frontiers in Neural Circuits

ORIGINAL RESEARCH
published: 22 January 2019
doi: 10.3389/fncir.2018.00119

Adaptive Stimulus Design for Dynamic Recurrent Neural Network Models

R. Ozgur Doruk[1,2]* *and Kechen Zhang*[2]

[1] *Department of Electrical and Electronic Engineering, Atilim University, Golbasi, Turkey,* [2] *Department of Biomedical Engineering, Johns Hopkins University School of Medicine, Baltimore, MD, United States*

We present an adaptive stimulus design method for efficiently estimating the parameters of a dynamic recurrent network model with interacting excitatory and inhibitory neuronal populations. Although stimuli that are optimized for model parameter estimation should, in theory, have advantages over nonadaptive random stimuli, in practice it remains unclear in what way and to what extent an optimal design of time-varying stimuli may actually improve parameter estimation for this common type of recurrent network models. Here we specified the time course of each stimulus by a Fourier series whose amplitudes and phases were determined by maximizing a utility function based on the Fisher information matrix. To facilitate the optimization process, we have derived differential equations that govern the time evolution of the gradients of the utility function with respect to the stimulus parameters. The network parameters were estimated by maximum likelihood from the spike train data generated by an inhomogeneous Poisson process from the continuous network state. The adaptive design process was repeated in a closed loop, alternating between optimal stimulus design and parameter estimation from the updated stimulus-response data. Our results confirmed that, compared with random stimuli, optimally designed stimuli elicited responses with significantly better likelihood values for parameter estimation. Furthermore, all individual parameters, including the time constants and the connection weights, were recovered more accurately by the optimal design method. We also examined how the errors of different parameter estimates were correlated, and proposed heuristic formulas to account for the correlation patterns by an approximate parameter-confounding theory. Our results suggest that although adaptive optimal stimulus design incurs considerable computational cost even for the simplest excitatory-inhibitory recurrent network model, it may potentially help save time in experiments by reducing the number of stimuli needed for network parameter estimation.

Keywords: optimal stimulus design, Fisher information matrix, excitatory-inhibitory network, inhomogeneous poisson spike train, maximum likelihood estimation, parameter confounding, Fourier series, sensory coding

OPEN ACCESS

Edited by:
Guo-Qiang Bi,
University of Science and Technology of China, China

Reviewed by:
Pengcheng Zhou,
Columbia University, United States
Ingo Fründ,
York University, Canada

****Correspondence:***
R. Ozgur Doruk
resat.doruk@atilim.edu.tr

Received: *10 August 2018*
Accepted: *17 December 2018*
Published: *22 January 2019*

Citation:
Doruk RO and Zhang K (2019) Adaptive Stimulus Design for Dynamic Recurrent Neural Network Models. Front. Neural Circuits 12:119. doi: 10.3389/fncir.2018.00119

1. INTRODUCTION

One basic problem in systems neuroscience is to understand the relationship between sensory stimulus and neural activity (Simoncelli et al., 2004; Van Hemmen and Sejnowski, 2005). Once the mathematical expression of the stimulus-response relation of a sensory neuron is chosen, all the free parameters of the model can potentially be determined by system identification, or fitting the model to a neurophysiological dataset which consists of the stimuli used and the corresponding elicited neural responses (Westwick and Kearney, 2003; Wu et al., 2006). Accurate parameter estimation requires properly chosen stimuli. For example, stimuli that elicited hardly any responses from the sensory neuron would yield unreliable parameter estimates since the responses reveal little information about the underlying neural system (Christopher deCharms et al., 1998; Wang et al., 2005). Many types of stimuli have been used for model parameter estimation, ranging from random stimuli (Marmarelis and Marmarelis, 1978; Chichilnisky, 2001; Pillow et al., 2008) to natural stimuli (Theunissen et al., 2001). The focus of this paper on automated methods of stimulus design.

In this paper we adopt an adaptive approach to stimulus design to optimize model parameter estimation for a generic excitatory-inhibitory recurrent network model. The adaptive procedure takes into account of the responses to the preceding stimuli, and generates each new stimulus adaptively by maximizing a utility function, which quantifies the usefulness of a given stimulus for model parameter estimation. This approach belongs to a widely used statistical methodology that is often called optimal experimental design, with applications in various disciplines, sometimes under other names (Atkinson and Donev, 1992; Pukelsheim, 1993; Fedorov and Leonov, 2013). The general idea of choosing stimuli adaptively has been explored by many researchers, and various methods have been proposed, with the alopex algorithm as an early example (Harth and Tzanakou, 1974; Benda et al., 2007; DiMattina and Zhang, 2013). Some adaptive methods are not model-based; that is, they do not rely on an explicit model of the stimulus-response relationship, such as the iso-response method (Bölinger and Gollisch, 2012; Gollisch and Herz, 2012; Horwitz and Hass, 2012), stimulus ensemble optimization (Machens et al., 2005), and response maximization by hill-climbing (Nelken et al., 1994; O'Connor et al., 2005) or by genetic algorithms (Bleeck et al., 2003; Yamane et al., 2008; Carlson et al., 2011; Chambers et al., 2014).

The optimal experimental design approach requires an explicit stimulus-response model, and the aim is to estimate the parameters of the given model as accurately as possible. Once all the parameters are determined, the stimulus-response model can potentially predict the responses to all possible stimuli allowed by the model. From an information-theoretic point of view, optimal experimental design based on a suitable model is expected to yield responses that are more informative about the model parameters than responses to nonadaptive random stimuli (Paninski, 2005). For generalized linear models, optimal experimental design method has been shown to be particularly efficient for its parameter estimation, with guaranteed unique solution for Gaussian stimuli (Lewi et al., 2009). Optimal design method can improve parameter estimation for hierarchical feedforward networks or multilayer perceptrons by optimizing the D-optimal metric based on the Fisher information matrix (DiMattina and Zhang, 2011). This method has been used in neurophysiological experiments where sound stimuli with multiple spectral bands were generated adaptively to optimize parameter estimation of the feedforward circuit models of the inferior colliculus (Dekel, 2012; Tam, 2012) and the auditory cortex (Feng et al., 2012; Feng, 2013). The feedforward network models used in these experiments are appropriate for stationary sound stimuli where the emphasis is on spectral features rather than temporal features. Extension to time-varying stimuli would require dynamical neural network models.

Although in this paper we focus on parameter estimation problem only, it is worth mentioning that optimal design can have different goals. One particularly useful example is the optimal design approach that facilitates the comparison of competing models because the best stimuli for model estimation may not be the best stimuli for model discrimination, and vice versa (Sugiyama and Rubens, 2008; Cavagnaro et al., 2010). For example, to compare different feedforward network models, one could combine model estimation and model comparison procedures to generate stimuli that maximally distinguishing the competing models based on the current parameter estimates (DiMattina and Zhang, 2011, 2013). This model comparison method has been successfully applied to auditory neurons recorded from the inferior colliculus in a preliminary neurophysiological experiment where the stimulus design was based on feedforward network models with different numbers of inhibitory neurons (Tam, 2012).

The neural network model considered in this paper is a simplified recurrent network based on the standard firing rate model (**Figure 1**). This type of networks has a long history (Amari, 1972; Wilson and Cowan, 1972) and has been used in various research topics (Hopfield, 1984; Beer, 1995; Ledoux and Brunel, 2011; Miller and Fumarola, 2012; Zhang, 2014; Doruk and Zhang, 2018). In particular, such networks have been used to model the neurons in the auditory system (Hancock et al., 1997; Hancock and Voigt, 1999; de la Rocha et al., 2008). Neurophysiological experiments with online adaptive stimulus design for auditory neurons in the inferior colliculus (Dekel, 2012; Tam, 2012) and the auditory cortex (Feng et al., 2012; Feng, 2013) were also based on this type of networks although only spectral information was used while temporal dynamics was ignored.

We seek to extend the optimal design approach to the dynamic recurrent networks with time-varying stimuli. The goal is to estimate the time constants and weight parameters of a given network. The optimal experiment design will be performed by maximizing a metric based on the Fisher information matrix, which is a function of both the stimulus input and the network parameters. As the true network parameters are unknown in real experiments, the Fisher Information matrix is calculated based on the estimated values of the parameters in the current step. An optimization on a time-varying stimulus is not easy and often its parametrization is required. For time-varying inputs such as the auditory stimuli, that can be done by representing the stimuli by a sum of phased cosine elements. We will model the experimental data as extracellular recording of neuronal spike trains, where the only information available is the spiking timing. This situation prevents one to apply traditional parameter estimation techniques such as minimum mean square estimation as it is more suitable for continuous rate data. We will use the likelihood function of the inhomogeneous Poisson spike model whose instantaneous rate is given by the continuous state of a

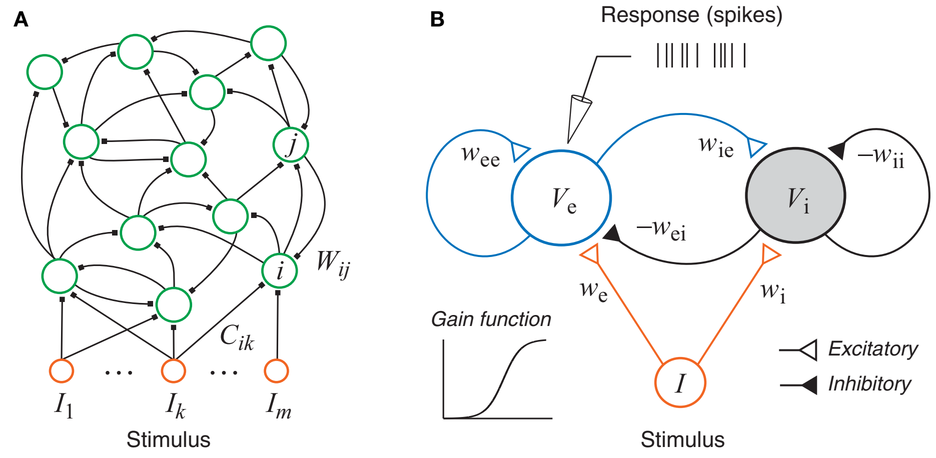

FIGURE 1 | (A) A generic recurrent network with external inputs (stimulus). **(B)** A recurrent network with one excitatory unit and one inhibitory unit. Each unit has a sigmoidal gain function and could represent a group of neurons. The responses, as simulated recordings in an experiment, are assumed to be inhomogeneous Poisson spikes based on the continuous rate generated by the state of the excitatory unit.

unit. Due to the transient dynamics, we need to derive differential equations to solve for the time courses of the derivatives of the utility function in order to speed up the optimization process. We will examine the advantages as well as the limitations of using adaptive optimal design for dynamic excitatory-inhibitory networks based on the performance of parameter estimation and the computational cost.

2. MODELS AND METHODS

2.1. Continuous Time Recurrent Neural Networks

A generic recurrent neural network may have any number of neurons, and may allow any pattern of connections within the network and any connection weights for the external inputs (**Figure 1A**). The intrinsic dynamics of each unit is much simpler than conductance-based model (Hodgkin and Huxley, 1952) and resembles the dynamics of the voltage of a passive membrane with a time constant. The output may be interpreted as the firing rate which is related to the voltage by a nonlinear gain function, and the synaptic excitation and inhibition are represented as connection weights between units. The continuous dynamical equation of this generic network can be written as:

$$\tau_{\mathrm{i}} \frac{dV_{\mathrm{i}}}{dt} = -V_{\mathrm{i}} + \sum_j W_{ij} g_j\left(V_j\right) + \sum_k C_{ik} I_k \tag{1}$$

where τ_{i} is the time constant of neuron i in the network, V_{i} is the state of neuron i (loosely interpreted as its membrane potential), W_{ij} is the weight of the synaptic connection from neuron j to neuron i, the gain function g_j is the input-output function of neuron j which transforms its membrane potential to its firing rate, and C_{ik} is the connection weight from external input I_k to neuron i. The gain function is the only source of nonlinearity in Equation (1). A popular choice of the gain function is the logistic sigmoid function given by:

$$g_j\left(V_j\right) = \frac{\Gamma_j}{1 + \exp\left(-a_j\left(V_j - h_j\right)\right)} \tag{2}$$

where Γ_j is the maximum rate at which neuron j can fire, h_j is a soft threshold parameter, and a_j is a slope parameter.

In this paper we focus on the generic excitatory-inhibitory network as shown in **Figure 1B**. This is the simplest form of the recurrent network described above with only two units, which may be interpreted either as two neurons or as two groups of neurons, one excitatory (e) and one inhibitory (i). The dynamical equations of the system can be written as:

$$\tau_{\mathrm{e}} \dot{V}_{\mathrm{e}} = -V_{\mathrm{e}} + w_{\mathrm{ee}} g_{\mathrm{e}}\left(V_{\mathrm{e}}\right) - w_{\mathrm{ei}} g_{\mathrm{i}}\left(V_{\mathrm{i}}\right) + w_{\mathrm{e}} I \tag{3}$$

$$\tau_{\mathrm{i}} \dot{V}_{\mathrm{i}} = -V_{\mathrm{i}} + w_{\mathrm{ie}} g_{\mathrm{e}}\left(V_{\mathrm{e}}\right) - w_{\mathrm{ii}} g_{\mathrm{i}}\left(V_{\mathrm{i}}\right) + w_{\mathrm{i}} I \tag{4}$$

where the subscripts e and i stands for excitatory and inhibitory neurons, respectively, and I is a single stimulus variable. One may also interpret V_{e} and V_{i} as the average activities of populations of excitatory neurons and inhibitory neurons, respectively . Note that in Equations (3) and (4), the weight parameters themselves are assumed to be positive numbers, and their signs are shown explicitly in the equations.

We can rewrite Equations (3) and (4) equivalently as a single equation:

$$\frac{d}{dt}\begin{bmatrix} V_{\mathrm{e}} \\ V_{\mathrm{i}} \end{bmatrix} = \begin{bmatrix} \beta_{\mathrm{e}} & 0 \\ 0 & \beta_{\mathrm{i}} \end{bmatrix} \left\{ -\begin{bmatrix} V_{\mathrm{e}} \\ V_{\mathrm{i}} \end{bmatrix} + \begin{bmatrix} w_{\mathrm{ee}} & -w_{\mathrm{ei}} \\ w_{\mathrm{ie}} & -w_{\mathrm{ii}} \end{bmatrix} \begin{bmatrix} g_{\mathrm{e}}\left(V_{\mathrm{e}}\right) \\ g_{\mathrm{i}}\left(V_{\mathrm{i}}\right) \end{bmatrix} + \begin{bmatrix} w_{\mathrm{e}} \\ w_{\mathrm{i}} \end{bmatrix} I \right\} \tag{5}$$

where we define

$$\beta_{\mathrm{e}} = 1/\tau_{\mathrm{e}}\ , \qquad \beta_{\mathrm{i}} = 1/\tau_{\mathrm{i}} \tag{6}$$

for easier manipulations of the equations. This vector-matrix form is more convenient for mathematical treatment in the following sections.

2.2. Inhomogeneous Poisson Spike Model and Likelihood Function

In the model described above, the final output of the excitatory neuron is a continuous firing rate: $r_{\mathrm{e}} = g_{\mathrm{e}}\left(V_{\mathrm{e}}\right)$. Since neuronal spiking activity often has Poisson-like noise (Shadlen and Newsome, 1994), for a better comparison with extracellular neurophysiological recording data, we consider spike trains generated by an inhomogeneous Poisson process, and take $r_{\mathrm{e}} = g_{\mathrm{e}}\left(V_{\mathrm{e}}\right)$ as the instantaneous firing rate (Lewis and Shedler, 1979). We write the rate as $r_{\mathrm{e}}\left(t\right)$ to emphasize its dependence on time. The probability of finding a spike in the infinitesmal time interval $\left[t, t+dt\right)$ is equal to $r_{\mathrm{e}}\left(t\right) dt$. Given a spike train $(t_1, t_2, \ldots, t_K)$ in the time interval $[0, T]$, with any finite $T > 0$, the probability density function for this spike train can be derived from the inhomogeneous Poisson process (Snyder and Miller, 1991; Brown et al., 2002; Eden, 2008), and the result reads:

$$p\left(t_1, t_2, \ldots, t_K\right) = \exp\left(-\int_0^T r_{\mathrm{e}}\left(t\right) dt\right) \prod_{k=1}^{K} r_{\mathrm{e}}\left(t_k\right) \qquad (7)$$

This probability density describes how likely a particular spike train $(t_1, t_2, \ldots, t_K)$ is generated by the inhomogeneous Poisson process with the rate function $r_{\mathrm{e}}(t)$. Of course, this rate function depends implicitly on the network parameters and the stimulus used.

2.3. Maximum Likelihood Methods and Parameter Estimation

The network parameters to be estimated are listed below as a vector:

$$\theta = [\theta_1, \ldots, \theta_8] = [\beta_{\mathrm{e}}, \beta_{\mathrm{i}}, w_{\mathrm{e}}, w_{\mathrm{i}}, w_{\mathrm{ee}}, w_{\mathrm{ei}}, w_{\mathrm{ie}}, w_{\mathrm{ii}}] \qquad (8)$$

which includes the time constants and all the connection weights in the excitatory-inhibitory network. Our maximum-likelihood estimation of the network parameters is based on the likelihood function given by Equation (7). It is well known from estimation theory that maximum likelihood estimation is asymptotically efficient; that is, it reaches the Cramér-Rao lower bound for any unbiased estimator in the limit of large data size. Sometimes parameter estimation in optimal design is based on maximum a posteriori estimation (DeGroot, 1970; Pukelsheim, 1993) which is closely linked to the maximum likelihood estimation (Myung, 2003).

It is straightforward to extend the likelihood function in Equation (7) to the situation where there are multiple spike trains elicited by a sequence of stimuli. Suppose there are M stimuli and the m-th stimulus $(m = 1, \ldots, M)$ elicits a spike train with a total of K_m spikes in the time window $[0, T]$, and the spike timings are given by $S_m = \left(t_1^{(m)}, t_2^{(m)}, \ldots, t_{K_m}^{(m)}\right)$. By (7), the likelihood function for the spike train S_m is

$$p\left(S_m \mid \theta\right) = \exp\left(-\int_0^T r_{\mathrm{e}}^{(m)}(t)\, dt\right) \prod_{k=1}^{K_m} r_{\mathrm{e}}^{(m)}\left(t_k^{(m)}\right) \qquad (9)$$

where $r_{\mathrm{e}}^{(m)}$ is the firing rate in response to the m-th stimulus. Note that the rate function $r_{\mathrm{e}}^{(m)}$ depends implicitly on the network parameters θ and on the stimulus parameters. The left-hand side of (9) emphasizes the dependence on network parameters θ, which is convenient for parameter estimation. The dependence on the stimulus parameters will be discussed in the next section.

We assume that the responses to different stimuli are independent, which is a reasonable assumption when the inter-stimulus intervals are sufficiently large so that any adaptation and plasticity can be ignored. Under this assumption, the overall likelihood function for the collection of all M spike trains can be written as

$$L(S_1, \ldots, S_M \mid \theta) = \prod_{m=1}^{M} p\left(S_m \mid \theta\right) \qquad (10)$$

By taking natural logarithm, we obtain the log likelihood function:

$$\begin{aligned} l(S_1, \ldots, S_M \mid \theta) &= \ln L(S_1, \ldots, S_M \mid \theta) \\ &= -\sum_{m=1}^{M} \int_0^T r_{\mathrm{e}}^{(m)}(t)\, dt + \sum_{m=1}^{M} \sum_{k=1}^{K_m} \ln r_{\mathrm{e}}^{(m)}\!\left(t_k^{(m)}\right) \end{aligned} \qquad (11)$$

Maximum-likelihood estimation of the parameter set is given formally by

$$\hat{\theta} = \arg\max_{\theta}\ l(S_1, \ldots, S_M \mid \theta) \qquad (12)$$

Numerical issues related to this optimization problem will be discussed in sections 2.5 and 2.6. In addition, some discussion on the local maxima problems will be provided in section 3.4.

2.4. Utility Function of Optimal Design of Stimuli

The optimal design method generates each stimulus by maximizing a utility function, which quantifies the usefulness of a given stimulus for parameter estimation based on the network model. The basic idea is to design stimuli to elicit responses that are most informative about the network parameters. In optimal design method, the utility function $U(\mathbf{x}, \theta)$ depends on the stimulus parameters $\mathbf{x}$, and typically also on the model parameters θ. An intuitive explanation of the dependence on the model parameters is best illustrated with an example. Suppose we want to estimate a Gaussian tuning curve model with unknown parameters although we may have some idea about the sensible ranges of these parameters. To estimate the height of the tuning curve accurately, we should place a probing stimulus around the likely location of the peak. To estimate the width, the probing stimulus should go to where the tuning curve is likely to have the steepest slope. For the baseline, we should go for the lowest response. This simple example illustrates the fact that an optimally designed stimulus depends on which parameter is to be estimated, and on the prior knowledge of possible parameter values. Since a scalar utility function $U(\mathbf{x}, \theta)$ depends on all the parameters $\theta = [\theta_1, \theta_2, \ldots]$, optimizing this single scalar function is sufficient to recover all the parameters, at least in theory. Indeed, when a sequence of stimuli are generated by optimal design, the stimuli may alternate spontaneously as if the optimization was performed with respect to each of the parameters one by one (DiMattina and Zhang, 2011). Alternatively, we may also optimize the parameters one by one, as described below.

Once the utility function $U(\mathbf{x},\theta)$ is chosen, the optimally designed stimulus may be written formally as:

$$\hat{\mathbf{x}} = \arg\max_{\mathbf{x}} U(\mathbf{x},\theta) \tag{13}$$

where the network parameters θ can be obtained by maximum-likelihood estimation from the existing spike data as described in the preceding section. Here the stimulus is specified by vector $\mathbf{x}$, which is a set of parameters rather than the actual stimulus itself. Direct computation of the actual time-varying stimulus is not easy because no closed analytical form of the objective function is available and furthermore the computation of the optimal control input generally requires a backward integration or recursion. Instead of struggling with this difficulty, one can restrict the stimulus I to a well known natural form such as sum of phased cosines as shown below:

$$I = \sum_{n=1}^{N} A_n \cos\left(\omega_n t + \phi_n\right) \tag{14}$$

where A_n is the amplitude, ω_n is the frequency of the n-th Fourier component, and ϕ_n is the phase of the component. We choose a base frequency ω_1 and set the frequencies of all other components as the harmonics: $\omega_n = n\omega_1$ for $n = 1, \ldots, N$. Now the stimulus parameters can be summarized by the stimulus parameter vector:

$$\mathbf{x} = [x_1, \cdots, x_{2N}] = [A_1, \cdots, A_N, \phi_1, \cdots, \phi_N] \tag{15}$$

We sometimes refer to $\mathbf{x}$ as the stimulus, with it understood that it really means a set of parameters that uniquely specify the actual stimulus I.

The Fourier representation of the stimulus in (14) is generic and flexible in the sense that given enough terms, it may potentially approximate any continuous function to arbitrary precision. The optimal design procedure should be able to automatically identify what stimuli are best for recovering the parameters from the observed network state. In general, allowing the stimuli to be time-varying is more powerful than using only stationary stimuli. For example, if we only consider the equilibrium states under stationary stimuli, we always have $\dot{V}_{\mathrm{e}} = \dot{V}_{\mathrm{i}} = 0$ so that the time constants τ_{e} and τ_{i} effectively disappear from Equations (3) and (4). In other words, the time constants cannot be recovered solely from the equilibrium states of the networks under stationary stimuli. Some transient responses to time-varying stimuli are required to recover these parameters. In this paper we use the Fourier form as a convenient choice although other representations are also possible.

Some popular choices of the objective function are based on the Fisher information matrix, which is generally defined as:

$$F_{ij}\left(\mathbf{x},\theta\right) = \left\langle \frac{\partial \ln p(r \mid \mathbf{x},\theta)}{\partial \theta_{\mathrm{i}}} \frac{\partial \ln p(r \mid \mathbf{x},\theta)}{\partial \theta_j} \right\rangle \tag{16}$$

where $p(r \mid \mathbf{x},\theta)$ is the probability distribution of the response r (number of spikes) elicited by a given stimulus $\mathbf{x}$ within a certain time window, and the average $\langle\ \rangle$ is over all possible responses $\mathbf{r}$ according to the probability distribution $p(r \mid \mathbf{x},\theta)$ for fixed $\mathbf{x}$ and θ. The Fisher information matrix reflects the amount of information contained in the noisy response $\mathbf{r}$ about the model parameters θ, assuming a generative model given by the conditional probability $p(\mathbf{r} \mid \mathbf{x},\theta)$. So the stimulus designed by maximizing a certain measure of the Fisher information matrix (16) is expected to decrease the error of the estimation of the parameters θ.

Next we give an explicit expression for the response likelihood function $p(\mathbf{r} \mid \mathbf{x},\theta)$, which is different from the spike train likelihood function such as $p\left(t_1, t_2, \ldots, t_K\right)$ in (7) and $p\left(S_m \mid \theta\right)$ in (9) although they all follow from the same inhomogeneous Poisson model. In our network model, the recorded spike train has an inhomogeneous Poisson distribution with the rate function $r_{\mathrm{e}}\left(t\right)$. We write this rate as $r_{\mathrm{e}}\left(t,\mathbf{x},\theta\right)$ to emphasize its dependence on the stimulus $\mathbf{x}$ and the network parameters θ. Assuming a small time window of duration Δt centered at time t, the response (number of spikes) obeys the Poisson distribution:

$$p(r \mid \mathbf{x},\theta) = \frac{\lambda(t,\mathbf{x},\theta)^r}{r!} \exp(-\lambda(t,\mathbf{x},\theta)) \tag{17}$$

where the mean response is given by

$$\lambda(t,\mathbf{x},\theta) = r_{\mathrm{e}}(t,\mathbf{x},\theta)\Delta t \tag{18}$$

Now the entry of the Fisher information matrix in (16) becomes

$$F_{ij}\left(t,\mathbf{x},\theta\right) = \frac{\Delta t}{r_{\mathrm{e}}\left(t,\mathbf{x},\theta\right)} \frac{\partial r_{\mathrm{e}}\left(t,\mathbf{x},\theta\right)}{\partial \theta_i} \frac{\partial r_{\mathrm{e}}\left(t,\mathbf{x},\theta\right)}{\partial \theta_j} \tag{19}$$

where t is added as a variable of F_{ij} to emphasize its dependence on time.

The utility function U can be chosen as a scalar function of the Fisher information matrix $\mathbf{F}$. A popular and theoretically well founded choice is the D-optimal design with the utility function:

$$U(\mathbf{x},\theta) = \det \mathbf{F}(\mathbf{x},\theta) \tag{20}$$

which reflects the inverse volume of the error covariance ellipsoid for all the parameters of the model. One drawback of the method is that the determinant of the Fisher information matrix is not always easy to optimize. The A-optimal design, based on the trace of the Fisher information matrix, is much easier to optimize:

$$U(\mathbf{x},\theta) = \mathrm{tr}\, \mathbf{F}(\mathbf{x},\theta) \tag{21}$$

Another alternative is the E-optimal design where the objective function is the smallest eigenvalue of the Fisher information matrix.

In this paper the A-optimality measure of the information matrix is preferred. There is an obvious reason for this preference. As the computational complexity of the optimization algorithms are expected to be high, the necessity of numerical derivative computation should be avoided as much as possible. Since it is not easy to evaluate the derivatives of the eigenvalues and determinants by any means other than numerical approximations it will be convenient to apply a criterion like A-optimality which is simply a the sum of the diagonal elements. Since the Poisson rate function r_e varies with time, the A-optimal utility function in (21) should be modified by including an integration over time:

$$U(\mathbf{x}, \theta) = \int_0^T \operatorname{tr} \mathbf{F}(t, \mathbf{x}, \theta) dt = \int_0^T \sum_{k=1}^{8} \frac{1}{r_e(t, \mathbf{x}, \theta)} \left(\frac{\partial r_e(t, \mathbf{x}, \theta)}{\partial \theta_k} \right)^2 dt \tag{22}$$

where 8 is the total number of parameters $(\theta_1, \cdots, \theta_8)$, and the time window Δt is ignored because it is a constant coefficient that does not affect the result of the optimization.

For convenience, we can also define the objective function with respect to a single parameter θ_k as:

$$U_k(\mathbf{x}, \theta) = \int_0^T \frac{1}{r_e(t, \mathbf{x}, \theta)} \left(\frac{\partial r_e(t, \mathbf{x}, \theta)}{\partial \theta_k} \right)^2 dt \tag{23}$$

The objective function in (22) is identical to

$$U(\mathbf{x}, \theta) = \sum_{k=1}^{8} U_k(\mathbf{x}, \theta) \tag{24}$$

where 8 is the total number of parameters in (8).

The optimization of the D-optimal criterion in (20) is not affected by parameter rescaling, or changing the units of parameters. For example, changing the unit of parameter θ_1 (say, from msec^{-1} to sec^{-1}) is equivalent to rescaling the parameter by a constant coefficient so that $\theta_1 \rightarrow c\theta_1$. The effect of this transformation is equivalent to a rescaling of the determinant of the Fisher information matrix by a constant factor, namely, $\det \mathbf{F} \rightarrow (\det \mathbf{F})/c^{16}$, which does not affect the location of the maximum of (20). By contrast, the criterion function in (21) or (22) are affected by parameter rescaling. A parameter with a smaller unit would tend to have larger derivative value and therefore contribute more to (22) than a parameter with a large unit.

To alleviate this scaling problem, we use U_k one by one to generate the stimuli. That is, stimulus 1 is generated by maximizing U_1, and stimulus 2 is generated by maximizing U_2, and so on. Once the 8th stimulus is generated by maximizing U_8, we go back and use U_1 to generate the next stimulus, and so on. Finally, an alternative way to get rid of scale dependence is to introduce logarithm and use $U = \sum_k \ln U_k$ as the criterion, which, however, may become degenerate when U_k approaches 0.

2.5. Gradient Computation by Solving Differential Equations

Gradient computation often helps speed up optimization algorithms. The main difficulty of computing gradients in a dynamic network model is the lack of closed form expressions like in the case of static nonlinear mapping given by a multilayer perceptron. Although gradient computations can be easily performed numerically on an explicit function by testing its values in a small neighborhood, this approach is not suitable for our implicit functions because our variables are all governed by differential equations and the increments in perturbation may not be compatible with the differential equations. One feasible remedy is to directly solve for the gradients in self-contained differential equations derived from the original equations (Flila et al., 2010; Telen et al., 2012a,b). We will adapt this method for our excitatory-inhibitory network.

We have two optimization problems in this paper, namely, optimizing the utility function for stimulus design, and optimizing the likelihood function for parameter estimation. To compute the gradients in these two cases, we need to first evaluate the gradients of the network state variable with respect to either the stimulus parameters or the network parameters. We will derive the differential equations satisfied by these gradients by taking derivatives on both sides the original dynamical Equation (5) .

First we consider the gradients with respect to the stimulus parameters, which include the amplitudes A_n and the phases ϕ_n of the Fourier series in Equation (14). We write the stimulus parameter as a vector:

$$\mathbf{x} = [x_1, x_2, \ldots, x_{2N}] = [A_1, \ldots, A_N, \phi_1, \ldots, \phi_N] \tag{25}$$

We write the state of the network also as a vector: $\mathbf{v} = [V_e, V_i]^T$. We take derivatives such as $\partial\mathbf{v}/\partial\mathbf{x}$ as an independent variable, and solve it directly from the differential equation derived from the original dynamical Equation (5). Taking derivative with respect to $\mathbf{x}$ on both sides of Equation (5), we obtain the desired differential equation:

$$\begin{aligned} \frac{d}{dt}\frac{\partial \mathbf{v}}{\partial \mathbf{x}} &= \begin{bmatrix} \beta_e & 0 \\ 0 & \beta_i \end{bmatrix} \left\{ -\frac{\partial \mathbf{v}}{\partial \mathbf{x}} \right. \\ &\left. + \begin{bmatrix} w_{ee} & -w_{ei} \\ w_{ie} & -w_{ii} \end{bmatrix} \begin{bmatrix} g_e'(V_e) & 0 \\ 0 & g_i'(V_i) \end{bmatrix} \frac{\partial \mathbf{v}}{\partial \mathbf{x}} + \begin{bmatrix} w_e \\ w_i \end{bmatrix} \frac{\partial I}{\partial \mathbf{x}} \right\} \end{aligned} \tag{26}$$

where g_e' and g_i' are the derivatives of the gain functions g_e and g_i, respectively, and the matrices derivatives are defined in the usual manner:

$$\frac{\partial I}{\partial \mathbf{x}} = \left[\frac{\partial I}{\partial A_1} \cdots \frac{\partial I}{\partial A_N}, \frac{\partial I}{\partial \phi_1} \cdots \frac{\partial I}{\partial \phi_N} \right] \tag{27}$$

and

$$\frac{\partial \mathbf{v}}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial V_e}{\partial A_1} & \cdots & \frac{\partial V_e}{\partial A_N} & \frac{\partial V_e}{\partial \phi_1} & \cdots & \frac{\partial V_e}{\partial \phi_N} \\ \frac{\partial V_i}{\partial A_1} & \cdots & \frac{\partial V_i}{\partial A_N} & \frac{\partial V_i}{\partial \phi_1} & \cdots & \frac{\partial V_i}{\partial \phi_N} \end{bmatrix} \tag{28}$$

Equation (26) can be written equivalently in the shorthand form:

$$\frac{d}{dt}\frac{\partial \mathbf{v}}{\partial \mathbf{x}} = \mathbf{B} \left\{ -\frac{\partial \mathbf{v}}{\partial \mathbf{x}} + \mathbf{WG}\frac{\partial \mathbf{v}}{\partial \mathbf{x}} + \mathbf{w}\frac{\partial I}{\partial \mathbf{x}} \right\} \tag{29}$$

where $\mathbf{B} = \begin{bmatrix} \beta_{\mathrm{e}} & 0 \\ 0 & \beta_{\mathrm{i}} \end{bmatrix}$, $\mathbf{W} = \begin{bmatrix} w_{\mathrm{ee}} & -w_{\mathrm{ei}} \\ w_{\mathrm{ie}} & -w_{\mathrm{ii}} \end{bmatrix}$, $\mathbf{G} = \begin{bmatrix} g_{\mathrm{e}}'(V_{\mathrm{e}}) & 0 \\ 0 & g_{\mathrm{i}}'(V_{\mathrm{i}}) \end{bmatrix}$, and $\mathbf{w} = \begin{bmatrix} w_{\mathrm{e}} \\ w_{\mathrm{i}} \end{bmatrix}$.

Next we consider the gradients with respect to the network parameters: $\theta = [\theta_1, \ldots, \theta_8] = [\beta_{\mathrm{e}}, \beta_{\mathrm{i}}, w_{\mathrm{e}}, w_{\mathrm{i}}, w_{\mathrm{ee}}, w_{\mathrm{ei}}, w_{\mathrm{ie}}, w_{\mathrm{ii}}]$. By taking derivative with respect to $\theta_{\mathbf{k}}$ on both sides of Equation (5), we obtain

$$\frac{d}{dt}\frac{\partial \mathbf{v}}{\partial \theta_k} = \mathbf{B}\left\{-\frac{\partial \mathbf{v}}{\partial \theta_k} + \mathbf{WG}\frac{\partial \mathbf{v}}{\partial \theta_k}\right\} + \mathbf{z}_k \tag{30}$$

where $\dfrac{\partial \mathbf{v}}{\partial \theta_k} = \left[\dfrac{\partial V_{\mathrm{e}}}{\partial \theta_k}, \dfrac{\partial V_{\mathrm{i}}}{\partial \theta_k}\right]^{\mathrm{T}}$ and the last term $\mathbf{z}_k$ refers to the extra components resulting from the chain rule of differentiation. These extra terms are presented in **Table 1**.

We also need to consider the second-order cross derivatives as needed for maximizing the trace of the Fisher information matrix. Taking derivative of (29) with respect to θ_k, we find:

$$\frac{d}{dt}\frac{\partial^2 \mathbf{v}}{\partial \mathbf{x}\partial \theta_k} = \mathbf{B}\left\{-\frac{\partial^2 \mathbf{v}}{\partial \mathbf{x}\partial \theta_k} + \mathbf{WG}\frac{\partial^2 \mathbf{v}}{\partial \mathbf{x}\partial \theta_k} + \mathbf{WG}'\,\mathrm{diag}\left(\frac{\partial \mathbf{v}}{\partial \theta_k}\right)\frac{\partial \mathbf{v}}{\partial \mathbf{x}}\right\} + \mathbf{Z}_k \tag{31}$$

where $\mathbf{G}' = \begin{bmatrix} g_{\mathrm{e}}''(V_{\mathrm{e}}) & 0 \\ 0 & g_{\mathrm{i}}''(V_{\mathrm{i}}) \end{bmatrix}$, $\mathrm{diag}\left(\dfrac{\partial \mathbf{v}}{\partial \theta_k}\right) = \begin{bmatrix} \frac{\partial V_{\mathrm{e}}}{\partial \theta_k} & 0 \\ 0 & \frac{\partial V_{\mathrm{i}}}{\partial \theta_k} \end{bmatrix}$, and

$$\frac{\partial^2 \mathbf{v}}{\partial \mathbf{x}\partial \theta_k} = \begin{bmatrix} \dfrac{\partial^2 V_{\mathrm{e}}}{\partial A_1 \partial \theta_k} & \cdots & \dfrac{\partial^2 V_{\mathrm{e}}}{\partial A_N \partial \theta_k} & \dfrac{\partial^2 V_{\mathrm{e}}}{\partial \phi_1 \partial \theta_k} & \cdots & \dfrac{\partial^2 V_{\mathrm{e}}}{\partial \phi_N \partial \theta_k} \\[2ex] \dfrac{\partial^2 V_{\mathrm{i}}}{\partial A_1 \partial \theta_k} & \cdots & \dfrac{\partial^2 V_{\mathrm{i}}}{\partial A_N \partial \theta_k} & \dfrac{\partial^2 V_{\mathrm{i}}}{\partial \phi_1 \partial \theta_k} & \cdots & \dfrac{\partial^2 V_{\mathrm{i}}}{\partial \phi_N \partial \theta_k} \end{bmatrix} \tag{32}$$

which is compatible with (28). The last term $\mathbf{Z}_k$ is specified in **Table 2**.

Now we are ready to evaluate the derivatives of the mean firing rate $r_{\mathrm{e}} = g_{\mathrm{e}}(V_{\mathrm{e}})$ with respect to the network parameters θ_k in (8) and the stimulus parameters x_j in (25). The first and the second order derivatives are:

$$\frac{\partial r_{\mathrm{e}}}{\partial \theta_k} = g_{\mathrm{e}}'(V_{\mathrm{e}})\frac{\partial V_{\mathrm{e}}}{\partial \theta_k} \tag{33}$$

$$\frac{\partial r_{\mathrm{e}}}{\partial x_l} = g_{\mathrm{e}}'(V_{\mathrm{e}})\frac{\partial V_{\mathrm{e}}}{\partial x_l} \tag{34}$$

$$\frac{\partial^2 r_{\mathrm{e}}}{\partial x_l \partial \theta_k} = g_{\mathrm{e}}''(V_{\mathrm{e}})\frac{\partial V_{\mathrm{e}}}{\partial x_l}\frac{\partial V_{\mathrm{e}}}{\partial \theta_k} + g_{\mathrm{e}}'(V_{\mathrm{e}})\frac{\partial^2 V_{\mathrm{e}}}{\partial x_l \partial \theta_k} \tag{35}$$

TABLE 1 | The extra components z_k in Equation (30).

k	**Parameter θ_k**	**Extra term z_k in Equation (30)**
1	β_{e}	$\begin{bmatrix}1&0\\0&0\end{bmatrix}\left\{-\begin{bmatrix}V_{\mathrm{e}}\\V_{\mathrm{i}}\end{bmatrix}+\mathbf{w}\begin{bmatrix}g_{\mathrm{e}}(V_{\mathrm{e}})\\g_{\mathrm{i}}(V_{\mathrm{i}})\end{bmatrix}+wI\right\}$
2	β_{i}	$\begin{bmatrix}0&0\\0&1\end{bmatrix}\left\{-\begin{bmatrix}V_{\mathrm{e}}\\V_{\mathrm{i}}\end{bmatrix}+\mathbf{w}\begin{bmatrix}g_{\mathrm{e}}(V_{\mathrm{e}})\\g_{\mathrm{i}}(V_{\mathrm{i}})\end{bmatrix}+wI\right\}$
3	w_{e}	$\mathbf{B}\begin{bmatrix}1\\0\end{bmatrix}I$
4	w_{i}	$\mathbf{B}\begin{bmatrix}0\\1\end{bmatrix}I$
5	w_{ee}	$\mathbf{B}\begin{bmatrix}1&0\\0&0\end{bmatrix}\begin{bmatrix}g_{\mathrm{e}}(V_{\mathrm{e}})\\g_{\mathrm{i}}(V_{\mathrm{i}})\end{bmatrix}$
6	w_{ei}	$\mathbf{B}\begin{bmatrix}0&-1\\0&0\end{bmatrix}\begin{bmatrix}g_{\mathrm{e}}(V_{\mathrm{e}})\\g_{\mathrm{i}}(V_{\mathrm{i}})\end{bmatrix}$
7	w_{ie}	$\mathbf{B}\begin{bmatrix}0&0\\1&0\end{bmatrix}\begin{bmatrix}g_{\mathrm{e}}(V_{\mathrm{e}})\\g_{\mathrm{i}}(V_{\mathrm{i}})\end{bmatrix}$
8	w_{ii}	$\mathbf{B}\begin{bmatrix}0&0\\0&-1\end{bmatrix}\begin{bmatrix}g_{\mathrm{e}}(V_{\mathrm{e}})\\g_{\mathrm{i}}(V_{\mathrm{i}})\end{bmatrix}$

TABLE 2 | The extra components Z_k in Equation (31).

k	**Parameter θ_k**	**Extra term Z_k in Equation (31)**
1	β_{e}	$\begin{bmatrix}1&0\\0&0\end{bmatrix}\left\{-\frac{\partial \boldsymbol{v}}{\partial \boldsymbol{x}}+\mathbf{WG}\frac{\partial \boldsymbol{v}}{\partial \boldsymbol{x}}+\boldsymbol{w}\frac{\partial I}{\partial \boldsymbol{x}}\right\}$
1	β_{i}	$\begin{bmatrix}0&0\\0&1\end{bmatrix}\left\{-\frac{\partial \boldsymbol{v}}{\partial \boldsymbol{x}}+\mathbf{WG}\frac{\partial \boldsymbol{v}}{\partial \boldsymbol{x}}+\boldsymbol{w}\frac{\partial I}{\partial \boldsymbol{x}}\right\}$
3	w_{e}	$\mathbf{B}\begin{bmatrix}1\\0\end{bmatrix}\frac{\partial I}{\partial \boldsymbol{x}}$
4	w_{i}	$\mathbf{B}\begin{bmatrix}0\\1\end{bmatrix}\frac{\partial I}{\partial \boldsymbol{x}}$
5	w_{ee}	$\mathbf{B}\begin{bmatrix}1&0\\0&0\end{bmatrix}\boldsymbol{G}\frac{\partial \boldsymbol{v}}{\partial \boldsymbol{x}}$
6	w_{ei}	$\mathbf{B}\begin{bmatrix}0&-1\\0&0\end{bmatrix}\boldsymbol{G}\frac{\partial \boldsymbol{v}}{\partial \boldsymbol{x}}$
7	w_{ie}	$\mathbf{B}\begin{bmatrix}0&0\\1&0\end{bmatrix}\boldsymbol{G}\frac{\partial \boldsymbol{v}}{\partial \boldsymbol{x}}$
8	w_{ii}	$\mathbf{B}\begin{bmatrix}0&0\\0&-1\end{bmatrix}\boldsymbol{G}\frac{\partial \boldsymbol{v}}{\partial \boldsymbol{x}}$

These formulas are expressed in terms of the derivatives $\frac{\partial V_{\mathrm{e}}}{\partial \theta_k}$, $\frac{\partial V_{\mathrm{e}}}{\partial x_l}$ and $\frac{\partial^2 V_{\mathrm{e}}}{\partial x_l \partial \theta_k}$, which are regarded as independent dynamical variables that can be solved from the three differential Equations (29)–(31). In our simulations, the initial conditions were always assumed to be the equilibrium state, and the initial values of the derivatives were set to zero.

Next, we evaluate the gradient of the utility function in Equation (22) with respect to the stimulus parameters x_l as follows:

$$\frac{\partial U}{\partial x_l} = \int_0^T \frac{\partial}{\partial x_l} \sum_{k=1}^{8} \frac{1}{r_{\mathrm{e}}} \left(\frac{\partial r_{\mathrm{e}}}{\partial \theta_k} \right)^2 dt \tag{36}$$

$$= \int_0^T \sum_{k=1}^{8} \left\{ -\frac{1}{r_{\mathrm{e}}^2} \frac{\partial r_{\mathrm{e}}}{\partial x_l} \left(\frac{\partial r_{\mathrm{e}}}{\partial \theta_k} \right)^2 + \frac{2}{r_{\mathrm{e}}} \frac{\partial r_{\mathrm{e}}}{\partial \theta_k} \frac{\partial^2 r_{\mathrm{e}}}{\partial x_l \partial \theta_k} \right\} dt \tag{37}$$

where the last expression is written in terms of the derivatives that are already evaluated by Equations (33)–(35). If we design the stimulus with respect to one parameter by optimizing the utility function in Equation (23), we rewrite the above by removing the summation and obtain:

$$\frac{\partial U_k}{\partial x_l} = \int_0^T \frac{\partial}{\partial x_l} \frac{1}{r_{\mathrm{e}}} \left(\frac{\partial r_{\mathrm{e}}}{\partial \theta_k} \right)^2 dt \tag{38}$$

$$= \int_0^T \left\{ -\frac{1}{r_{\mathrm{e}}^2} \frac{\partial r_{\mathrm{e}}}{\partial x_l} \left(\frac{\partial r_{\mathrm{e}}}{\partial \theta_k} \right)^2 + \frac{2}{r_{\mathrm{e}}} \frac{\partial r_{\mathrm{e}}}{\partial \theta_k} \frac{\partial^2 r_{\mathrm{e}}}{\partial x_l \partial \theta_k} \right\} dt \tag{39}$$

Lastly, for maximum likelihood estimation, one needs the gradient of the log likelihood function of spike trains in (11):

$$\frac{\partial l}{\partial \theta_k} = -\sum_{m=1}^{M} \int_0^T \frac{\partial r_{\mathrm{e}}^{(m)}(t)}{\partial \theta_k} dt + \sum_{m=1}^{M} \sum_{k=1}^{K_m} \frac{1}{r_{\mathrm{e}}^{(m)}\big(t_k^{(m)}\big)} \frac{\partial r_{\mathrm{e}}^{(m)}\big(t_k^{(m)}\big)}{\partial \theta_k} \tag{40}$$

The right-hand side is already written in terms of derivatives that can be evaluated by (33).

2.6. Practical Numerical Issues Related to Optimization

As described in the preceding sections, our optimal design method requires solving two separate maximization problems as given by Equations (12) and (13). Since the goal of this paper is not to develop or implement optimization algorithms, we used the optimization programs available in MATLAB (R2013b) for all the simulation results reported in the paper. After comparison with several MATLAB optimization programs including genetic algorithms, simulated annealing and pattern search, we found that the function *fmincon*, with the default interior-point method of constrained nonlinear optimization, performed adequately for our problems.

The function *fmincon* and similar algorithms are local optimizers. In order to find a good optimum, the algorithms are often repeated with multiple initial guesses and the best one is chosen according to the value of the objective and gradient value at the termination point. This is especially important in the optimal design part as the utility function in Equation (23) may have lots of local maxima. The local maxima problem also exists for the likelihood function (11), but to a lesser extent. This is because as the number of stimuli increases, the likelihood function tends to converge to the same optimum for different initial guesses. See section 3.4 for a detailed discussion of this issue.

The local optimization algorithms such as *fmincon* need the gradient of the objective function. