

Data-Driven Reconstruction of Firing Rate Dynamics in Brain Networks

Xuan Wang

Jorge Cortés

Abstract—This paper studies the reconstruction from data of firing rate dynamics in linear-threshold network models of brain activity. Instead of identifying the system parameters directly, which would lead to a large number of variables and a highly non-convex objective function, the novelty of our approach stems from reformulating the identification problem as a scalar variable optimization of a discontinuous, nonconvex objective function. We formally show that the reformulated optimization problem has a unique solution and establish that it leads to the identification of all the desired system parameters. These results form the basis for the introduction of an algorithm to find the optimizer that identifies the different regions in the domain of definition of the objective function. The results not only validate the system identifiability but also provide the foundation for further research on data-driven control of firing rate dynamics. We demonstrate the algorithm effectiveness in simulation.

I. INTRODUCTION

A key goal of neuroscience is to understand brain function from its dynamical behavior. Among all the measures for brain neuronal activity, firing rate (e.g., number of spikes per second) is a widely adopted tool due to its trial-to-trial reproducibility [1]. Brain neurons are highly interactive [2] and so are their firing behaviors. A common way to describe such interaction is by means of network models [3], [4], composed of a set of nodes with internal dynamics representing populations of neurons and the associated edges with weights characterizing the nodal interactions. However, measuring and quantifying the strength of such interactions is challenging. Motivated by this, our research focuses on using sampled data to reconstruct the firing rate dynamics of brain neural networks, with the ultimate goal of enabling prediction and control of such models.

Literature review: In control theory, two techniques closely related to our research goals are system identification and data-driven control. The former aims to learn the system parameters from data; the latter aims to skip the identification process and design data-based controller directly. With abundant literature in these fields [5], if the systems are linear, well studied frameworks have been proposed for both system identification [6] and data-driven control [7]. However, for nonlinear systems, since they are inherently complex and vary in model structures, developing unified approaches for general systems can be challenging. In fact, nonlinear identification usually requires a model selection process [8]

before parameterizing the system; and for nonlinear data-driven control, the controller structures are usually assumed to be known a priori, for example, in iterative feedback tuning [9], unfalsified control [10], and simultaneous perturbation stochastic approximation [11]. Apart from these, there also exist methods like model-free adaptive control [12] that provide general data-driven approaches for a wider range of nonlinear systems at the cost of ignoring known information about the model structure, potentially leading to performance losses on control accuracy or training efficiency.

In the context of network reconstruction of firing rate dynamics, [13], [14] present results for the linear case by assuming the system can be linearized around its fixed point. Here, following [15], [16], we employ a linear-threshold network model to describe the dynamical behavior, and then introduce a novel identification method to determine its parameters. Such identification process looks similar to the training of neural networks with the rectified linear unit (RELU) activation function [17] in the deep learning literature. However, the research goal for the two problems are fundamentally different. Namely, here we seek to reconstruct the dynamical behavior of a real system, whose nodes' states evolve with time, corresponding to their current states and the system input. In deep learning, the static network model [18], [19] seeks to establish a virtual mapping between input and output data sets, and the model does not involve dynamical behavior. Because of this difference, results in both fields are not directly transferable.

Statement of Contributions: We study the reconstruction from data of firing rate dynamics in linear-threshold network models. We start by noting that the identification of all system parameters would give rise to a highly non-convex problem with a large number of variables. Instead, our approach takes advantage of the specific structure of the linear-threshold dynamics to formulate a scalar variable optimization problem with an piecewise smooth objective function that in general is discontinuous and nonconvex. Our analysis shows that the minimizer of the objective function is unique and establishes that the solution of the optimization problem leads to the identification of all the desired system parameters. Based on this fact, we propose an algorithm that exhaustively identifies the different regions in the domain of definition of the objective function where it is smooth. We show that the proposed algorithm finds the optimizer and characterizes its time and computational complexity. We validate the effectiveness of the proposed algorithm in both synthetic and experimental data from the activity of rodents' brain executing a selective listening task. For reasons of space, the proofs are omitted and will appear elsewhere.

X. Wang is with the Department of Electrical and Computer Engineering, George Mason University, Fairfax, xwang64@gmu.edu. This work started when he was a postdoc researcher at University of California, San Diego. J. Cortés is with the Department of Mechanical and Aerospace Engineering, University of California, San Diego, cortes@ucsd.edu

Notation: Let $\mathbf{1}_r$ denote the vector in \mathbb{R}^r with all entries equal to 1. Let I_r denote the $r \times r$ identity matrix. We let $\text{col}\{A_1, A_2, \dots, A_r\} = [A_1^\top \ A_2^\top \ \dots \ A_r^\top]^\top$ be a vertical stack of matrices A_i possessing the same number of columns. Let $x[i] \in \mathbb{R}$ be the i th entry of vector x ; correspondingly, let $M[i, j] \in \mathbb{R}$ be the entry of matrix M on its i th row and j th column. Let M^\top be the transpose of a matrix M . For $x \in \mathbb{R}$, define the threshold function $[x]_0^s$ with $s > 0$ as

$$[x]_0^s = \begin{cases} s & \text{for } x > s \\ x & \text{for } 0 \leq x \leq s \\ 0 & \text{for } x < 0 \end{cases}$$

For a vector x , $[x]_0^s$ denotes the component-wise application of this definitions. For $x \in \mathbb{R}^r$ and $1 \leq i \leq r$, x_{-i} denotes the vector in \mathbb{R}^{r-1} obtained by removing the i th entry of x .

