An Iterative Approach to Optimal Control Design for Oscillator Networks

Bharat Singhal[1], Minh Vu[1], Shen Zeng[1], and Jr-Shin Li[1,2]

Abstract— **We propose a computational framework for optimal control design of oscillator networks. We first introduce a new system representation to eliminate challenges arising from the periodic nature of oscillators. The representation allows us to consider the general problem of pattern formation for oscillators as a classical point-to-point steering. We then develop a novel control design technique that offers the flexibility to blend the time-optimal and energy-optimal considerations with a parameter of choice. We demonstrate the applicability of the proposed framework to a variety of neuroscience applications.**

I. INTRODUCTION

The majority of our physiological activities, such as sleep-cycles [1], memory formation [2], and central nervous system rhythms [3], are governed by the collective behavior of interactive oscillating neurons. Often, for these complex systems to function properly, specific dynamical patterns, such as synchrony or, in some cases, dyssynchrony, must be formed. Disruptions in the system's natural dynamical patterns could quickly lead to pathological conditions. To address this issue, it is of practical interest to be able to design external stimuli that can quickly and efficiently regulate the neuronal oscillators toward desirable configurations. One such example is Parkinson's disease, which is caused by excessive neuron synchronization, and desynchronization is necessary to alleviate tremors, which is typically accomplished by injecting a high-frequency current [4]. In this neurological treatment, it is of clinical importance to desynchronize the neurons as fast as possible yet with minimal electric power to prevent damage to neural tissues.

The immense scale of the oscillatory systems observed in nature and the uncertainty associated with their dynamics often present great challenges to the control of oscillators. The theory of phase model reduction provides a compelling tool to facilitate tractable control design by representing the dynamics of an oscillator around its stable limit cycle with a one-dimensional model that describes the evolution of the phase of oscillation [5]. The main advantage of employing phase models is that it reduces the complexity of the oscillator dynamics while maintaining the interpretability of the model with a small number of parameters. Various control-theoretic methods leveraging the phase model description of oscillators have been proposed. These methods include the design of time-optimal control of a single neural oscillator [6], [7] and minimum-power control to regulate the spiking behavior of neurons [8], [9], [10]. A procedure for designing an optimal energy input that desynchronizes the neural population by maximizing the Lyapunov exponent was also developed in [11].

Despite recent progress, the existing control methods mostly focus on designing either a time-optimal or energy-optimal control solution (where only a single oscillator is considered in the time-optimal case) [10], [6], [8], [11]. In addition, certain practical issues originating from the periodicity of the oscillator's phase during implementations have not been addressed. In this paper, we present a general computational optimal control framework, including both system representation and control design, to construct desired dynamical patterns for an ensemble of coupled oscillators. In Section II, we first introduce a new representation of the oscillators' phase dynamics to avoid the 2π-ambiguity arising from the periodic evolution of phase. Then, in Section III, we build upon the idea of iterative control design [12], [13], [14] to develop a novel mechanism that allows the user to effectively blend time optimality and energy optimality by regulating a single parameter, which offers great flexibility for the optimal control design. In Section IV, we demonstrate the applicability of the proposed framework through a wide range of control examples for ensembles of oscillators.

II. A NEW REPRESENTATION OF OSCILLATOR DYNAMICS FOR PHASE ASSIGNMENT

A. Phase models

For a nonlinear oscillator exhibiting stable limit-cycle oscillation, we consider a reduced-order one-dimensional phase model in the form of

$$\dot{\theta} = \omega + z(\theta)u \tag{1}$$

where θ, $\omega = \frac{2\pi}{T}$, and T denote the phase, the natural frequency, and the period of the oscillator, respectively. $z(\theta)$ is the Phase Response Curve (PRC) of the oscillator, which is a $2\pi-$periodic function describing the infinitesimal change in phase due to an external input [15].

B. Representation of phase dynamics for a single oscillator

Given an oscillator described by equation (1), the phase assignment problem is to construct a control input u that drives the oscillator from a given initial phase θ_0 to a desired final phase θ_d in a given time T. A major practical challenge in designing such control inputs arises from the 2π-periodicity of phase, which creates an infinite number

[1]Department of Electrical and Systems Engineering, Washington University in St. Louis, St. Louis, MO, 63130, USA. Correspondence to: jsli@wustl.edu. [2] Division of Biology and Biomedical Sciences, Washington University in St. Louis, St. Louis, MO, 63130, USA. Bharat Singhal and Minh Vu contributed equally to this work. This work was supported in part by the National Science Foundation under the awards ECCS-1810202 and CMMI-1933976, and by the Air Force Office of Scientific Research under the award FA9550-21-1-0335.

of possible final states, i.e., all the final states $2m\pi + \theta_d$, $m \in \mathbb{Z}^+$, corresponds to the desired phase θ_d. Oscillator control techniques often leave it up to the practitioner to (manually) identify the final state with an appropriate value of m [10]. When the number of oscillators to be controlled is large and the oscillators are coupled, determining a suitable value of m for each oscillator becomes extremely difficult, yet failure to do so can result in considerably large and undesirable controls. This limitation poses great challenges on practical implementations and control designs, especially for a large-scale ensemble of coupled oscillators.

