2018 Annual American Control Conference (ACC)
June 27–29, 2018. Wisconsin Center, Milwaukee, USA

Learning-based Approaches for Controlling Neural Spiking

Sensen Liu[1], Noah M. Sock[3] and ShiNung Ching[1,2]

***Abstract*— We consider the problem of controlling populations of interconnected neurons using extrinsic stimulation. Such a problem, which is relevant to applications in both basic neuroscience as well as brain medicine, is challenging due to the nonlinearity of neuronal dynamics and the highly unpredictable structure of underlying neuronal networks. Compounding this difficulty is the fact that most neurostimulation technologies offer a single degree of freedom to actuate tens to hundreds of interconnected neurons. To meet these challenges, here we consider an adaptive, learning-based approach to controlling neural spike trains. Rather than explicitly modeling neural dynamics and designing optimal controls, we instead synthesize a so-called control network (CONET) that interacts with the spiking network by maximizing the Shannon mutual information between it and the realized spiking outputs. Thus, the CONET learns a representation of the spiking network that subsequently allows it to learn suitable control signals through a reinforcement-type mechanism. We demonstrate feasibility of the approach by controlling networks of stochastic spiking neurons, wherein desired patterns are induced for neuron-to-actuator ratios in excess of 10 to 1.**

I. INTRODUCTION

Networks in the brain are composed of neurons that propagate information through impulsive electrical signals known as action potentials, or 'spikes' [1]. Understanding the precise mechanisms of how spiking dynamics mediate information processing is a fundamental neuroscience question. One approach to studying this question is to use extrinsic 'neurocontrol' [2] to stimulate populations of neurons *in vivo*, so as to observe consequent changes in animal behavior. In this context, stimulation can be understood as an experimentally delivered input (e.g., an electrical field, or optical illumination) that excites the actuated region of the brain. Given the prevalence of such technologies in both clinical and basic scientific domains, there is interest in using neurostimulation technologies [3], [4] to induce spiking patterns in neural populations and networks.

In this vein, there has been as desire for theoretical and engineering schema that address the neurocontrol problem [3]. These approaches generally follow optimal control frameworks towards objectives such as desynchronizing a neural population (e.g., [5], [6] and the references therein), or selectively firing specific neurons within a population in ordered sequences [3], [7] or in a time-optimal fashion [8]. Other approaches have taken a probabilistic view of the neurocontrol problem, focusing on manipulating the *likelihood* of neural spiking [9], [10], subject to input constraints.

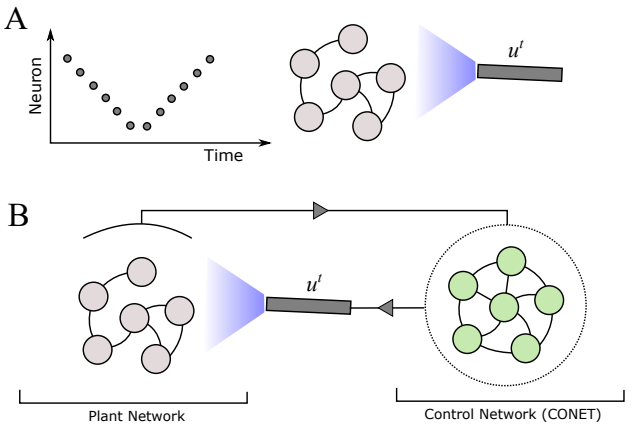

Fig. 1. Control network concept. (A) The spike-selective neurocontrol problem seeks to design stimulation inputs u^t that can produce spatiotemporally specific patterns of spiking activity. Here, stimulation can be understood as an extrinsic input that actuates a network of cells. (B) The approach explored in this paper involves the design of a recurrent control network (CONET) that interfaces with the (generally unknown) plant network and learns to control it.

The above approaches are useful insofar as they enable basic and important insights into fundamental limitations associated with neurocontrol. For example, in [8] it is shown that heterogeneity in the dynamics of neurons is essential for enabling temporally precise spiking objectives. However, from a practical perspective, these approaches suffer from needing a well-parametrized model (either a dynamical systems-based or statistical) of the network that is being controlled. This presents a major analytical challenge, since the dynamics within neuronal networks are usually highly non-linear and stochastic. Thus, performing formal control analysis and design on systems larger than a few neurons rapidly becomes intractable.

Compounding this difficulty is the fact that for many technologies, the degrees of actuator freedom are quite restricted (e.g., a single actuator that can deliver only piecewise constant inputs). In other words, individual neurons do not receive independent inputs, but rather are simultaneously controlled through a single stimulating device (see Figure 1 for schematic).

In this work we attempt to obviate some of these challenges through a non-classical, model-free control design

S. Ching holds a Career Award at the Scientific Interface from the Burroughs-Wellcome Fund. This work was partially supported by AFOSR 15RT0189, NSF ECCS 1509342, NSF CMMI 1537015 and NSF CMMI 1653589, from the US Air Force Office of Scientific Research and the US National Science Foundation, respectively.

[1] S. Liu and S. Ching are with the Department of Electrical and Systems Engineering, Washington University in St. Louis, St. Louis, MO, USA shinung at ese.wustl.edu

[2] S. Ching is with the Department of Biomedical Engineering and the Division of Biology and Biomedical Sciences, Washington University in St. Louis, St. Louis, MO, USA

[3] N. Sock is presently at the California Institute of Technology, Pasadena, CA, USA

approach. Specifically, here we consider the problem of inducing neural spike trains by means of learning, wherein the 'controller' is itself a network of simulated neurons. This *control network*, or CONET, resides beside the target spiking network and *learns* to control it without a prior dynamical model. This approach is appealing at a conceptual level, since it conforms quite directly to the internal model principle of control [11]. If successful, our controller would mirror the system being controlled(i.e, a network controlling a network).

From a technical perspective, our approach can be viewed as a model-free control design using an artificial neural network. Such a framework has a long history of success in a variety of control applications (see, e.g., [12]). However, unlike conventional neural networks, the CONET has a fully recurrent connectivity and is probabilistic in its output. Our principal objective is to find a learning rule for the (recurrent) connection weights so that the desired control objective is met. To do so, we build on our recent developments in network-based information maximization [13], wherein we developed a pairwise learning rule that allows a recurrent network to retain information about its inputs over time. In this paper, we exploit this framework for the purposes of control by: (i) tailoring the architecture of the CONET so that it maximizes the information between its activity and that of the plant network, and (ii) endowing the overall learning rule with a reinforcement mechanism, towards enabling the CONET to issue control signals that realize tracking of the desired spike pattern. It turns out that this overall framework is remarkably effective in generating controls that can induce nontrivial spiking patterns.

The remainder of this paper is organized as follows. In section II we formalize the control problem we consider and introduce the model used for our study. Section III provides the main technical results, and we show several examples illustrating the efficacy of the CONET. Section IV concludes the paper.

II. Formulation and Preliminary Results

A. 'Plant' Spiking Network

Our goal is to induce prescribed spiking patterns in a network of spiking neurons by means of extrinsic stimulation. For clarity, and in concordance with control-theoretic parlance, we will heretofore refer to this controlled network as the 'plant' network. For simplicity, we will model the plant network in discrete time wherein the i^{th} neuron is characterized by a variable $\bar{x}_i^t = \{0,1\}$ at time $t \in \mathbb{N}^+$. Neurons are linked by synaptic coupling weights $\bar{w}_{ij}$. The variable $\bar{x}_i^t$ is obtained as $P(\bar{x}_i^t = 1) = g_\beta(\bar{v}_i^t)$, where $\bar{v}_i^t$ is an underlying state variable that aggregates input from pre-synpatic neurons via

$$\bar{v}_i^t = \sum_{j=1}^{N_P} \bar{w}_{ij}\bar{x}_j^{t-1} + u^{t-1}, \tag{1}$$

and u^{t-1} is the control input, N_P represents the number of neurons in the plant network and $g_\beta(\cdot)$ is a sigmoidal function:

$$g_\beta(\bar{v}_i(t-1)) = \frac{1}{1+\exp(\theta)}, \text{ where } \theta = -2\beta(\bar{v}_i(t-1)). \tag{2}$$

Thus, at a given time, each neuron is either spiking ($x^t = 1$) or silent ($x^t = 0$), governed by a time-varying Bernoulli process. Importantly, the entire network receives a *single* input u^t, which mimics the scenario described in the Introduction wherein actuation is common to many neurons in a population.

B. CONET Description

The control network (CONET) is modeled in a similar fashion to the plant network. Here, the i^{th} neuron is specified in terms of $x_i^t = \{0,1\}$ at time t, and is obtained as $P(x_i^t = 1) = g_\beta(v_i^t)$, where v_i^t is

$$v_i^t = \sum_{j=1}^{N} w_{ij}x_j^{t-1} + I_i^{t-1}, \tag{3}$$

and I_i^{t-1} is an extrinsic input. Since the CONET is entirely simulated, this input can be indexed by i. N represents the number of neurons in the CONET and $g_\beta(\cdot)$ is a sigmoidal function, similar to (2).

It is critical to note the conceptual point that the plant network and the CONET are distinct entities. The former is the object being controlled, while the latter is the object generating control signals. Several points are worth emphasizing here:

- The CONET is fully recurrent, since any neuron can be connected to any other.
- The CONET *does not* assume any knowledge of the plant network. In fact, the plant network does not need to modeled as in (1). Most notably, the number of neurons in the plant network can be different from the number of neurons used in the CONET (i.e., $N_P \neq N$), though for reasons that will soon be clear, we will generally assume that $N \geq N_P$.
- The (recurrent) connections of the CONET (w_{ij}) do not, *a priori*, have any relationship with the with connections of the plant network (which are assumed unknown to the CONET).

C. CONET Design

The CONET interacts with the plant network in two ways:

a) Spike feedback: We assume that the CONET can observe the spiking activity of the plant network and use it as a feedback signal, via

$$[I_1(t), ..., I_N(t)] = h(\bar{\mathbf{x}}^t), \tag{4}$$

where $\bar{\mathbf{x}}^t = [\bar{x}_1^t, ..., \bar{x}_{N_P}^t]$ and $h(\cdot) \in \mathbb{R}^{N \times N_P}$ is a feedback function.

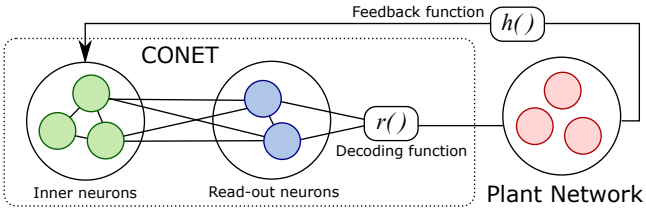

Fig. 2. Schematic of CONET design structure.

b) Control signal read-out (decoder): The CONET generates a control signal that will be fed back to the plant network. To simplify this design, we will implement two dedicated subsets of neurons: one that receive spike feedback from the plant network and another that provides the control signal readout. We will assume that the $N - N_P$ read-out neurons in the CONET generate a control signal that is decoded from the read-out neurons via

$$u(t) = r(\mathbf{x}_{out}^t), \tag{5}$$

where $\mathbf{x}_{out}^t = [x_{N_P+1}^t, ..., x_N^t]$ and $r(\cdot) \in \mathbb{R}^{N-N_P}$ is a decoding function. This decoding structure presumes that the CONET has more neurons than the plant network, as schematically depicted in Figure 2. Our specific design of the decoding function will be presented in Section III.

The remaining inner neurons in the CONET will enable learning of the intended control objective.

The overall question is thus: How should the connectivity weights of the CONET be set or adaptively learned in order to enable the CONET to issue effective control signals? In particular, we would like $u(t)$ to produce desired patterns of spiking in the plant network.

Towards this objective we will exploit the idea of information maximization, wherein the CONET will try and maximize information between the target and achieved spike patterns. An important preliminary result towards this goal is provided below.

D. Learning for Maximization of Mutual Information

Denoting the state vector of the whole CONET as $\mathbf{x}^t \in \mathbb{R}^N$, and the history of the network from t_1 to t_2 by $X_{t_1}^{t_2} \in \mathbb{R}^{N\times(t_2-t_1)}$, we consider the basic problem of maximizing the cumulative information retention within the CONET over time. Mathematically, this can be written as:

$$\max_{\mathbf{w}} MI(\mathbf{x}^t; X_1^{t-1}), \tag{6}$$

where $\mathbf{w}$ denotes the network connectivity weights. And

$$MI(\mathbf{x}^t; X_1^{t-1}) = \sum_{\mathbf{x}^t, X_1^{t-1}} P(\mathbf{x}^t, X_1^{t-1}) \log \frac{P(\mathbf{x}^t|X_1^{t-1})}{P(\mathbf{x}^t)}, \tag{7}$$

that is, the Shannon mutual information between the current state of the network and its history (X_1^{t-1}) over the horizon $t-1$ time steps.

The underlying idea behind this maximization is that since the CONET will receive feedback from the plant network, such optimization might allow it to learn a latent representation of the plant network dynamics that can then enable control.

In our prior work [13], we used a typical gradient approach to derive a learning rule that solves (6), understanding that the non-convexity of the objective means that global solutions are not assured. The derived rule can be written in the form of

$$\Delta w_{ij}^t = \gamma \mathbf{E}_{X^t}\left[\phi_{ij}^h(t) + \phi_{ij}^a(t)\right], \tag{8}$$

where γ is the gradient-based learning rate. Borrowing terminology from neuroscience, $\phi_{ij}^h(t)$ is known as a Hebbian modification function [14] since it promotes co-activation of neurons, while $\phi_{ij}^a(t)$ is anti-Hebbian. Each of them is composed of two components as follows:

$$\begin{aligned}\phi_{ij}^h(t) &= \left(\phi^h(p_i^t, x_i^t, x_j^{t-1}) + \phi^h(p_i^t, 1-x_i^t, x_j^{t-1})\right),\\ \phi_{ij}^a(t) &= \left(\phi^a(p_i^t, x_i^t, x_j^{t-1}) + \phi^a(p_i^t, 1-x_i^t, x_j^{t-1})\right),\end{aligned} \tag{9}$$

where $p_i^t = P(x_i^t = 1)$. More specifically, the anti-Hebbian $\phi_{ij}^a(t)$ has:

$$\begin{aligned}\phi^a(p_i^t, x_i^t, x_j^{t-1}) &= 2\beta\left(\frac{\mathbf{E}[(p_i^t)^2] - \mathbf{E}[p_i^t]}{\mathbf{E}[p_i^t]}\right) x_j^{t-1}x_i^t + \\ &\quad 2\beta(1-p_i^t)x_j^{t-1}x_i^t;\\ \phi^a(p_i^t, 1-x_i^t, x_j^{t-1}) &= 2\beta\left(-\frac{\mathbf{E}[(p_i^t)^2] - \mathbf{E}[p_i^t]}{1-\mathbf{E}[p_i^t]}\right)\times \\ &\quad x_j^{t-1}(1-x_i^t) + 2\beta(-p_i^t)x_j^{t-1}(1-x_i^t).\end{aligned} \tag{10}$$

Similarly, the Hebbian part $\phi_{ij}^h(t)$ consists of:

$$\begin{aligned}\phi^h(p_i^t, x_i^t, x_j^{t-1}) &= 2\beta\left((1-p_i^t)\log\frac{p_i^t}{\mathbf{E}[p_i^t]}\right)x_j^{t-1}x_i^t;\\ \phi^h(p_i^t, 1-x_i^t, x_j^{t-1}) &= 2\beta\left(-p_i^t\log(\frac{1-p_i^t}{1-\mathbf{E}[p_i^t]})\right)x_j^{t-1}(1-x_i^t).\end{aligned} \tag{11}$$

The information-optimal learning rule (8) thus promotes either correlation or de-correlation between neurons, through alternations between Hebbian and anti-Hebbian variables (9). Intuitively, the Hebbian component strengthens connections during correlated firing and thus helps 'memorization'. Oppositely, the anti-Hebbian term promotes forgetting or correction through connection weakening.

The derived information-optimal plasticity rule (8) contains 'global' variables, such that each variable in (10) and (11) require knowledge of all other neurons of the entire network (since the expectations are taken with respect the joint distribution of the entire network). Thus, this form of learning is computationally arduous and does not scale gracefully.

However, in [13], by assuming ergodicity in the recurrent activity, we developed a local, nested recursive estimator for the expectations of $\phi_{ij}^h(t)$ and $\phi_{ij}^a(t)$, expressed as $\bar{\phi}_{ij}^h(t)$ and $\bar{\phi}_{ij}^a(t)$ such that

$$\Delta w_{ij}^t = \gamma \left[\bar{\phi}_{ij}^h(\mathbf{s}(t)) + \bar{\phi_{ij}}^a(\mathbf{s}(t)) \right], \tag{12}$$

where $\mathbf{s}(t) = (s_{1,i}^t, s_{2,i}^t, s_{3,i}^t, s_{4,ij}^t, s_{5,ij}^t, s_{6,ij}^t)$ and

$$\begin{aligned} \bar{\phi}_{ij}^h(t) &= (\frac{s_{2,i}^t}{s_{1,i}^t}) s_{4,ij}^t - (\frac{s_{2,i}^t - s_{1,i}^t}{1 - s_{1,i}^t}) s_{4,ij}^t, \\ \bar{\phi_{ij}}^a(t) &= s_{5,ij}^t + + s_{6,ij}^t. \end{aligned} \tag{13}$$

These surrogate state variables evolve according to:

$$\begin{aligned} \tau_1 \Delta s_{1,i}^t &= -s_{1,i}^{t-1} + g(v_i^t); \\ \tau_2 \Delta s_{2,i}^t &= -s_{2,i}^{t-1} + (g(v_i^t))^2; \\ \tau_3 \Delta s_{3,i}^t &= -s_{3,i}^{t-1} + g(v_i^t) x_i^t; \\ \tau_4 \Delta s_{4,ij}^t &= -s_{4,ij}^{t-1} + s_{3,i}^t x_j^{t-1}; \\ \tau_5 \Delta s_{5,ij}^t &= -s_{5,ij}^{t-1} + (-g(v_i^t)) s_{3,i}^t x_j^{t-1} + \\ &\quad (1 - g(v_i^t)) \log \left(\frac{g(v_i^t)}{s_{1,i}^t} \right) s_{3,i}^t x_j^{t-1}; \\ \tau_6 \Delta s_{6,ij}^t &= -s_{6,ij}^{t-1} + (-g(v_i^t))(1 - s_{3,i}^t) x_j^{t-1} + \\ &\quad (-g(v_i^t)) \log \left(\frac{1 - g(v_i^t)}{1 - s_{1,i}^t} \right) (1 - s_{3,i}^t) x_j^{t-1}. \end{aligned} \tag{14}$$

Here, $s_{1,i}^t$, $s_{2,i}^t$, $s_{3,i}^t$ are variables that depend on the state of the post-synaptic neuron (i.e., the neuron on the end of the connection), while $s_{4,ij}^t$, $s_{5,ij}^t$, $s_{6,ij}^t$ are variables that depend on the pairwise activity of the pre- and post-synaptic neurons. Each of these variables can be understood in terms of performing a particular step toward the overall estimation of the joint distribution, through purely pairwise operations. In essence, (12)-(14) can be viewed as a scheme to perform recurrent information optimization in a computationally efficient, distributed fashion.

III. RESULTS

We proceed with our CONET design in two steps. First, we begin by showing the ability of the CONET to learn the latent structure of the plant network, consistent with the internal model principle. We then extent the capability of the CONET by incorporating a reinforcement mechanism that naturally fits with the learning dynamics in (12)-(14).

A. Latent Structure Inference of 'Plant' Network

In this section, we demonstrate the CONET capability of inferring the plant network dynamics based on the observation of output spike trains. As an example, we construct the plant as a network of 20 interconnected neurons with each neuron modeled as a spiking unit based on equations (1) and (2). For neurophysiological consistency, the network consists of dedicated inhibitory neurons and excitatory neurons and an approximately balanced ratio of excitatory-inhibitory connection weights [15]–[17]. In particular, there are fewer inhibitory neurons than the excitatory, but the inhibitory links are on average stronger than the excitatory links. Here, the plant network has 6 inhibitory neurons and the rest are excitatory. In the network, we can distinguish these two types of neuron populations according to the connections emanating from them: inhibitory neurons emanate only negative links, while excitatory neurons connect to other neurons through only positive links (Figure 3A).

For ease of illustration, all the inhibitory connections are of strength -2, and all the positive connections are 1. Neurons are not connected if the link between them is 0. Note that neurons do not produce self-excitation or self-inhibition.

We construct the CONET to be a 20-neuron network, where each neuron reads the spiking activity directly from one of the neurons in the plant network via the feedback function

$$I_i^t = \alpha \bar{x}_i^t - \alpha/2, \tag{15}$$

where $\alpha = 4$. This feedback function scales the amplitude of the binary $\bar{x}_i^t$ so as to be commensurate with the neuronal input $\sum_{j=1}^N w_{ij} x_j^{t-1}$ in (3).

With this feedback function we expect that, when endowed with the learning rule (12)-(14), the CONET is able to learn the latent structure from the plant network based on observation of the spikes. Indeed, this is precisely what occurs. The connectivity of the CONET after learning is illustrated in Figure 3B. To emphasize the point, we thresholded the learned CONET weights, such that all the positive links larger than 0.1 are rounded to 1 and all the negative links -0.1 are set to -2 (Figure 3C).

B. CONET learning algorithm

In Section II, we derived a learning rule for recurrent information optimization *within* the spiking network activity (i.e., (6)). We have shown that the learning rule is capable of correlating and decorrelating actions through the Hebbian and anti-Hebbian learning components, respectively, when either is favorable for optimality.

However, the derived learning rule does not yet consider the optimization with respect to the a prescribed control objective, or this case a desired spike patterns. Therefore, expanding on (8), we introduce an augmented learning rule by employing a reinforcement mechanism such that can modulate the alternations between the Hebbian and anti-Hebbian regimes based on an external objective function.

More specifically, the alternation is guided through two real reinforcement coefficients c_h^t, c_a^t, such that the new learning rule is:

$$\Delta w_{ij}^t = \gamma [c_h^t \bar{\phi_{ij}}^h(t) + c_a^t \bar{\phi_{ij}}^a(t)]. \tag{16}$$

The coefficients c_h^t, c_a^t adjust the weights of Hebbian and anti-Hebbian learning components in the synaptic learning rule according to:

$$\begin{aligned} \tau_h \Delta c_h^t &= -c_h^t + (\tau_h - 1) Q^t; \\ \tau_a \Delta c_a^t &= -c_a^t - (\tau_a - 1) Q^t. \end{aligned} \tag{17}$$

Here, Q^t is our reward function at time t, which is calculated based on the ℓ_2 distance between the network output $\bar{\mathbf{x}}^t$ and the desired pattern at time t:

$$Q^t = \sqrt{N_P}/2 - \| \sqrt{\bar{\mathbf{x}}^t - \mathbf{y}^t} \|, \tag{18}$$

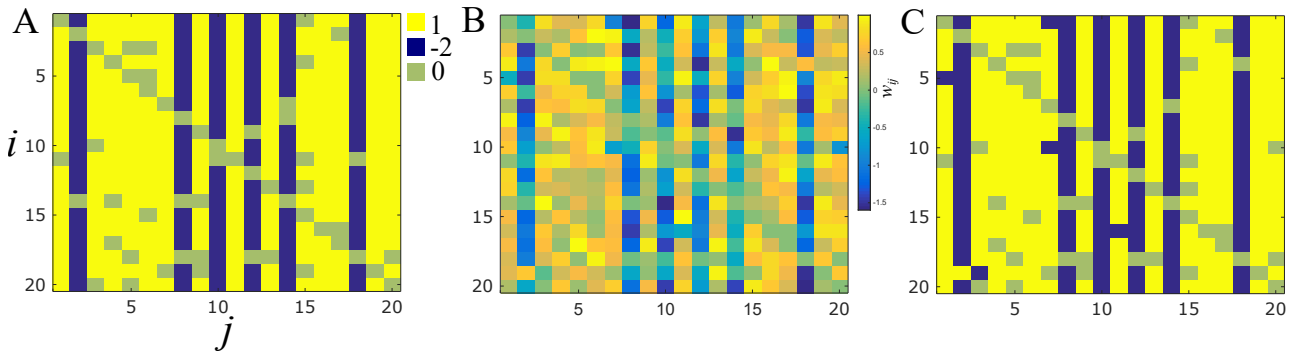

Fig. 3. **The CONET learns the latent structure of the plant network from the plant spiking activity.**(A) Actual connectivity of the plant network. (B) The CONET connectivity matrix after learning and (C) thresholding.

where $\bar{\mathbf{x}}^t$ represents the (observed) plant network activity and $\mathbf{y}^t$ is the desired spiking pattern at time t. The reward Q^t takes value over a range of $\left[-\sqrt{N_P}/2, \sqrt{N_P}/2\right]$. This form of reward function allows for small distance between $\bar{\mathbf{x}}^t$ and $\mathbf{y}^t$ to generate positive reward, whereas large distance leads to negative reward..

We note that the reinforcement mechanism in (17) differs from the conventional reinforcement learning algorithms that usually address the reward optimization problem directly [18]–[20], e.g., via explicit gradient ascent on the objective function. Here, we do not have a closed form solution of the reward function Q^t in terms of the network dynamics (recall that $\bar{x}^t$ comes from the plant network, whose dynamics are opaque to the CONET). Thus, optimization methods that rely on functional manipulation of Q^t [21], [22] are difficult to apply in this setting. Instead, here we approach our problem by incorporating the reward and its history within the derived synaptic learning (16) through its dynamics (17).

In particular, (17) implements a basic filtering operation on the reward Q^t, which in turns modulates either the strengthening of weakening of connections within the Hebbian/anti-Hebbian learning framework. The time constants τ_h, τ_a in (17) represent a forgetting factor in the low-pass-filtering dynamics in (17). They imply how much immediate reward Q^t is preserved in determining the alternations between Hebbian and anti-Hebbian regimes. Larger time constants lead to actions that benefit more immediate outcome.

C. Control in a population of neurons

The addition of the reinforcement dynamics of (16) - (18) enables the CONET to learn to manipulate the plant network spike trains to a target pattern. To demonstrate the capacity of the new learning rule, here we consider an example of controlling an $11-$neuron network to produce an hourglass-shaped pattern (Figure 4A) via a single control input. In this case, we augmented the CONET used in the previous inference example with an additional output layer that provides control signal readout (i.e., consistent with Figure 2). We define the decoding function of (5) as:

$$u(t) = r(\mathbf{x}_{out}^t) = \sum_{k=N_P+1}^{N} x_k^t, \tag{19}$$

recalling that without loss of generality, the last $(N - N_P)$ neurons of the CONET are designated as the readout layer. The readout $u(t)$, by aggregating the spikes from the output layer, reflects the instantaneous firing intensity of the output population in the CONET. Here, we emphasize that neurons in both layers are all recurrently connected such that the CONET is a recurrent network as a whole. The connections in CONET adapt according to the learning rule given in (16) - (18) during the learning process.