II. PROBLEM FORMULATION

In this section, we first introduce a continuous-time firing rate dynamical model for neuronal networks following [15] and then convert it to its discrete-time form.

Consider a network, where each node represents a population of neurons with similar activation patterns, evolving according to the linear-threshold dynamics, for $t \geq 0$,

$$\tau \dot{\mathbf{x}}(t) = -\mathbf{x}(t) + [W\mathbf{x}(t) + B\mathbf{u}(t)]_0^s, \quad (1)$$

Here, τ is a time constant capturing the timescale of the neuronal system [1], $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{x} \geq 0$ is the system state, corresponding to the firing rate of the nodes; and $W \in \mathbb{R}^{n \times n}$ is the synaptic connectivity matrix, characterizing the interactions (excitation or inhibition) between different nodes. For $i \in \{1, \dots, n\}$, we assume $W[i, i] = 0$, that is, the nodes do not have self-loops. $\mathbf{u} \in \mathbb{R}^m$ and $B \in \mathbb{R}^{n \times m}$ ($m \leq n$) are the external inputs and the associated input matrix. For each node, the stimulation it receives from its neighboring nodes and external inputs is non-negative and bounded by a threshold s , denoted by $[\cdot]_0^s$.

The discretization of the system (1) by the forward Euler method with a constant step-size $\delta \ll \tau$ yields

$$\frac{\tau}{\delta} (\mathbf{x}^+ - \mathbf{x}) = -\mathbf{x} + [W\mathbf{x} + B\mathbf{u}]_0^s. \quad (2)$$

Here, \mathbf{x} , \mathbf{u} are the current system state and input, and \mathbf{x}^+ is the system state after the interval δ . For convenience of presentation, let

$$\alpha \triangleq 1 - \frac{\delta}{\tau} \in (0, 1), \quad W_D \triangleq \frac{\delta}{\tau} W, \quad B_D \triangleq \frac{\delta}{\tau} B, \quad s_D \triangleq \frac{\delta}{\tau} s,$$

and rewrite (2) into an equivalent form as:

$$\mathbf{x}^+ = \alpha \mathbf{x} + [W_D \mathbf{x} + B_D \mathbf{u}]_0^{s_D}. \quad (3)$$

We assume the system states \mathbf{x}_d , \mathbf{x}_d^+ , and the system inputs \mathbf{u}_d can be sampled. We denote the data samples by $\mathbf{x}_d(k)$, $\mathbf{x}_d^+(k)$ and $\mathbf{u}_d(k)$, respectively, for $k \in \{1, \dots, T_d\}$, where T_d is the total number of data sets.

Remark 2.1: (Data collection): Note that the index k in the notations $\mathbf{x}_d(k)$, $\mathbf{x}_d^+(k)$ and $\mathbf{u}_d(k)$ is simply an indicator that distinguishes one data sample from another. In fact,

for each sample set, we only require that the time interval between $\mathbf{x}_d^+(k)$ and $\mathbf{x}_d(k)$ satisfies the discretization step-size δ . Of course, it is possible that all the sampling instances of the data are chosen consecutively from a system trajectory with a fixed interval δ , which means that all the data samples are head-tail connected, i.e., $\mathbf{x}_d^+(k)$ of the former data can be used as the $\mathbf{x}_d(k)$ of the latter one. However, in general, we allow the data samples to be collected at independent time instances, and even from various trajectories of the system. \square

Problem 1: Given data samples $\mathbf{x}_d(k)$, $\mathbf{x}_d^+(k)$ and $\mathbf{u}_d(k)$, $k \in \{1, \dots, T_d\}$, identify the parameters α , W_D , B_D , and s_D of system (3).

To solve this problem, one could seek to fit the model (3) with the given data samples $\mathbf{x}_d(k)$, $\mathbf{x}_d^+(k)$ and $\mathbf{u}_d(k)$. However, due to the presence of the (non-linear, non-convex) threshold function, such approach would involve a non-convex minimization problem with a large number of variables. Motivated by this observation, we develop a more efficient approach by exploiting the specific structure of (3).

III. SCALAR OPTIMIZATION FOR PARAMETER IDENTIFICATION

Here, we reformulate the parameter identification as a scalar variable optimization problem. For $k \in \{1, \dots, T_d\}$, bringing system inputs $\mathbf{u}_d(k)$ and states $\mathbf{x}_d(k)$, $\mathbf{x}_d^+(k)$ into (3), we have

$$\mathbf{x}_d^+(k) - \alpha \mathbf{x}_d(k) = [H\mathbf{p}_d(k)]_0^{s_D} \quad (4)$$

where $\mathbf{p}_d(k) = \text{col}\{\mathbf{x}_d(k), \mathbf{u}_d(k)\}$, and

$$H = [W_D \quad B_D] = \begin{bmatrix} - & h_1^\top & - \\ - & h_2^\top & - \\ & \vdots & \\ - & h_n^\top & - \end{bmatrix} \in \mathbb{R}^{n \times (n+m)}. \quad (5)$$

