space grows significantly and finer discretization of the action space is required. This introduces huge complexity and memory needs and exploration in large state and action spaces is a challenging task.

We place the problem of relay motion control in a continuous RL framework and consider model-free, off-policy approaches. The state of the RL problem is a continuous 3D vector (or 2D vector for the 2D case) that corresponds to the relay position. The action is a continuous 3D vector (2D respectively) that belongs to a compact set and corresponds to the relay displacement vector. The reward is the relay's contribution to the SINR at the destination. We argue that the randomness of the channels in time and space requires the construction of stochastic policies rather than deterministic ones and verify this experimentally. In this direction we propose the adaptation of a continuous control off-policy stochastic actor-critic algorithm called soft actor-critic [10] that employs neural networks for function approximation. Off-policy actor-critic methods, in conjunction with the use of neural networks, even though are very sample efficient, they are notoriously unstable. In that spirit, we propose a modification of the soft actor-critic algorithm[10], by employing Sinusoidal Representation Networks (SIRENs) [11] for approximating the value function. This modification provides significant improvement both in performance and stability in comparison to the direct application of the soft actor-critic algorithm and the direct application of the Twin Delayed Deep Deterministic Policy Gradient (TD3) algorithm [12] for the 2D case. The algorithm that we propose maintains its performance for the 3D motion case without the need for added complexity or separate tuning. Both the soft actor-critic (stochastic policies) and the TD3 (deterministic policies) are the State-of-the-Art in off-policy model-free continuous control.

2. SIGNAL MODEL

We consider a source S, at position $\mathbf{p}_{\mathsf{S}} \in R^2$ and a destination D, located at $\mathbf{p}_{\mathsf{D}} \in R^2$. The Line-of-Sight communication is not feasible, so we employ R relays to facilitate it. The relays are deployed over a 3D (or 2D) space and can move continuously through that space. Time is divided in slots of equal duration (each slot denoted as t). Source transmits the signal $s(t) \in C$, with $E[|s(t)|^2] = 1$, using power $\sqrt{P_S} > 0$. The signal received at relay R_k, located at $\mathbf{p}_k(t)$, $k = 1, \dots, R$, is, after dropping for brevity the dependence of the relay position on t,

$$x_k(t) = \sqrt{P} f_r(\mathbf{p}_k, t) s(t) + n_k(t), \tag{1}$$

where f_k denotes the source-relay channel and $n_k(t)$ reception noise at the relay, assumed to be white with variance σ^2. Each relay multiplies $x_k(t)$, by weight $w_k(t) \in C$. All R relays transmit the weighted signal simultaneously. The signal received at D equals

$$y(t) = \sum_{k=1}^{R} g_k(\mathbf{p}_{\mathsf{D}}, t) w_k(t) x_k(t) + n_{\mathsf{D}}(t), \tag{2}$$

where g_k denotes the relay-destination channel and $n_{\mathsf{D}}(t)$ the reception noise, assumed to be white with variance σ_D^2. The problem of computing the relay weights that maximize the SINR at the destination, subject to a total relay transmission power budget, P_R, has a closed form solution [4], with the maximum achievable SINR equal to

$$\begin{aligned} V(t) &= \sum_{k=1}^{R} \frac{P_R P_S |f_k(\mathbf{p}_k, t)|^2 |g_k(\mathbf{p}_k, t)|^2}{P_S \sigma_D^2 |f_k(\mathbf{p}_k, t)|^2 + P_R \sigma^2 |g_k(\mathbf{p}_k, t)|^2 + \sigma^2 \sigma_D^2} \\ &= \sum_{k=1}^{R} V_I(\mathbf{p}_k, t). \end{aligned} \tag{3}$$

The SINR is the sum of individual relay contributions. So to maximize it, each relay should maximize its own V_I.

3. REINFORCEMENT LEARNING FOR CONTINUOUS RELAY MOTION CONTROL

RL[13] is concerned with settings where an agent interacts with an environment in discrete time steps. At time t, the agent is in state s_t and performs action a_t. At the next time step, the agent arrives in the next state s_{t+1} and collects scalar reward r_t. The goal of RL is to derive a mapping from states to actions, namely a policy, so that if the agent selects actions according to that mapping, the expected sum of rewards $E[\sum_{i=0}^{T} \gamma^i r_i]$ is maximized. Parameter $\gamma \leq 1$ is a discount factor quantifying how interested the agent is in long-term rewards.

In continuous control, the action a_t at every time step is continuous, meaning that $a_t \in A \equiv [-a, a]^d$, where A is a compact set. Actor-critic algorithms exhibit the best overall performance when it comes to model-free continuous RL tasks. In summary, the critic learns a value function $V(s_t)$ that estimates the expected sum of rewards from state s_t and the actor uses that estimate to update the policy $\pi(a_t|s_t)$ that provides the action a_t that the agent should take at state s_t to maximize the expected sum of rewards. Typically, neural networks are used to parameterize both actor and critic. One can distinguish between 2 types of actors: (i) *Deterministic actors*, that learn a deterministic mapping $\pi_{det}(a_t|s_t) : S \rightarrow A$, where S is the continuous set of the states and A is the continuous compact set of actions; and (ii) *Stochastic actors*, who learn a mapping $\pi_{stoch}(a_t|s_t) : S \rightarrow P_A$ from the set of states to a distribution over the action space. The chosen action is sampled from this estimated distribution.

Since the expression of the total SINR is distributed (3), we can have a single actor-critic model that estimates the policy which is shared by all the relays. The state s is the vector of coordinates that corresponds to the relay position, $s = [x, y, z]^T$ where $x \in [0, X_{max}]$, $y \in [0, Y_{max}]$ and $z \in [0, Z_{max}]$ ($s = [x, y]^T$ in the 2D case). The X_{max}, Y_{max}, Z_{max} are the maximum ranges in the x,y and z dimension respectively. The action $a = [dx, dy, dz]$ is a 3D vector that corresponds to the relay displacement, $a \in [-a_{max}, a_{max}]^3$, where a_{max} is chosen in conjunction with the relay speed and the duration of the time slot so as to make sure that during the maximum time for relay displacement, the channel does not change. Finally the reward is the relay's contribution (V_I) to the overall SINR. (3).

4. SOFT ACTOR-CRITIC

Here we focus on model-free off-policy actor-critic algorithms, in particular, the soft actor-critic[10] (stochastic policies) and the TD3[12] (deterministic policies). To the best of our knowledge, these algorithms have shown great promise in continuous control tasks. In our problem, the randomness of the channels in time and space (especially because of multipath [14]) introduces a stochastic reward. If a relay performs action a_t at state-position s_t at a time step t, it is probably going to receive different reward (V_I) than performing the same action at the same state at a different time step t'. One can infer that the stochasticity in the reward function entails that the best choice is to implement a stochastic actor-critic algorithm for control. We verify this in our experiments. So, we focus on the soft actor-critic algorithm.

The soft actor-critic is developed in the Maximum Entropy (ME) RL framework, meaning that the actor attempts to learn a stochastic policy (distribution over the action space) that maximizes the expected reward and, at the same time, exhibits maximum entropy. ME enforces exploration for avoiding suboptimal local minima. The value function $V_\psi(s)$ and the Q function $Q_\theta(s, a)$ are parameterized by neural networks. The value function estimates the sum of rewards from state s and the Q function estimates the sum of rewards from state s when performing action a. The policy $\pi_\phi(a|s)$ is Gaussian over the action space with mean and covariance estimated by a neural network. The experiences ($\{s_t, a_t, s_{t+1}, r_t\}$) from an agent's trajectory are stored in a memory called Experience Replay (ER) (off-policy). For every network weight update step we uniformly sample a batch of experiences from the ER and update the networks' weights with gradient descent using the loss functions

$$J_V(\psi) = E_{s_t \sim ER}[\frac{1}{2}(V_\psi(s_t) - E_{a_t \sim \pi_\phi}[Q_\theta(s_t, a_t) - log\pi_\phi(a_t|s_t)])^2] \quad (4)$$

$$J_Q(\theta) = E_{(s_t,a_t) \sim ER}[\frac{1}{2}(Q_\theta(s_t, a_t) - r_t - \gamma V_{\bar{\psi}}(s_{t+1}))^2] \quad (5)$$

$$J_\pi(\phi) = E_{s_t \sim ER}[D_{KL}(\pi(\cdot|s_t)||\frac{\exp Q_\theta(s_t, \cdot)}{\zeta(s_t)})] \quad (6)$$

D_{KL} is the KL divergence. The quantity $\zeta(s_t)$ is for normalization but contributes nothing to the gradient so we discard it in the implementation. The actions are sampled with the reparameterization trick from the estimated Gaussian policy to allow weight updates with backpropagation. $V_{\bar{\psi}}$ is a target network used to stabilize training [15]. $\bar{\psi}$ is an exponential moving average of ψ. Two Q networks ($Q_{\theta_1}, Q_{\theta_2}$) are trained independently to minimize $J_Q(\theta)$ to mitigate overestimation [12]. The minimum of the two is used for the update of the value network at every step (4)

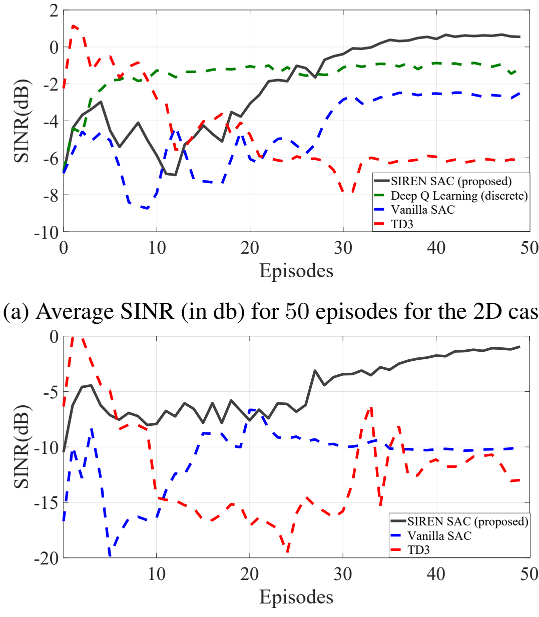
5. SOFT ACTOR-CRITIC WITH SIREN

The natural way to directly adapt the soft actor-critic for relay control is to use Multilayer Perceptrons with ReLU activations between layers (ReLU MLPs) for all the neural networks that parameterize the respective functions. This variation we call Vanilla Soft Actor-Critic (Vanilla SAC). Recent results in using deep neural networks for function approximation reveal that MLPs, performing low-dimensional regression tasks with coordinate inputs, fail to converge for the high frequency components of the target signal [16, 17], a phenomenon called *Spectral Bias*. Additionally, the authors in [12] argue convincingly that the main source of instability for off-policy actor-critic algorithms is the coupling between value and policy networks. When the value estimates are inaccurate, the policy is updated on these bad estimates and is poor. The interplay between policy and value amplifies this divergent behavior when the value estimates are inaccurate.

In the case of Vanilla SAC for relay control, the value network ($V_\psi(s_t)$) is an MLP that implicitly performs low dimensional regression from coordinate inputs(state). It is reasonable to infer that, due to the variability of the channels with respect to time and space, the underlying value function possesses high frequencies and the MLP cannot learn them. In that sense, the value estimates are prone to be inaccurate and the algorithm unstable. This is verified by our experiments where the Vanilla SAC performs very differently across seeds. The authors in [11] introduce a new class of neural network architectures called Sinusoidal Representation Networks (SIRENs) to tackle the spectral bias. The SIREN converges reliably for the high frequencies in the target signal and in fact, early during training. The architecture is comprised by dense layers, but the activations between layers are sinusoids. A weight initialization scheme is devised to preserve the distribution of activations through the network during training. SIRENs outperform ReLU MLPs for a range of regression tasks. We propose the use of SIRENs for the value ($V_\psi(s_t)$) and the target value ($V_{\bar{\psi}}(s_t)$) networks. We call this variation SIREN SAC.

6. EXPERIMENTS

The relay motion can be learned only when the channels are correlated in time and space. To test the proposed method we first need to simulate an environment with such channels; for this purpose we simulate the channels so that they have statistics as described in [2]. In particular, the log-magnitude of the channel has 3 additive components, the pathloss with exponent $l = 2.3$, the multipath, which is i.i.d zero-mean Gaussian with variance $\sigma_\xi = 0.6$, and the shadowing which is a a zero-mean Gaussian correlated in time and space. The correlation distance is $c_1 = 1.2$, the correlation time $c_2 = 0.6$, and the shadowing power is $\eta^2 = 4$. The source and destination channels exhibit correlation with distance $c_3 = 0.5$. The source transmission power is $P_S = 57dbm$ and the transmission power of the relays, $P_R = 55dbm$. The variance of the reception noise at the relays and the destination is $\sigma_D^2 = \sigma^2 = 0.5$. These parameters are consistent with real time measurements[18]. We perform our simulations for a 20^3 cube (20^2 square for the 2D case), 3 relays and 1 source-destination pair. In our simulations we compare 3 algorithms, namely, Vanilla SAC (soft actor-critic with ReLU MLPs for all networks), SIREN SAC(soft actor-critic with SIRENs for the value networks), which is what we propose and TD3 (Twin Delayed Deep Deterministic Policy Gradient with ReLU MLPs for all networks) . For the 2D motion, we also compare the continuous algorithms with an off-policy model-free deep Q learning algorithm proposed for discrete relay motion in [9]. Every network (ReLU MLP or SIREN) is comprised by 3 layers (200 neurons each). The Adam optimizer [19] is used for updating the weights with a learning rate of 2e-4 and batch size of 100. The Experience Replay size is 1e+6. The simulations are for 50 episodes (400 time slots per episode). The relays respect the boundaries. We enforce this by clipping the action if the next state violates the boundaries. We also clip the actions if the next states correspond to collision (the distance between any 2 relays is smaller than $1m$). Finally, the maximum range of the components of the action, a_{max}, is $\frac{1}{\sqrt{3}}$ for the 3D case ($\frac{1}{\sqrt{2}}$ for the 2D case respectively). The same configuration for every continuous algorithm (number of parameters, hyperparameter values) is used for both scenarios (2D and 3D). For the 2D scenario, to make fair comparison between the discrete control algorithm (deep Q learning) and the continuous control methods, to evaluate the discrete method, we discretize the 2D area ($1m \times 1m$ grid cells) so as to make sure that the maximum relay displacement at every time slot is approximately equal for all methods. The performance for each algorithm is an average over 12 seeds (Fig. 1).



(a) Average SINR (in db) for 50 episodes for the 2D case

(b) Average SINR (in db) for 50 episodes for the 3D case

Fig. 1: Experiments for 2D and 3D relay motion control.

The TD3 algorithm performs poorly for both scenarios. This is indication that stochastic policies are more suitable for continuous relay motion control in comparison to deterministic ones. We attribute this to the channels' randomness inducing a stochastic reward. The employment of SIRENs provides significant improvement both in reward and in robustness. The SIREN SAC outperforms all the other continuous control algorithms and also achieves higher rewards in comparison to the deep Q learning approach (discrete), in the 2D case. The SIREN SAC also retains its 2D performance for the 3D scenario as well, without the need for extra parameters or separate tuning. The value network converges early for the high frequencies of the underlying value function. So, it provides high quality estimates for the updates of the Q networks and the policy. We should note that, even though the SIREN SAC achieves higher reward policies than the deep Q learning method (discrete) in the mean sense, the deep Q learning approach exhibits less variance between seeds.

Although it would be a reasonable idea to employ SIRENs for the Q networks as well, the Q networks' input is the concatenation of state and action. The range of values for the components of the state is different than the range of the action components. This causes instability that cannot be alleviated by normalization. Employing SIRENs for TD3 provides no improvement.

7. CONCLUSIONS

We pose the problem of relay motion control in a continuous model-free set up. We focus on off-policy deep actor-critic methods to keep the sample complexity low, which is critical for real world deployment. We provide intuition on why stochastic policies are more suitable than deterministic policies for the problem and verify this with experiments. Finally we propose the use of SIRENs for approximating the value function, which provides significant improvement in performance and robustness. The proposed variation retains the performance of the 2D motion scenario on the 3D scenario without need for additional complexity or retuning.

References

[1] P. Kumari et al. “IEEE 802.11ad-Based Radar: An Approach to Joint Vehicular Communication-Radar System”. In: *IEEE Transactions on Vehicular Technology* 67.4 (2018), pp. 3012–3027. DOI: 10.1109/TVT.2017.2774762.

[2] D. S. Kalogerias and A. P. Petropulu. “Spatially controlled relay beamforming”. In: *IEEE Transactions on Signal Processing* 66.24 (2018), pp. 6418–6433.

[3] J. Li, A. P. Petropulu, and H. V. Poor. “Cooperative transmission for relay networks based on second-order statistics of channel state information”. In: *IEEE Transactions on Signal Processing* 59.3 (2010), pp. 1280–1291.

[4] V. Havary-Nassab et al. “Distributed beamforming for relay networks based on second-order statistics of the channel state information”. In: *IEEE Transactions on Signal Processing* 56.9 (2008), pp. 4306–4316.

[5] K. Diamantaras and A. Petropulu. “Optimal Mobile relay beamforming via Reinforcement Learning”. In: *International Workshop on Machine Learning for Signal Processing (MLSP) 2019* (2019).

[6] Y. Huang et al. “Reinforcement Learning for Maneuver Design in UAV-Enabled NOMA System with Segmented Channel”. In: *arXiv preprint arXiv:1908.03984* (2019).

[7] H. Bayerlein, P. De Kerret, and D. Gesbert. “Trajectory optimization for autonomous flying base station via reinforcement learning”. In: *2018 IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*. IEEE. 2018, pp. 1–5.

[8] X. Liu, Y. Liu, and Y. Chen. “Reinforcement learning in multiple-UAV networks: Deployment and movement design”. In: *IEEE Transactions on Vehicular Technology* 68.8 (2019), pp. 8036–8049.

[9] S. Evmorfos, K. Diamantaras, and A. Petropulu. “Reinforcement Learning for Motion Policies in Mobile Relaying Networks”. In: *IEEE Transactions on Signal Processing* (2022), pp. 1–1. DOI: 10.1109/TSP.2022.3141305.

[10] T. Haarnoja et al. “Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor”. In: *International conference on machine learning*. PMLR. 2018, pp. 1861–1870.

[11] V. Sitzmann et al. “Implicit neural representations with periodic activation functions”. In: *Advances in Neural Information Processing Systems* 33 (2020).

[12] S. Fujimoto, H. Hoof, and D. Meger. “Addressing function approximation error in actor-critic methods”. In: *International Conference on Machine Learning*. PMLR. 2018, pp. 1587–1596.

[13] R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[14] M. C. Vanderveen, A.-J. Van der Veen, and A. Paulraj. “Estimation of multipath parameters in wireless communications”. In: *IEEE Transactions on Signal Processing* 46.3 (1998), pp. 682–690.

[15] V. Mnih et al. “Playing Atari with deep reinforcement learning”. In: *arXiv preprint arXiv:1312.5602* (2013).

[16] N. Rahaman et al. “On the spectral bias of neural networks”. In: *International Conference on Machine Learning*. PMLR. 2019, pp. 5301–5310.

[17] M. Tancik et al. “Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains”. In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., 2020, pp. 7537–7547.

[18] R. Wang et al. “Stationarity region of mm-wave channel based on outdoor microcellular measurements at 28 GHz”. In: *MILCOM 2017-2017 IEEE Military Communications Conference (MILCOM)*. IEEE. 2017, pp. 782–787.

[19] D. P. Kingma and J. Ba. “Adam: A method for stochastic optimization”. In: *arXiv preprint arXiv:1412.6980* (2014).