For the above scenario, we simulated learning by repeatedly presenting the pattern to the CONET 128 times. We observed that the CONET converges at around $t = 300$ steps, reflected by the average reward in Figure 5. Since the initial condition of the CONET is randomized, the initial reward is usually negative. From Figure 5, we see that the reward increases rapidly from this initial negative value to positive, and then grows until it converges to around $Q^* = 0.6$. We note that the theoretical maximal level of the reward, based on (18), is $Q_{max} = \sqrt{N_P}/2 \approx 1.66$ (for a plant network with $N_P = 11$). Although it may appear to us that the maximal reward in simulations Q^* is far less than the theoretical Q_{max}, the result indicates that the error between $\bar{\mathbf{x}}^t$ and $\mathbf{y}^t$ after training is in fact, on average, one spike per time step, since:

$$\|\sqrt{\bar{\mathbf{x}}^t - \mathbf{y}^t}\| = Q_{max} - Q^* = \sqrt{N_P}/2 - Q^* \approx 1. \tag{20}$$

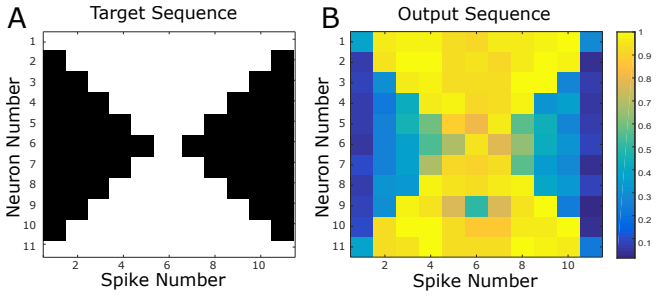

Fig. 4. **Averaged controlled spiking activity for a target hourglass pattern in a population of** 11 **neurons.** (A) The target spike sequence is used to control $N = 11$ connected neurons with unknown connection. (B) The mean spiking activity of the controlled plant network at the time of each spike.

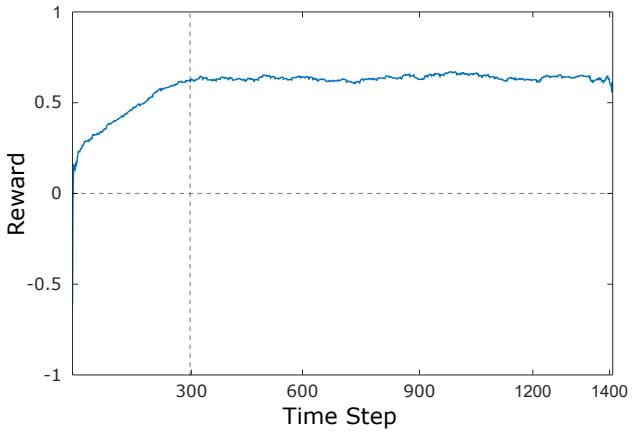

Fig. 5. **The averaged reward increases initially and saturates around** 300 **time steps.** (Average over 50 simulations.)

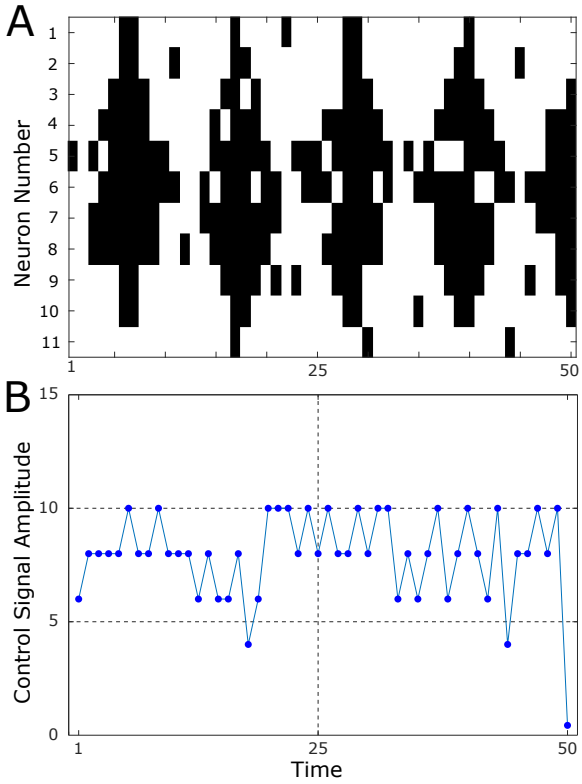

Fig. 6. (A) The CONET successfully controls the plant network to produce hourglass-like patterns through the control signal as shown in (B).

Therefore in Figure 5 we see that CONET successfully learned to control the plant to induce the example spike sequence. Figure 6A shows that, indeed, this the hourglass pattern was induced through a single, one-dimensional, underactuated control signal (Figure 6B) (with some modest error, as expected). The mean output (over the 128 repetitions) of the plant network is shown in Figure 4B, where again shows that the desired sequence is achieved.

IV. CONCLUSIONS

We considered the problem of controlling neural spiking using a small number of control inputs. Given the complexity of neural dynamics, we explored the possibility of using a learning-based approach, wherein an artificial network construct interfaces with network being controlled via the stimulator. We showed numerical proof-of-concept that such an approach can be used to learn a control strategy on-the-fly. More detailed investigation of this approach, including study of convergence properties and efficacy for biophysical neuronal networks, will be the subject of future work.

REFERENCES

[1] P. Dayan and L. F. Abbott, *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems (Computational Neuroscience)*. The MIT Press, 2005.

[2] J. T. Ritt and S. Ching, "Neurocontrol: Methods, models and technologies for manipulating dynamics in the brain." in *IEEE American Control Conference (ACC)*, 2015, pp. 3765–3780.

[3] S. Ching and J. T. Ritt, "Control strategies for underactuated neural ensembles driven by optogenetic stimulation." *Front Neural Circuits*, vol. 7, p. 54, 2013.

[4] L. Grosenick, J. H. Marshel, and K. Deisseroth, "Closed-loop and activity-guided optogenetic control," *Neuron*, vol. 86, no. 1, pp. 106–139, 2015.

[5] I. Dasanayake and J.-S. Li, "Optimal design of minimum-power stimuli for phase models of neuron oscillators," *Phys. Rev. E*, vol. 83, p. 061916, 2011.

[6] D. Wilson and J. Moehlis, "Optimal chaotic desynchronization for neural populations," *SIAM Journal on Applied Dynamical Systems*, vol. 13, no. 1, p. 276, 2014.

[7] A. Nandi, H. Schttler, and S. Ching, "Selective spiking in neuronal populations," in *2017 American Control Conference (ACC)*, May 2017, pp. 2811–2816.

[8] A. Nandi, J. R. H. Schttler, and S. Ching, "Fundamental limits of forced asynchronous spiking with integrate and fire dynamics," *Journal of Mathematical Neuroscience (to appear)*, pp. 2811–2816, May 2017.

[9] Y. Ahmadian, A. M. Packer, R. Yuste, and L. Paninski, "Designing optimal stimuli to control neuronal spike timing," *J. Neurophysiol.*, vol. 106, pp. 1038– 1053, 2011.

[10] A. Nandi, M. Kafashan, and S. Ching, "Control analysis and design for statistical models of spiking networks," *IEEE Transactions on Control of Network Systems*, vol. PP, no. 99, pp. 1–1, 2017.

[11] B. A. Francis and W. M. Wonham, "The internal model principle of control theory," *Automatica*, vol. 12, no. 5, pp. 457–465, 1976.

[12] O. Omidvar and D. L. Elliott, *Neural systems for control*. Elsevier, 1997.

[13] S. Liu and S. Ching, "Recurrent information optimization with local, metaplastic synaptic dynamics." *Neural Computation*, vol. 29, no. 9, pp. 2528–2552, 2017.

[14] L. N. Cooper, N. Intrator, B. S. Blais, and H. Z. Shouval, *Theory Of Cortical Plasticity*. World Scientific Publishing Company, 2004.

[15] Y. Shu, A. Hasenstaub, and D. A. McCormick, "Turning on and off recurrent balanced cortical activity." *Nature*, vol. 423, p. 288293, 2003.

[16] T. P. Vogels, H. Sprekeler, F. Zenke, C. Clopath, and W. Gerstner, "Inhibitory plasticity balances excitation and inhibition in sensory pathways and memory networks." *Science*, vol. 334, p. 1569, Dec. 2011.

[17] B. Haider, A. Duque, A. R. Hasenstaub, and D. A. McCormick, "Neocortical network activity in vivo is generated through a dynamic balance of excitation and inhibition." *J. Neurosci.*, vol. 26, p. 45354545, 2006.

[18] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," *J. Artif. Intell. Res.*, vol. 4, pp. 237–285, 1996.

[19] A. Gosavi, "Reinforcement learning: A tutorial survey and recent advances," *INFORMS J. Comput.*, vol. 21, no. 2, pp. 178–192, 2009.

[20] C. Szepesvari and C. Szepesvari, *Algorithms for Reinforcement Learning*. Morgan & Claypool Publishers, 2010.

[21] V. R. Konda and J. N. Tsitsiklis, "On actor-critic algorithms," *SIAM J. Control Optim.*, vol. 42, no. 4, pp. 1143–1166, 2003.

[22] I. Grondman, L. Busoniu, G. A. D. Lopes, and R. Babuska, "A survey of actor-critic reinforcement learning: Standard and natural policy gradients," *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, vol. 42, no. 6, pp. 1291–1307, 2012.