Note that in (5), since the diagonal entries of W_D are zero, i.e., $h_i[i] = 0$, not all the entries of matrix H are variables that need to be parameterized for identification. To characterize this, for $i \in \{1, \dots, n\}$, define $\bar{h}_i = (h_i)_{-i} \in \mathbb{R}^{n+m-1}$, which removes the i th entry from h_i . Correspondingly, let $\bar{\mathbf{p}}_i(k) = (\mathbf{p}_d(k))_{-i}$. Let $\mathbf{h} = \text{col}\{\bar{h}_1, \bar{h}_2, \dots, \bar{h}_n\} \in \mathbb{R}^{n(n+m-1)}$ and $\mathbf{P}_d(k) = \text{diag}\{\bar{\mathbf{p}}_1^\top(k), \bar{\mathbf{p}}_2^\top(k), \dots, \bar{\mathbf{p}}_n^\top(k)\} \in \mathbb{R}^{n \times n(n+m-1)}$. Then, one can write

$$\begin{aligned} H\mathbf{p}_d(k) &= \begin{bmatrix} h_1^\top \mathbf{p}_d(k) \\ h_2^\top \mathbf{p}_d(k) \\ \vdots \\ h_n^\top \mathbf{p}_d(k) \end{bmatrix} = \begin{bmatrix} \bar{h}_1^\top \bar{\mathbf{p}}_1(k) \\ \bar{h}_2^\top \bar{\mathbf{p}}_2(k) \\ \vdots \\ \bar{h}_n^\top \bar{\mathbf{p}}_n(k) \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{p}}_1^\top(k) \bar{h}_1 \\ \bar{\mathbf{p}}_2^\top(k) \bar{h}_2 \\ \vdots \\ \bar{\mathbf{p}}_n^\top(k) \bar{h}_n \end{bmatrix} \\ &= \mathbf{P}_d(k) \mathbf{h}, \end{aligned} \quad (6)$$

where the second equality holds because $h_i[i] = 0$. All entries in \mathbf{h} are variables to be identified. To proceed, define

compact vectors/matrices:

$$\mathcal{X} = \begin{bmatrix} \mathbf{x}_d(1) \\ \mathbf{x}_d(2) \\ \vdots \\ \mathbf{x}_d(T_d) \end{bmatrix}, \quad \mathcal{X}^+ = \begin{bmatrix} \mathbf{x}_d^+(1) \\ \mathbf{x}_d^+(2) \\ \vdots \\ \mathbf{x}_d^+(T_d) \end{bmatrix}, \quad \mathcal{P} = \begin{bmatrix} \mathbf{P}_d(1) \\ \mathbf{P}_d(2) \\ \vdots \\ \mathbf{P}_d(T_d) \end{bmatrix} \quad (7)$$

such that $\mathcal{X} \in \mathbb{R}^{nT_d}$; $\mathcal{X}^+ \in \mathbb{R}^{nT_d}$; $\mathcal{P} \in \mathbb{R}^{nT_d \times n(n+m-1)}$. Then, (4) reads

$$\mathcal{X}^+ - \alpha\mathcal{X} = [\mathcal{P}\mathbf{h}]_0^{s_D} \quad (8)$$

Now, given variables $v_i \geq 0$ to be determined, let

$$f(\mathcal{X}^+ - \alpha\mathcal{X})[i] = \begin{cases} v_i & \text{if } (\mathcal{X}^+ - \alpha\mathcal{X})[i] = \max(\mathcal{X}^+ - \alpha\mathcal{X}) \\ -v_i & \text{if } (\mathcal{X}^+ - \alpha\mathcal{X})[i] = 0 \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

for $i \in \{1, \dots, nT_d\}$. Note that, with the right choice of v_i 's, one can decompose $\mathcal{P}\mathbf{h} = [\mathcal{P}\mathbf{h}]_0^{s_D} + f(\mathcal{X}^+ - \alpha\mathcal{X})$, i.e., the role of $f(\mathcal{X}^+ - \alpha\mathcal{X})$ is to *compensate* for the parts of $\mathcal{P}\mathbf{h}$ that are truncated by the threshold $[\cdot]_0^{s_D}$. Equation (8) can then be written as

$$\mathcal{X}^+ - \alpha\mathcal{X} - \mathcal{P}\mathbf{h} + f(\mathcal{X}^+ - \alpha\mathcal{X}) = 0 \quad (10)$$

To further simplify the non-linear mapping $f(\mathcal{X}^+ - \alpha\mathcal{X})$, we rewrite

$$f(\mathcal{X}^+ - \alpha\mathcal{X}) = C(\alpha)v, \quad v \geq 0 \quad (11)$$

where for any fixed α , $C(\alpha) \in \mathbb{R}^{nT_d \times d(\alpha)}$ is a matrix that can be constructed by the following two-step procedure:

- 1) Define a diagonal matrix $E(\alpha) \in \mathbb{R}^{nT_d \times nT_d}$ such that for all $i \in \{1, \dots, nT_d\}$,

$$E(\alpha)[i, i] = \begin{cases} 1 & \text{if } (\mathcal{X}^+ - \alpha\mathcal{X})[i] = \max(\mathcal{X}^+ - \alpha\mathcal{X}) \\ -1 & \text{if } (\mathcal{X}^+ - \alpha\mathcal{X})[i] = 0 \\ 0 & \text{otherwise} \end{cases} \quad (12)$$

- 2) Construct $C(\alpha)$ by removing all zero columns in $E(\alpha)$. Note that the number of columns of $C(\alpha)$, denoted $d(\alpha)$, is dependent on α . This matrix has the following properties

$$C(\alpha)^\top C(\alpha) = I_{d(\alpha)} \quad \text{and} \quad C(\alpha)C(\alpha)^\top = E(\alpha)^2. \quad (13)$$

The following result is an immediate consequence of these definitions.