To overcome the aforementioned limitation, our approach is to map the system dynamics onto another space so that all of the possible final phases merge into a single point. More specifically, we consider a differentiable mapping $g : \mathbb{R} \to \mathbb{S}$, where $\mathbb{S}$ denotes a unit circle, such that $g(\theta(t)) = (\sin(\theta(t)), \cos(\theta(t)))'$ with $\theta(t)$ representing the phase of the oscillator. Due to the introduced mapping, all possible final phases $2m\pi + \theta_d$ are mapped, or coincide, to a new and unique state $(x_1, x_2)' := (\sin(\theta_d), \cos(\theta_d))'$. Now, we begin to establish the dynamics of new states with

$$\begin{aligned} \dot{x}_1(t) &= x_2(\omega + z(\theta)u) \\ \dot{x}_2(t) &= -x_1(\omega + z(\theta)u). \end{aligned} \tag{2}$$

To eliminate the explicit dependency of the new state variables on the phase variable θ, we leverage the fact that $z(\theta)$ is a $2\pi-$periodic function and thus can be written, without loss of generality, as a truncated Fourier series, which allows us to express the PRC as a function of the new state variables, i.e., $z(\theta) = h(\sin(\theta), \cos(\theta))$. By substituting the PRC representation into the equation (2), we obtain the dynamics of the system in the new state space, i.e.,

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} \omega x_2 + x_2 h(x_1, x_2)u \\ -\omega x_1 - x_1 h(x_1, x_2)u \end{bmatrix}. \tag{3}$$

C. Representation of phase dynamics for an ensemble of coupled oscillators

We now generalize the above presented approach to a population of oscillators. For a network of n weakly coupled oscillators, the dynamics of the phase of the i^{th} oscillator, $i = 1, \ldots, n$, can be expressed as

$$\dot{\theta}_i(t) = \omega_i + \sum_{1=j\neq i}^{n} k_{ij}(\theta_i(t) - \theta_j(t)) + z_i(\theta_i)u \tag{4}$$

where θ_i, ω_i, and $z_i(\theta_i)$ are the phase, natural frequency, and the PRC of the oscillator i, respectively [16]. The interaction between two oscillators i and j is characterized by the coupling function k_{ij}, where if there is no interaction between the oscillators i and j then the coupling functions k_{ij} and k_{ji} are trivially zero.

Given a network of coupled oscillators as described by equation (4), the objective of the phase assignment problem is to design a broadcast control input that steers the system from a given initial phase $\Theta_0 = (\theta_{1,0}, \theta_{2,0}, \ldots, \theta_{n,0})' \in \mathbb{R}^n$ to a desired final phase $\Theta_d = (\theta_{1,d}, \theta_{2,d}, \ldots, \theta_{n,d})' \in \mathbb{R}^n$ in a given time T. By following a similar procedure as in Section II-B, we define a differentiable mapping g such that

$$g : \begin{pmatrix} \theta_1 \\ \vdots \\ \theta_n \end{pmatrix} \mapsto \begin{pmatrix} \sin(\theta_1) \\ \cos(\theta_1) \\ \vdots \\ \sin(\theta_n) \\ \cos(\theta_n) \end{pmatrix}$$

and let $\boldsymbol{x} = (x_{1,1}, x_{1,2}, \ldots, x_{n,1}, x_{n,2})' \in \mathbb{R}^{2n}$ be the new state variables, where $x_{i,1} = \sin(\theta_i)$ and $x_{i,2} = \cos(\theta_i)$ for $i = 1, 2, \ldots, n$. To transform system dynamics given by equation (4) to the new state space, we first represent the PRC of each oscillator i as a function of the new state variables (as in Section II-B), i.e.,

$$z_i(\theta_i) := h_i(\sin(\theta_i), \cos(\theta_i)) = h_i(x_{i,1}, x_{i,2}) = h_i(\boldsymbol{x}).$$

Similar to PRC, the coupling functions between oscillators are $2\pi-$periodic functions of their phase differences and can be expressed as $k_{ij}(\theta_i - \theta_j) = h_{ij}(\sin(\theta_i - \theta_j), \cos(\theta_i - \theta_j))$, where the phase differences can be further described in terms of $\boldsymbol{x}$, i.e.,

$$\begin{aligned} \sin(\theta_i - \theta_j) &= x_{i,1}x_{j,2} - x_{i,2}x_{j,1} \\ \cos(\theta_i - \theta_j) &= x_{i,1}x_{j,1} + x_{i,2}x_{j,2}. \end{aligned}$$

With these preparations, we now obtain the following dynamics of the oscillator system in the new state space

$$\begin{bmatrix} \dot{x}_{1,1} \\ \dot{x}_{1,2} \\ \vdots \\ \dot{x}_{n,1} \\ \dot{x}_{n,2} \end{bmatrix} = \begin{bmatrix} x_{1,2}(\omega_1 + f_1(\boldsymbol{x}) + h_1(\boldsymbol{x})u) \\ -x_{1,1}(\omega_1 + f_1(\boldsymbol{x}) + h_1(\boldsymbol{x})u) \\ \vdots \\ x_{n,2}(\omega_n + f_n(\boldsymbol{x}) + h_n(\boldsymbol{x})u) \\ -x_{n,1}(\omega_n + f_n(\boldsymbol{x}) + h_n(\boldsymbol{x})u) \end{bmatrix} \tag{5}$$

where $f_i(\boldsymbol{x}) = \sum_{1=j\neq i}^{n} h_{ij}(x_{i,1}, x_{i,2}, x_{j,1}, x_{j,2})$.

Up to this point, we have introduced a new state-space representation for the dynamics of the oscillator ensemble. The proposed representation eliminates the 2π-ambiguity arising from the periodic nature of the oscillator's phase and allows us to completely avoid the problem of manually determining an appropriate final state (for phase assignment). In the following, we present a general optimal control framework which allows us to blend the consideration of time optimality with energy optimality. In our framework, we also relax the consideration of a fixed control horizon and let the algorithm search for a (favorable) optimal control solution with a varying final time $T \in [T_{\min}, T_{\max}]$.

III. ITERATIVE OPTIMAL CONTROL SYNTHESIS FOR AN ENSEMBLE OF OSCILLATORS

We consider the problem of steering an oscillator ensemble from a given initial phase $\Theta_0 = (\theta_{1,0}, \theta_{2,0}, \ldots, \theta_{n,0})' \in \mathbb{R}^n$ to a desired final phase $\Theta_d = (\theta_{1,d}, \theta_{2,d}, \ldots, \theta_{n,d})' \in \mathbb{R}^n$. This problem, in the proposed state space, is equivalent to steering the system, as described in (5), from the initial state $\boldsymbol{x}_0 = g(\Theta_0)$ to the corresponding desired state $\boldsymbol{x}_d = g(\Theta_d)$. We start by discretizing the continuous-time system in (5) using the zero-order hold assumption, i.e., $u(t) \equiv u_k \in \mathbb{R}$, $t \in [k\tau, (k+1)\tau]$ where $\tau > 0$ denotes the time step size, and obtain a discrete-time evolution of the system in the form $\boldsymbol{x}_{k+1} = F(\boldsymbol{x}_k, u_k, \tau)$. We utilize high-order Taylor expansion, similar to [17], to symbolically calculate F and the Jacobian of F, which prepares for the following linearization of the system at each time step.

Now, given an initial state $\boldsymbol{x}_0$, a nominal control input $U := [u_0, u_1, \ldots, u_{N-1}]'$, and the resulting state trajectory $\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$, with N being the number of time steps, for small perturbations of the nominal input U and the nominal step size τ, we consider the discrete-time linearization of the system, i.e.,

$$\delta\boldsymbol{x}_{k+1} = A_k \delta\boldsymbol{x}_k + B_k \delta u_k + C_k \delta\tau, \qquad \delta\boldsymbol{x}_0 = 0 \tag{6}$$

where $A_k := \frac{\partial F}{\partial x}(\boldsymbol{x}_k, u_k, \tau), B_k := \frac{\partial F}{\partial u}(\boldsymbol{x}_k, u_k, \tau), C_k := \frac{\partial F}{\partial \tau}(\boldsymbol{x}_k, u_k, \tau)$, and $\delta\boldsymbol{x}_k$, δu_k, $\delta\tau$ denote the incremental changes of the nominal state trajectory, the nominal input, and the nominal step size, respectively. Iterating (6), we have

$$\begin{aligned}
\delta\boldsymbol{x}_1 &= B_0 \delta u_0 + C_0 \delta\tau \\
\delta\boldsymbol{x}_2 &= A_1 B_0 \delta u_0 + B_1 \delta u_1 + A_1 C_0 \delta\tau + C_1 \delta\tau \\
&\vdots \\
\delta\boldsymbol{x}_N &= A_{N-1} \ldots A_1 B_0 \delta u_0 + \cdots + B_{N-1} \delta u_{N-1} \\
&\qquad + A_{N-1} \ldots A_1 C_0 \delta\tau + \cdots + C_{N-1} \delta\tau.
\end{aligned}$$

Then, given that the perturbations $\delta\tau$ and $\Delta U := [\delta u_0, \delta u_1, \ldots, \delta u_{N-1}]'$ are sufficiently small, we utilize the above linearizations to approximately describe the terminal state of the system trajectory, i.e.,

$$\tilde{\boldsymbol{x}}_N \approx \boldsymbol{x}_N + H_U \Delta U + H_\tau \mathbf{1} \delta\tau \tag{7}$$

where $\tilde{\boldsymbol{x}}_N$ is the resulting terminal state of the trajectory due to the perturbed input $U + \Delta U$ and the adjusted step size $\tau + \delta\tau$, $H_U := \begin{bmatrix} A_{N-1} \ldots A_1 B_0 & \cdots & B_{N-1} \end{bmatrix}$, $H_\tau := \begin{bmatrix} A_{N-1} \ldots A_1 C_0 & \cdots & C_{N-1} \end{bmatrix}$, and $\mathbf{1} := \begin{bmatrix} 1 & \cdots & 1 \end{bmatrix}' \in \mathbb{R}^N$.

The central idea of our control method is to leverage the locally faithful approximation in (7) so as to gradually alter the input U and the step size τ, by determining appropriate values of ΔU and $\delta\tau$, to step-by-step optimize some desired control objective (described below). Using this idea, we lay out in the following a (two-part) sequence of iterations to synthesize optimal control inputs for an ensemble of oscillators. We first focus on steering the system to the desired target state then fine-tune the control input and the step size in order to achieve a minimum-time and/or minimum-energy control.

A. Iterative Steering

To step by step steer the system closer to the desired state, in each iteration, we consider the following optimization

$$\begin{aligned}
\underset{\Delta U, \delta\tau}{\text{minimize}} \quad & \|\boldsymbol{x}_N + H_U \Delta U + H_\tau \mathbf{1} \delta\tau - \boldsymbol{x}_d\|_2^2 \\
\text{subject to} \quad & U_{\text{min}} \leq U + \Delta U \leq U_{\text{max}} \\
& \Delta U_{\text{lb}} \leq \Delta U \leq \Delta U_{\text{ub}} \\
& T_{\text{min}} \leq N(\tau + \delta\tau) \leq T_{\text{max}} \\
& \delta\tau_{\text{lb}} \leq \delta\tau \leq \delta\tau_{\text{ub}}
\end{aligned} \tag{8}$$

where U_{min}, U_{max}, T_{min}, T_{max} denote the control and time limits, and ΔU_{lb}, ΔU_{ub}, $\delta\tau_{\text{lb}}$, $\delta\tau_{\text{ub}}$ denote sufficiently tight lower and upper bounds of the control perturbation and the adjustment of step size so as to ensure the appropriateness of the approximation in (7). We use the solution of the quadratic program (8) to update (or improve) the current control and the current step size. This process is then repeated, which is outlined in the following algorithm.

Algorithm 1 Steering oscillators to the desired phases

Require: Desired final phases Θ_d for an ensemble of oscillators and an initial (arbitrary) input U.
1: Compute the corresponding final state $\boldsymbol{x}_d = g(\Theta_d)$.
2: Apply the input U to the system and store all A_k, B_k, C_k.
3: Calculate H_U and H_τ.
4: Solve for ΔU^* and $\delta\tau^*$ of the optimization problem (8).
5: Update the input and the step size via $U = U + \Delta U^*$ and $\tau = \tau + \delta\tau^*$.
6: Repeat steps $2-5$ until $\|\boldsymbol{x}_N - \boldsymbol{x}_d\|_2 \leq \epsilon_{1,\text{tol}}$.

Note that we provide the algorithm flexibility to adjust both the control input and the total time horizon to achieve the steering task. If a particular control horizon is required, one can simplify problem (8) accordingly (by removing $\delta\tau$, $H_\tau \mathbf{1}$, and the last two constraints).

B. Optimal Steering

Once the system reaches sufficiently close to the desired target by the end of Algorithm 1, we consider the following optimization to step-by-step fine-tune the control solution

$$\begin{aligned}
\underset{\Delta U, \delta\tau}{\text{minimize}} \quad & \alpha \|U + \Delta U\|^2 + (1 - \alpha)(N(\tau + \delta\tau))^2 \\
\text{subject to} \quad & \boldsymbol{x}_N + H_U \Delta U + H_\tau \mathbf{1} \delta\tau = \boldsymbol{x}_d \\
& U_{\text{min}} \leq U + \Delta U \leq U_{\text{max}} \\
& \Delta U_{\text{lb}} \leq \Delta U \leq \Delta U_{\text{ub}} \\
& T_{\text{min}} \leq N(\tau + \delta\tau) \leq T_{\text{max}} \\
& \delta\tau_{\text{lb}} \leq \delta\tau \leq \delta\tau_{\text{ub}}
\end{aligned} \tag{9}$$

where U_{min}, U_{max}, T_{min}, T_{max} and ΔU_{lb}, ΔU_{ub}, $\delta\tau_{\text{lb}}$, $\delta\tau_{\text{ub}}$ serve similar purposes as in the first part. $\alpha \in [0, 1]$ is a blending factor that allows us to balance the penalty between control energy and total time. Similar to the first part, the procedure to achieve a time and/or energy-optimal control solution is outlined in Algorithm 2.

Algorithm 2 Minimizing control energy and time horizon

Require: A nominal input U steering the system to $\boldsymbol{x}_d$ and a blending factor $\alpha \in [0, 1]$.
1: Apply the input U to the system and store all A_k, B_k, C_k.
2: Calculate H_U and H_τ.
3: Solve for ΔU^* and $\delta\tau^*$ of the optimization problem (9).
4: Update the input and the step size via $U = U + \Delta U^*$ and $\tau = \tau + \delta\tau^*$.
5: Repeat step $1-4$ until $\|\Delta U^*\|_2 \leq \epsilon_{2,\text{tol}}$ and $|\delta\tau^*| \leq \epsilon_{3,\text{tol}}$.

The importance of the above blending factor is that it allows us to investigate the trade-off between the energy consumption and the duration of the control, with $\alpha = 1$ corresponding to the minimal-energy consideration and $\alpha = 0$ being the minimal-time. This mechanism provides us the flexibility to find the right balance for optimality depending on the application of interest.

In summary, the presented optimal control framework strategically transforms the steering problem into an iterative sequence of quadratic programs. We begin this process with Algorithm 1 to steer the oscillator system to the desired target. Then, using the input thereof as a nominal control, we employ Algorithm 2 to optimize the control input as well as step size and eventually obtain a minimal time-energy point-to-point control solution. Note that the presented framework is also applicable to scenarios where relative phase differences among the oscillators, not the absolute phase of each oscillator, are of particular interest, though with some necessary extensions which are included in the Appendix.

IV. NUMERICAL IMPLEMENTATION

In this section, we illustrate the flexibility of the presented control framework in achieving synchronization, resynchronization, and other pattern formations of neurons.

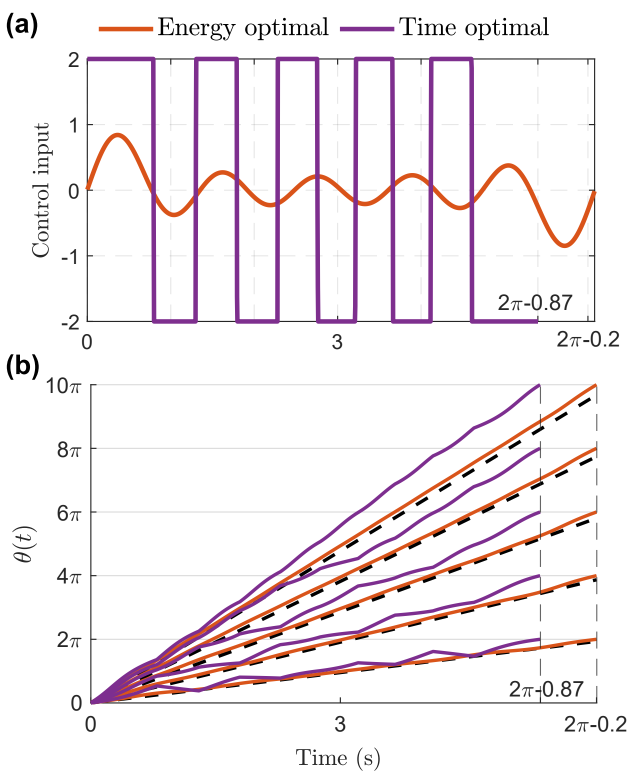

Fig. 1. Synchronization of 5 sinusoidal neurons using energy-optimal control and time-optimal control. (a) The orange line denotes the energy-optimal control for the simultaneously spiking of neurons; while the purple line is the time-optimal control input ($|u| \leq 2$) for fast synchronization. (b) Phase trajectories when no control is applied (dashed black lines), when energy-optimal control is applied (orange), and when time-optimal control is applied (purple).

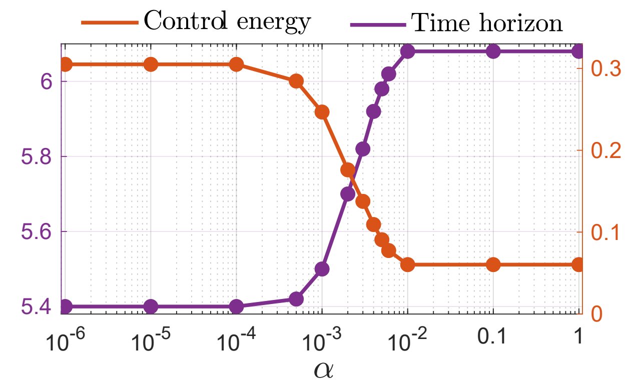

Fig. 2. Trade-off between input power and synchronization time is determined by the blending factor α. Control energy is computed by $\sum_{k=0}^{N} u_k^2 \tau$.

A. Minimal-energy synchronization of neurons

The essence of a neuron's firing dynamics can be captured by considering the evolution of the neuron's phase [16]. As a convention, a neuron is said to spike or fire at time T if its phase evolves from $\theta(0) = 0$ to $\theta(T) = 2n\pi$, $n \in \mathbb{N}_0$. Note that the spike of a neuron corresponds to multiple locations in the phase model but only a single state, i.e., $\sin(\theta) = 0$, $\cos(\theta) = 1$, in our proposed representation. This advantage allows us to avoid the manual task of selecting the exact target phase (in multiples of 2π) for the neurons as in [10] and focus our attention on the optimal control design.

The typical objectives for synchronization of neurons are to design a minimum-energy control input that fires all neurons together before a specified time T, or an input that fires all neurons together in the shortest duration with some given input bounds. In this example, we consider synchronization of five neurons having sinusoidal PRC, commonly used to represent Type II neuron models like Fitzhugh–Nagumo model [18], with dispersed frequencies $(\omega_1, \omega_2, \omega_3, \omega_4, \omega_5) = (1, 2, 3, 4, 5)$. Note that given these natural frequencies, in the absence of any control input, all the neurons will fire at $T = 2\pi$. We consider two objectives: one is to design a minimum-energy control that synchronizes the neurons before the time $T = 2\pi - 0.2$ and the second objective is to achieve the fastest synchronization with $|U| \leq 2$. To this end, in the proposed state space, we first apply Algorithm 1 (stopping criteria $\epsilon_{1,\text{tol}} = 0.01$) to steer each oscillator to the desired target state $(0, 1)'$. Then, we employ Algorithm 2 to obtain the minimum-time ($\alpha = 0$) and the minimum-energy ($\alpha = 1$) controls.

Figure 1 illustrates the two optimal controls and their effects on the phase evolution of the neurons. We observe that the energy-optimal control uses the entire allowable time horizon while the time-optimal control is of bang-bang nature (a usual property of time-optimal controls). In addition, by varying α we have the flexibility to balance between the consideration of minimum energy and minimum time. Figure 2 displays the trade-off between input energy and synchronization time for various α, illustrating that energy saving comes with a cost of an increase in control duration.

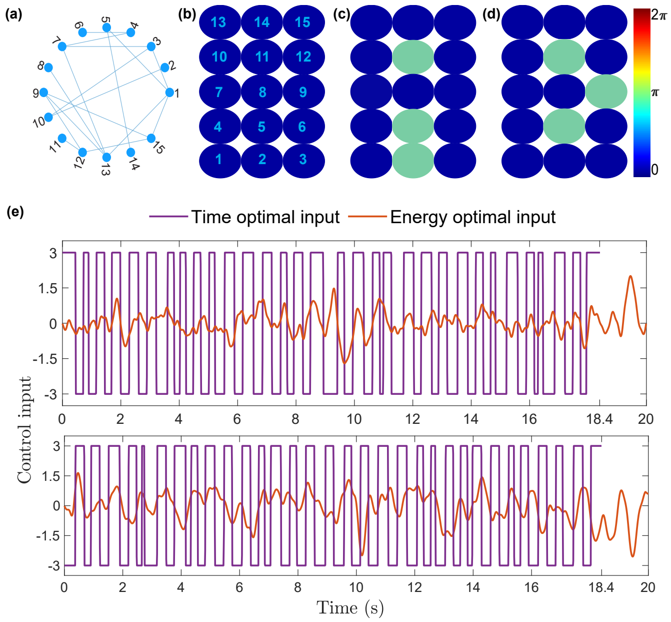

Fig. 3. Spatial pattern formation in a network of sinusoidally coupled neurons with uniformly distributed frequencies in $[1, 10]$ rad/s and sinusoidal PRCs. (a) The connectivity structure of the network. (b) Initial phases of the neurons. (c) Starting with the phases in (b), the pattern "A" is created. (d) From the pattern "A", the pattern "B" is formed. (e) Time-optimal (purple line) and energy-optimal (orange line) control inputs driving the system from (b) to (c) (Top panel); similarly, from (c) to (d) (Bottom panel).

In the following, we present different pattern formations for a large network of coupled oscillators where manually determining the final phase of each oscillator for the purpose of optimal control design becomes unmanageable.

B. Pattern formation in an ensemble of coupled neurons

Distinct spatial patterns of neuronal activity levels in the brain are associated with memory formation [19]. Motivated by this, we consider the problem of designing energy-optimal and time-optimal control inputs which steer an ensemble of coupled neurons between some spatially desired patterns. To accomplish this, we require the control inputs to enhance the firing of a spatially selected group of neurons while simultaneously inhibiting the firing of the remaining neurons. Following the convention in the previous example, we equate neuron firing to phase $2m\pi$, and neuron inhibition can thus be associated with phase $(2m + 1)\pi$, where $m \in \mathbb{Z}^+$. Equivalently, in the proposed state-space, a neuron can be fired or inhibited by steering it to the state $(0, 1)'$ or $(0, -1)'$, respectively. We consider a network of 15 weakly coupled sinusoidal PRC neurons, as in Figure 3(a), where $k_{ij}(\theta_j - \theta_i) = 0.1 \sin(\theta_j - \theta_i)$ if a connection exists between node i and j, otherwise 0. The neurons are spatially distributed in a 5×3 grid according to Figure 3(b) with neuron 1 (Bottom left circle) being the slowest and neuron 15 (Top right circle) being the fastest. The natural frequencies of the neurons are uniformly distributed in $[1, 10]$.

We start with a uniform phase distribution as in Figure 3(b) and fire spatially selected neurons such that the pattern "A" is created; similarly, we create pattern "B" from the initial pattern "A." For both of these patterns, we design minimum-energy input that generates the desired pattern before $T = 20s$ by first steering the system to its desired state using Algorithm 1 and then applying Algorithm 2 with $\alpha = 1$. We also construct control inputs that generate the desired spatial firing pattern in the shortest duration ($|U| \leq 3$) by applying Algorithm 2 with $\alpha = 0$ after steering the system to its desired states. The purple lines (similarly orange) in Figure 3(e) depict the corresponding time-optimal control (similarly energy-optimal) inputs to generate pattern "A" (top panel) and "B" (bottom panel). Note that similar to the previous example the energy-optimal input uses the entire allowable time horizon while the time-optimal input is of bang-bang nature.

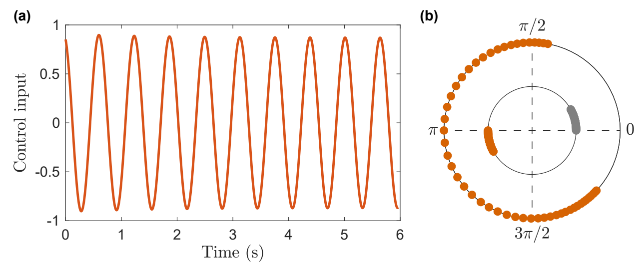

Fig. 4. Desynchronization of a population of 100 identical neurons with sinusoidal PRCs. (a) The designed control input. (b) The inner circle displays the uncontrolled final (orange) and initial (grey) phases of the neurons; while the outer circle depicts the final controlled phases.

C. Desynchronization in an ensemble of neurons

It has been suggested that pathological synchronization of neurons in the brain causes Parkinson's disease [11]. Deep Brain Stimulation (DBS), a well-established technique for alleviating such tremors, is known to desynchronize the pathologically synchronized neurons [4]. Here, we show a possible application of our control framework to desynchronize a population of neurons through the means of a numerical simulation. We consider a population of 100 identical neurons with sinusoidal PRCs ($\omega = 10$) and design a control input to distribute the phases of the neurons in the range $[0, 1.5\pi)$ at time $T = 1.9\pi s$. This is done by assigning a phase difference of $1.5\pi/100$ between successive oscillators. To achieve this, we slightly modify the control technique to regulate only the relative phases (see Appendix). The control input and the final phases are depicted in Figure 4(a) and (b), respectively.

V. CONCLUSIONS

In this paper, we introduced a new state-space representation for an oscillator's dynamics which eliminates the 2π-ambiguity of the oscillator's phase and a flexible optimal control design technique that allows the user to blend the consideration of time with control energy. The combined development offers a rather general computational framework for optimal control design of oscillator networks, which has been demonstrated through a wide range of examples. Owing to its generalizability and simple implementation, our work could open new pathways toward tractable control design for practical applications of networks of oscillators.

REFERENCES

[1] A. T. Winfree, *Circadian Rhythms in General*, pp. 545–591. New York, NY: Springer New York, 2001.

[2] S. Hanslmayr and T. Staudigl, "How brain oscillations form memories — a processing based perspective on oscillatory subsequent memory effects," *NeuroImage*, vol. 85, pp. 648–655, 2014. New Horizons for Neural Oscillations.

[3] G. Buzsáki, *Rhythms of the Brain*. Oxford University Press, oct 2006.

[4] C. J. Wilson, B. Beverlin, and T. Netoff, "Chaotic desynchronization as the therapeutic mechanism of deep brain stimulation," *Frontiers in Systems Neuroscience*, vol. 5, 2011.

[5] H. Nakao, "Phase reduction approach to synchronisation of nonlinear oscillators," *Contemporary Physics*, vol. 57, pp. 188–214, Oct. 2015.

[6] A. Nabi and J. Moehlis, "Time optimal control of spiking neurons," *Journal of mathematical biology*, vol. 64, no. 6, pp. 981–1004, 2012.

[7] I. Dasanayake and J.-S. Li, "Charge-balanced time-optimal control for spiking neuron oscillators," in *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, pp. 1651–1656, IEEE, 2012.

[8] I. Dasanayake and J.-S. Li, "Optimal design of minimum-power stimuli for phase models of neuron oscillators," *Physical Review E*, vol. 83, no. 6, p. 061916, 2011.

[9] W. B. Bomela, I. S. Dasanayake, J.-S. Li, Y. Chen, and I. Z. Kiss, "Optimal phase-to-phase control of chemical oscillations," *Industrial & Engineering Chemistry Research*, vol. 57, pp. 7764–7770, May 2018.

[10] J.-S. Li, I. Dasanayake, and J. Ruths, "Control and synchronization of neuron ensembles," *IEEE Transactions on Automatic Control*, vol. 58, pp. 1919–1930, Aug. 2013.

[11] D. Wilson and J. Moehlis, "Optimal chaotic desynchronization for neural populations," *SIAM Journal on Applied Dynamical Systems*, vol. 13, pp. 276–305, Jan. 2014.

[12] S. Zeng, "Iterative optimal control syntheses illustrated on the brockett integrator," *IFAC-PapersOnLine*, vol. 52, no. 16, pp. 138–143, 2019.

[13] M. Vu, S. Zeng, and H. Fang, "Health-aware battery charging via iterative nonlinear optimal control syntheses," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 12485–12490, 2020.

[14] M. Vu, H. Fang, and S. Zeng, "Nonlinear optimal control synthesis using basis functions: Algorithms and examples," in *2021 IEEE Conference on Control Technology and Applications (CCTA)*, pp. 493–498, IEEE, 2021.

[15] A. Pikovsky, J. Kurths, M. Rosenblum, J. Kurths, B. Chirikov, C. U. Press, P. Cvitanovic, F. Moss, and H. Swinney, *Synchronization: A Universal Concept in Nonlinear Sciences*. Cambridge Nonlinear Science Series, Cambridge University Press, 2001.

[16] R. M. Smeal, G. B. Ermentrout, and J. A. White, "Phase-response curves and synchronized neural networks," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 365, pp. 2407–2422, Aug. 2010.

[17] N. Kazantzis and C. Kravaris, "Time-discretization of nonlinear control systems via taylor methods," *Computers & chemical engineering*, vol. 23, no. 6, pp. 763–784, 1999.

[18] J. Keener and J. Sneyd, eds., *Mathematical Physiology*. Springer New York, 2009.

[19] E. Zarahn, B. C. Rakitin, D. Abela, J. Flynn, and Y. Stern, "Distinct spatial patterns of brain activity associated with memory storage and search," *NeuroImage*, vol. 33, pp. 794–804, Nov. 2006.

APPENDIX

In the main text, we presented a control framework for obtaining a specified desired phase for each oscillator in an ensemble. Nonetheless, in certain applications, such as uniform desynchronization of neurons, the relative phases between the oscillators, rather than the absolute phase of each oscillator, is of particular interest. To this end, we adapt the presented control framework to regulating the relative phase differences, $\Delta\theta_{i\setminus j} := \theta_i - \theta_j$, between oscillators. In the proposed state space, $\Delta\theta_{i\setminus j}$ can be conveniently and uniquely described by

$$\begin{aligned}\sin(\Delta\theta_{i\setminus j}) &= \sin(\theta_i - \theta_j) = x_{i,1}x_{j,2} - x_{i,2}x_{j,1}\\ \cos(\Delta\theta_{i\setminus j}) &= \cos(\theta_i - \theta_j) = x_{i,1}x_{j,1} + x_{i,2}x_{j,2}.\end{aligned}$$

Now, to characterize the relative phase differences between the successive oscillators in an ensemble of n oscillators, we consider $\mathcal{R} : \mathbb{R}^{2n} \to \mathbb{R}^{2(n-1)}$, where

$$\mathcal{R} : \boldsymbol{x} \mapsto \begin{pmatrix} x_{1,1}x_{2,2} - x_{1,2}x_{2,1} \\ x_{1,1}x_{2,1} + x_{1,2}x_{2,1} \\ \vdots \\ x_{n-1,1}x_{n,2} - x_{n-1,2}x_{n,1} \\ x_{n-1,1}x_{n,1} + x_{n-1,2}x_{n,1} \end{pmatrix}. \tag{10}$$

Let $\Delta\Theta^* = (\Delta\theta^*_{1\setminus 2}, \Delta\theta^*_{2\setminus 3}, \ldots, \Delta\theta^*_{n-1\setminus n})' \in \mathbb{R}^{2(n-1)}$ be some desired phase differences between successive oscillators. Our goal is to steer a population of oscillators from a given initial phase $\Theta_0 = (\theta_{1,0}, \theta_{2,0}, \ldots, \theta_{n,0})' \in \mathbb{R}^n$ to some final phase such that the desired phase difference $\Delta\Theta^*$ is obtained. This problem, in the proposed state space, is equivalent to steering the system, as described in (5), from the initial state $\boldsymbol{x}_0 = g(\Theta_0)$ to a final state $\boldsymbol{x}_N$ such that $\mathcal{R}(\boldsymbol{x}_N) = v^*$ where $v^* := (\sin(\Delta\theta^*_{1\setminus 2}), \cos(\Delta\theta^*_{1\setminus 2}), \ldots, \sin(\Delta\theta^*_{n-1\setminus n}), \cos(\Delta\theta^*_{n-1\setminus n}))'$.

By leveraging the fact that changes of the system trajectory can be adjusted to be minor in each iteration, i.e., $\|\boldsymbol{\delta x}_k\|_2$ are small, we approximately describe the value of $\mathcal{R}(\boldsymbol{x}_N + \boldsymbol{\delta x}_N)$ by considering the linearization of $\mathcal{R}$ at $\boldsymbol{x}_N$, i.e.,

$$\begin{aligned}\mathcal{R}(\boldsymbol{x}_N + \boldsymbol{\delta x}_N) &= \mathcal{R}(\boldsymbol{x}_N) + J_{\mathcal{R}}(\boldsymbol{x}_N)\boldsymbol{\delta x}_N \\ &= \mathcal{R}(\boldsymbol{x}_N) + J_{\mathcal{R}}(\boldsymbol{x}_N)(H_U \Delta U + H_\tau \mathbf{1}\delta\tau)\end{aligned}$$

where $J_{\mathcal{R}}(\boldsymbol{x}_N)$ denotes the Jacobian of $\mathcal{R}$ evaluated at $\boldsymbol{x}_N$. Then, instead of considering problem (8), we consider

$$\begin{aligned}\underset{\Delta U,\, \delta\tau}{\text{minimize}} \quad & \|\mathcal{R}(\boldsymbol{x}_N) + J_{\mathcal{R}}(\boldsymbol{x}_N)(H_U \Delta U + H_\tau \mathbf{1}\delta\tau) - v^*\|_2^2 \\ \text{subject to} \quad & U_{\text{min}} \le U + \Delta U \le U_{\text{max}} \\ & \Delta U_{\text{lb}} \le \Delta U \le \Delta U_{\text{ub}} \\ & T_{\text{min}} \le N(\tau + \delta\tau) \le T_{\text{max}} \\ & \delta\tau_{\text{lb}} \le \delta\tau \le \delta\tau_{\text{ub}}\end{aligned} \tag{11}$$

to gradually regulate $\mathcal{R}(\boldsymbol{x}_N)$ toward the desired value v^*. The procedure for the relative phase assignment is as follows.

Algorithm 3 Regulating phase differences among oscillators

Require: Desired phase differences $\Delta\Theta^*$ between successive oscillators and an initial (arbitrary) input U.

1: From $\Delta\Theta^*$, compute the corresponding desired value v^*.
2: Apply the input U to the system and store all A_k, B_k.
3: Calculate H_U and H_τ.
4: Solve for ΔU^* and $\delta\tau^*$ of the optimization problem (11).
5: Update the control input via $U = U + \Delta U^*$ and $\tau = \tau + \delta\tau^*$.
6: Repeat step $2-5$ until $\|\mathcal{R}(\boldsymbol{x}_N) - v^*\|_2 \le \epsilon_{4,\text{tol}}$.