Lemma 3.1: (Matrices $E(\alpha)$ and $C(\alpha)$ are piecewise constant): Given vectors $\mathcal{X}^+, \mathcal{X} \geq 0$, the matrices $E(\alpha)$ and $C(\alpha)$ are piecewise constant functions of α .

Looking at the expression (11) and the definition of $f(\mathcal{X}^+ - \alpha\mathcal{X})$ in (9), we see that the vector $v \in \mathbb{R}^{d(\alpha)}$ encodes the entries v_i 's whereas the sign in front of v_i gets encoded in the corresponding entry of $C(\alpha)$. Note that the structure of $C(\alpha)$ and the value of v depend nonlinearly on the choice of α . Substituting (11) into (10),

$$\mathcal{X}^+ - \alpha\mathcal{X} + C(\alpha)v - \mathcal{P}\mathbf{h} = 0, \quad v \geq 0 \quad (14)$$

To find α , \mathbf{h} , and v that satisfy (14), we can consider them as the critical points of the following objective function

$$\mathcal{J}(\alpha, v, \mathbf{h}) = \frac{1}{2} \|\mathcal{X}^+ - \alpha\mathcal{X} + C(\alpha)v - \mathcal{P}\mathbf{h}\|_2^2 \quad (15)$$

The minimization of (15) is subject to the constraint $v \geq 0$. The function \mathcal{J} is a non-smooth function of α , but smooth in \mathbf{h} and v . Thus, if we temporarily drop the inequality constraint $v \geq 0$, and let

$$\mathcal{Q}(\alpha) = [C(\alpha) \quad -\mathcal{P}], \quad \xi = \begin{bmatrix} v \\ \mathbf{h} \end{bmatrix}, \quad (16)$$

equation (15) turns into

$$\mathcal{J}(\alpha, \xi) = \frac{1}{2} \|\mathcal{X}^+ - \alpha\mathcal{X} + \mathcal{Q}(\alpha)\xi\|_2^2, \quad (17)$$

which is quadratic in ξ . For fixed α , its minimizer is characterized by

$$\frac{\partial \mathcal{J}(\alpha, \xi)}{\partial \xi} = \mathcal{Q}(\alpha)^\top (\mathcal{X}^+ - \alpha\mathcal{X} + \mathcal{Q}(\alpha)\xi) = 0.$$

If $\mathcal{Q}(\alpha)^\top \mathcal{Q}(\alpha)$ is non-singular, the solution takes the form

$$\begin{bmatrix} \hat{v} \\ \hat{\mathbf{h}} \end{bmatrix} = \hat{\xi} = -(\mathcal{Q}(\alpha)^\top \mathcal{Q}(\alpha))^{-1} \mathcal{Q}(\alpha)^\top (\mathcal{X}^+ - \alpha\mathcal{X}) \quad (18)$$

Substituting back into the definition of \mathcal{J} , the problem (assuming $\hat{v} \geq 0$) becomes

$$\min_{\alpha} \mathcal{J}(\alpha) = \frac{1}{2} \|M(\alpha) (\mathcal{X}^+ - \alpha\mathcal{X})\|_2^2, \quad (19)$$

where $M(\alpha) = I - \mathcal{Q}(\alpha) (\mathcal{Q}(\alpha)^\top \mathcal{Q}(\alpha))^{-1} \mathcal{Q}(\alpha)^\top$. Note that the definition of $\hat{\xi}$ implies that, for any ξ ,

$$\|M(\alpha) (\mathcal{X}^+ - \alpha\mathcal{X})\|_2^2 \leq \|\mathcal{X}^+ - \alpha\mathcal{X} + \mathcal{Q}(\alpha)\xi\|_2^2 \quad (20)$$

By comparing (17) and (19), the advantage of the latter is that the dimension of the optimization problem is reduced from $(1 + n(n+m) + d(\alpha))$ to 1. This kind of elimination of variables is referred to as separable nonlinear least squares problems [20].

Nevertheless, several challenges must be addressed. First, our derivation above requires $\mathcal{Q}(\alpha)^\top \mathcal{Q}(\alpha)$ to be non-singular, which means that $\mathcal{Q}(\alpha)$ must have full column rank. Second, to identify the parameter α in (3) using (19), we need to guarantee that the minimizer of the latter is unique. Third, our derivation above has neglected the constraint $v \geq 0$. To guarantee that the two objective functions have the same minimizer, we must show that \hat{v} in (18) satisfies $\hat{v} \geq 0$. We tackle each of these challenges next.

IV. IDENTIFICATION OF THE FIRING RATE MODEL

In this section, we address the challenges outlined in Section III regarding the reformulation of the parameter identification as the scalar optimization problem (19). This provides the basis for the design of an algorithm to identify the parameters of system (3).

A. Establishing the validity of scalar optimization

Here we show that the scalar optimization problem (19) provides a valid reformulation of the parameter identification problem. We make the following assumption.

Assumption 1: Let α^* be the true parameter of system (3). Given the measured system states $\mathbf{x}_d(k)$ and system inputs $\mathbf{u}_d(k)$, $k \in \{1, \dots, T_d\}$, the matrix $(I - E(\alpha^*)^2)(I - E(\alpha)^2)[\mathcal{X} \ \mathcal{P}]$ has full column rank for all $\alpha \in (0, 1)$ ¹.

Remark 4.1: (Validity of Assumption 1): Note that in Assumption 1, all the matrices $E(\alpha^*)$, $E(\alpha)$, \mathcal{X} and \mathcal{P} are associated with the measurement data. Specifically, \mathcal{X} and \mathcal{P} are defined from $\mathbf{u}_d(k)$ and $\mathbf{x}_d(k)$ in (7); $E(\alpha^*)$ and $E(\alpha)$ are implicitly determined by $\mathbf{x}_d(k)$ and $\mathbf{x}_d^+(k)$ in (7) and (12). Besides, the row dimension of $(I - E(\alpha^*)^2)(I - E(\alpha)^2)[\mathcal{X} \ \mathcal{P}]$ grows with the number of data samples T_d , in general, one can expect Assumption 1 to hold for sufficiently large T_d . A sufficient way of checking whether Assumption 1 holds without knowing α^* is to compute the column rank of $(I - E_1)^2(I - E_2)^2[\mathcal{X} \ \mathcal{P}]$ for all $E_1, E_2 \in \mathbf{E}$, where $\mathbf{E} = \{E(\alpha) \mid \alpha \in (0, 1)\}$ is the set of all possible $E(\alpha)$, which is a finite set. The finiteness of \mathbf{E} arises directly from the fact that $E(\alpha)$ is a diagonal matrix, and its entries can only be $\pm 1, 0$. The cardinality of \mathbf{E} is therefore at least bounded by $|\mathbf{E}| \leq 3^{nT_d}$, which grows exponentially with the dimension and the number of data sets. Actually, as we show later in the proof Theorem 4.3b, a much better bound can be obtained as $|\mathbf{E}| \leq 4nT_d + 2$, which greatly reduces the complexity of validating Assumption 1. \square

The following result establishes that the scalar optimization is a valid way of finding the parameters of the system (3).

Proposition 4.2: (Validity of scalar optimization): Under Assumption 1, the following statements hold:

- [Invertibility]* For all $\alpha \in (0, 1)$, $\mathcal{Q}(\alpha) = [C(\alpha) \ -\mathcal{P}]$ has full column rank;
- [Uniqueness of minimizer]* The objective function $\mathcal{J}(\alpha)$ in (19) has a unique minimizer $\hat{\alpha} = \alpha^*$;
- [Satisfaction of constraints]* Given the unique $\hat{\alpha}$, the remaining parameters \hat{v} and $\hat{\mathbf{h}}$ can be computed by equation (18). In addition, $\hat{v} \geq 0$ holds, and $\hat{\mathbf{h}}$ is associated with the true parameters of system (3).

B. Algorithm for parameter identification

Given our discussion in Section III and Proposition 4.2, we know that all the parameters of system (3) can be determined by solving the minimization (19). The latter is challenging given the piecewise-constant nature of $M(\alpha)$ as a function of α , which in general, makes \mathcal{J} discontinuous and nonconvex.

We start by observing that the feasible region of (19) can be refined. From (8), we know $\mathcal{X}^+ - \alpha\mathcal{X} = [\mathcal{P}\mathbf{h}]_0^{s_D} \geq 0$. Thus, given data sets \mathcal{X}^+ and \mathcal{X} , the feasible region of α

¹Since the true α^* is unknown, the condition is required to hold for all $\alpha \in (0, 1)$.

can be shrunk to $(0, \alpha_{\max}]$, where

$$\alpha_{\max} = \min \left(1, \min_i \left(\frac{\mathcal{X}^+[i]}{\mathcal{X}[i]} \right) \right)$$

for all $i \in \{1, \dots, nT_d\}$ and $\mathcal{X}[i] \neq 0$. Note that if $\alpha_{\max} = 1$, this procedure actually enlarges the feasible region of α (with respect to $(0, 1)$) by adding the point $\alpha = 1$. However, since the extra point has no impact to the result of Proposition 4.2b, it does not change the solution to the optimization problem (19).

The key idea of the algorithm proposed here to solve the optimization problem (19) is to identify the domains where $M(\alpha)$ are constant matrices. Within each domain, (19) is a quadratic optimization problem, so its solution can be directly obtained. We then compare all the solutions to get the global optimum. In order to do so, a key challenge is to determine the boundary points on $(0, \alpha_{\max}]$, so that on each domain, $M(\alpha)$ is constant. As we show next, the boundary points is linear in nT_d . Algorithm 1 presents the pseudocode.

Theorem 4.3: (Properties of Algorithm 1): The Algorithm 1 has the following properties:

- [Minimizer]* The output value $\hat{\alpha}$ is the minimizer to problem (19);
- [Complexity]* Algorithm 1 terminates in at most $2nT_d + 1$ number of iterations. The computational complexity of the algorithm is $\mathcal{O}(nT_d)^{3,34}$, where n is the number of system nodes and T_d is the number of sampled data;
- [Identification]* Suppose Assumption 1 holds. Given the $\hat{\alpha} = \alpha^*$, the variables v^* and \mathbf{h}^* can be computed as:

$$\begin{bmatrix} v^* \\ \mathbf{h}^* \end{bmatrix} = -(\mathcal{Q}(\alpha^*)^\top \mathcal{Q}(\alpha^*))^{-1} \mathcal{Q}(\alpha^*)^\top (\mathcal{X}^+ - \alpha^* \mathcal{X}) \quad (21)$$

Here, α^* is the true parameter of system (3). According to equations (5)-(6), \mathbf{h}^* can be decoded as the matrices W_D and B_D in system (3). Finally, s_D can be determined by $s_D = \max(\mathcal{X}^+ - \alpha^* \mathcal{X})$.

Remark 4.4: (Size of the data and computational complexity): We have observed, cf. Remark 4.1, that large T_d is beneficial to make Assumption 1 hold while, at the same time, according to Theorem 4.3b, increasing the computational complexity of the algorithm. Since we do not consider measurement noise, it is sufficient to consider the smallest data set that satisfies Assumption 1 for the effectiveness of Theorem 4.3, keeping the computational complexity low. However, in the presence of measurement noise, large T_d can also be beneficial for obtaining more accurate system parameters. We plan to explore further the trade-off between accuracy and algorithm computational complexity. \square

V. EXAMPLES

In this section, we present simulation results to validate the effectiveness of the proposed results.

A. Simulation with synthetic data

We consider a network with $n = 10$ nodes. The dimension of input \mathbf{u} is chosen as $m = 10$. Given the state/input

Algorithm 1: Solving the optimization problem (19).

```

1 Input  $\mathcal{X}^+$ ,  $\mathcal{X}$  and  $\mathcal{P}$ ;
2 Define  $\mathcal{S} = \mathcal{M} = \mathcal{Z} = \emptyset$ ;  $\mathcal{T} \in \{1, \dots, nT_d\}$ ;
3 Initial values:  $\psi_0 = 0$ ,  $\ell = 0$ ;
4 Initial sets:  $\mathcal{S} = \{i \mid \mathcal{X}^+[i] = \max(\mathcal{X}^+), i \in \mathcal{T}\}$ ;  $\mathcal{Z} = \{i \mid \mathcal{X}^+[i] = 0, i \in \mathcal{T}\}$ ;  $\mathcal{M} = \{i \mid i \notin \mathcal{S} \cup \mathcal{Z}, i \in \mathcal{T}\}$ ;
5 while  $\psi_\ell < \alpha_{\max}$  do
6   Find the smallest  $\hat{\psi} > \psi_\ell$ , such that  $\max_{j \in \mathcal{M}} (\mathcal{X}^+ - \hat{\psi} \mathcal{X})[j] = \max(\mathcal{X}^+ - \hat{\psi} \mathcal{X})$  or  $\min_{j \in \mathcal{M}} (\mathcal{X}^+ - \hat{\psi} \mathcal{X})[j] = 0$ ;
   // When  $\alpha$  is on different sides of  $\hat{\psi}$ , the  $C(\alpha)$ 's take different values.
7   Let  $\psi_{\ell+1} = \hat{\psi}$ ;
8   Obtain  $C_{A\ell} = C(\alpha = \frac{\psi_\ell + \psi_{\ell+1}}{2})$ ;
9   Compute  $\mathcal{Q}_{A\ell} = \begin{bmatrix} C_{A\ell} & -\mathcal{P} \end{bmatrix}$  and  $M_{A\ell} = I - \mathcal{Q}_{A\ell} (\mathcal{Q}_{A\ell}^\top \mathcal{Q}_{A\ell})^{-1} \mathcal{Q}_{A\ell}^\top$ ; //  $\mathcal{Q}_{A\ell}$  is unchanged for
    $\alpha \in (\psi_\ell, \psi_{\ell+1})$ .
10  Solve  $\hat{\alpha}_{A\ell} = \arg \min_{\alpha \in (\psi_\ell, \psi_{\ell+1})} \frac{1}{2} \|M_{A\ell} (\mathcal{X}^+ - \alpha_{A\ell} \mathcal{X})\|_2^2$ ; // Solve the optimization problem.
11  Compute  $\mathcal{J}(\hat{\alpha}_{A\ell}) = \frac{1}{2} \|M_{A\ell} (\mathcal{X}^+ - \hat{\alpha}_{A\ell} \mathcal{X})\|_2^2$ ;
12  Obtain  $C_{B\ell} = C(\alpha = \psi_{\ell+1})$ ;
13  Compute  $\mathcal{Q}_{B\ell} = \begin{bmatrix} C_{B\ell} & -\mathcal{P} \end{bmatrix}$  and  $M_{B\ell} = I - \mathcal{Q}_{B\ell} (\mathcal{Q}_{B\ell}^\top \mathcal{Q}_{B\ell})^{-1} \mathcal{Q}_{B\ell}^\top$ ;
14  Let  $\hat{\alpha}_{B\ell} = \psi_{\ell+1}$ . Compute  $\mathcal{J}(\hat{\alpha}_{B\ell}) = \frac{1}{2} \|M_{B\ell} (\mathcal{X}^+ - \hat{\alpha}_{B\ell} \mathcal{X})\|_2^2$ ;
15  Update  $\mathcal{S} = \{i \mid (\mathcal{X}^+ - \psi_{\ell+1} \mathcal{X})[i] = \max(\mathcal{X}^+ - \psi_{\ell+1} \mathcal{X}), i \in \mathcal{T}\}$ ;  $\mathcal{Z} = \{i \mid (\mathcal{X}^+ - \psi_{\ell+1} \mathcal{X})[i] = 0, i \in \mathcal{T}\}$ ;
    $\mathcal{M} = \{i \mid i \notin \mathcal{S} \cup \mathcal{Z}, i \in \mathcal{T}\}$ ; // Update sets for  $\alpha = \psi_{\ell+1}$ .
16   $\ell = \ell + 1$ ;
17 end
18 Output  $\hat{\alpha} = \arg \min_{\alpha \in \{\hat{\alpha}_{A\ell}\} \cup \{\hat{\alpha}_{B\ell}\}} \mathcal{J}(\alpha)$ 

```

dimensions of the system, we first create matrices $W_D \in \mathbb{R}^{10 \times 10}$ and $B_D \in \mathbb{R}^{10 \times 10}$. By definition, W_D is a matrix with 0 diagonal entries. For the non-zero entries of W_D , we make sure they are consistent with *Dale's law*², i.e., each column of W_D is either non-negative or non-positive depending on the excitatory or inhibitory properties of the nodes. The values of these entries are randomly chosen from $[0 \ 0.1]$ or $[-0.05 \ 0]$ with uniform distributions. For $B_D \in \mathbb{R}^{10 \times 10}$, all its entries are randomly chosen from $[-0.04 \ 0.06]$ with uniform distributions. We set $\alpha^* = 0.9$ and $s_D = 2$. Based on W_D , B_D , α and s_D , we now create data samples, for $k \in \{1, \dots, T_d\}$ and $T_d = 250$. In this simulation, for different k , $\mathbf{x}_d(k)$ and $\mathbf{u}_d(k)$ are chosen independently, i.e., the entries of $\mathbf{x}_d(k)$ are randomly chosen from $[0 \ 4]$; the entries of $\mathbf{u}_d(k)$ are randomly chosen from $[0 \ 6]$, with uniform distributions. For each pair of $\mathbf{x}_d(k)$ and $\mathbf{u}_d(k)$, we compute the $\mathbf{x}_d^+(k)$ based on the discrete-time system model (4). It is worth pointing out that the obtained data set satisfies Assumption 1 for all $\alpha \in (0, 1)$.

By running Algorithm 1, we obtain $\alpha_{\max} = 0.9160$, and the function value of $\mathcal{J}(\alpha)$ takes 0.000 at $\alpha^* = 0.9000$. This validates that Algorithm 1 is capable of finding the minimizer

²*Dale's law*: A neuron performs the same chemical action at all of its synaptic connections to others, regardless of the identity of the target cell.

of the objection function (19), which is also the true parameter of the system. Then by equation (21) and definition (5), one can obtain the matrices W_D and B_D by computing \mathbf{h}^* . Finally, we identify $s_D = \max(\mathcal{X}^+ - \alpha^* \mathcal{X}) = 2.00$.

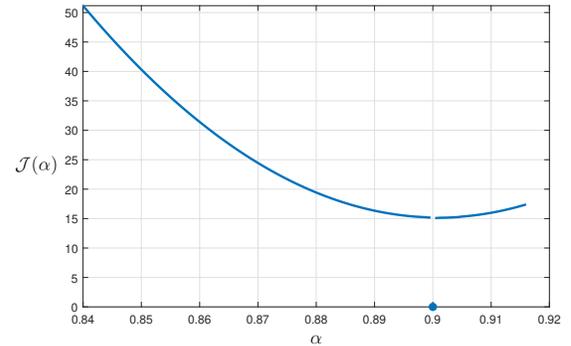


Fig. 1. Identify the system parameter α on a 10-node network.

Fig. 1 shows how \mathcal{J} changes discontinuously with α . The minimizer of the function, which is nonconvex, appears at an isolated point, which corresponds to one ψ_ℓ in Algorithm 1. Directly discretizing the feasible region $(0, \alpha_{\max}]$ for α even with a small stepsize could easily miss the isolated global minimizer.

B. Reconstructing the firing rate dynamics in rodents' brain

Here, we also apply our algorithm to a real-world example, the goal-driven attention of rodents. The data we use is from a carefully designed experimental paradigm [21], [22] that involves goal-driven selective listening in rodents. During the experiment, the rodents are given warble sounds, and the firing rates of the neuron cells are recorded in different areas of their brains. By using a classification method introduced in [23], we classify all the neuron cells into $2^3 = 8$ groups based on a combination of the following properties: region (PFC, A1); type (excitatory, inhibitory); and encoding (task relevant, irrelevant). Then we consider each class of neurons as a node of the system, and use the average firing rate of the populated neurons as the state of the node. The sampling duration in our example is 14 seconds, for $t \in [-7, 7]$ with an interval $\delta t = 0.1s$. The stimuli (warble) happens at $t = 0$.

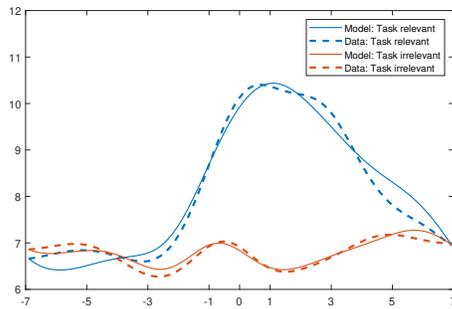


Fig. 2. Reconstructing the firing rate dynamics in rodents' brain [22].

In order to use the above introduced data set to validate our algorithm, we choose two nodes as the system states ($n = 2$), which are corresponding to the A1-inhibitory-relevant, and the A1-inhibitory-irrelevant groups of neurons. We take the readings of the other 6 nodes, along with three extra signals (i.e. system time $u_t = t$, impulse stimuli $u_s = \Delta(t)$, and a constant background activity $u_b = 1$) as system inputs. Thus, the dimension of the input is $m = 9$. After identifying the system with Algorithm 1, we use the same initial state at $x(t = -7)$, to compare the experimental data and the firing rate dynamics reconstructed by our model in Fig. 2. It can be seen that the identified linear-threshold network model is able to capture the trends of the real experimental data.

VI. CONCLUSIONS AND FUTURE WORK

We have considered the reconstruction of the firing rate dynamics in linear-threshold network models of brain activity. The study of the structure of the parameter identification problem has led us to introduce a scalar variable optimization with a piecewise smooth objective function. Our analysis of the latter has shown that its minimizer is unique and established that its solution leads to the identification of all the desired system parameters. Based on this fact, we have proposed an exhaustive algorithm that identifies the regions in the domain of definition of the objective function to find its optimizer. We have illustrated the effectiveness of the proposed algorithm in simulation. Future work will

extend the present treatment to consider data with measurement noise and investigate the trade-offs between accuracy, computational complexity, and size of datasets.

REFERENCES

- [1] W. Gerstner, A. K. Kreiter, H. Markram, and A. V. Herz, "Neural codes: firing rates and beyond," *Proceedings of the National Academy of Sciences*, vol. 94, no. 24, pp. 12 740–12 741, 1997.
- [2] E. M. Izhikevich, *Dynamical Systems in Neuroscience*. Cambridge, MA: MIT Press, 2007.
- [3] D. S. Bassett and O. Sporns, "Network neuroscience," *Nature Neuroscience*, vol. 20, no. 3, p. 353, 2017.
- [4] M. Rubinov and O. Sporns, "Complex network measures of brain connectivity: uses and interpretations," *NeuroImage*, vol. 52, no. 3, pp. 1059–1069, 2010.
- [5] Z. Hou and Z. Wang, "From model-based control to data-driven control: Survey, classification and perspective," *Information Sciences*, vol. 235, pp. 3–35, 2013.
- [6] K. J. Åström and P. Eykhoff, "System identification: a survey," *Automatica*, vol. 7, no. 2, pp. 123–162, 1971.
- [7] C. D. Persis and P. Tesi, "Formulas for data-driven control: Stabilization, optimality and robustness," *IEEE Transactions on Automatic Control*, vol. 65, no. 3, pp. 909–924, 2019.
- [8] X. Hong, R. J. Mitchell, S. Chen, C. J. Harris, K. Li, and G. W. Irwin, "Model selection approaches for non-linear system identification: a review," *International journal of systems science*, vol. 39, no. 10, pp. 925–946, 2008.
- [9] H. Hjalmarsson, S. Gunnarsson, and M. Gevers, "A convergent iterative restricted complexity control design scheme," in *IEEE Conf. on Decision and Control*, vol. 2, 1994, pp. 1735–1740.
- [10] M. G. Safonov and T. C. Tsao, "The unfalsified control concept: A direct path from experiment to controller," in *Feedback Control, Nonlinear Systems, and Complexity*. Springer, 1995, pp. 196–214.
- [11] J. C. Spall, "Multivariate stochastic approximation using a simultaneous perturbation gradient approximation," *IEEE Transactions on Automatic Control*, vol. 37, no. 3, pp. 332–341, 1992.
- [12] Z. Hou and S. Jin, "Data-driven model-free adaptive control for a class of mimo nonlinear discrete-time systems," *IEEE transactions on neural networks*, vol. 22, no. 12, pp. 2173–2188, 2011.
- [13] D. Sriharan and S. V. Sarma, "Fragility in dynamic networks: application to neural networks in the epileptic cortex," *Neural Computation*, vol. 26, no. 10, pp. 2294–2327, 2014.
- [14] D. Ehrens, D. Sriharan, and S. V. Sarma, "Closed-loop control of a fragile network: application to seizure-like dynamics of an epilepsy model," *Frontiers in Neuroscience*, vol. 9, p. 58, 2015.
- [15] P. Dayan and L. F. Abbott, *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*, ser. Computational Neuroscience. Cambridge, MA: MIT Press, 2001.
- [16] E. Nozari and J. Cortés, "Hierarchical selective recruitment in linear-threshold brain networks. Part I: Intra-layer dynamics and selective inhibition," *IEEE Transactions on Automatic Control*, vol. 66, no. 3, pp. 949–964, 2021.
- [17] A. Bakshi, R. Jayaram, and D. P. Woodruff, "Learning two layer rectified neural networks in polynomial time," in *Conference on Learning Theory*. PMLR, 2019, pp. 195–268.
- [18] F. Albertini, E. D. Sontag, and V. Maillot, "Uniqueness of weights for neural networks," *Artificial Neural Networks for Speech and Vision*, pp. 115–125, 1993.
- [19] G. Huang, "Learning capability and storage capacity of two-hidden-layer feedforward networks," *IEEE transactions on neural networks*, vol. 14, no. 2, pp. 274–281, 2003.
- [20] A. Ruhe and P. Å. Wedin, "Algorithms for separable nonlinear least squares problems," *SIAM Review*, vol. 22, no. 3, pp. 318–337, 1980.
- [21] C. C. Rodgers and M. R. DeWeese, "Neural correlates of task switching in prefrontal cortex and primary auditory cortex in a novel stimulus selection task for rodents," *Neuron*, vol. 82, no. 5, pp. 1157–1170, 2014.
- [22] —, "Spiking responses of neurons in rodent prefrontal cortex and auditory cortex during a novel stimulus selection task." CRCNS.org, 2014. [Online]. Available: <http://dx.doi.org/10.6080/KOW66HPJ>
- [23] E. Nozari and J. Cortés, "Hierarchical selective recruitment in linear-threshold brain networks. Part II: Inter-layer dynamics and top-down recruitment," *IEEE Transactions on Automatic Control*, vol. 66, no. 3, pp. 965–980, 2021.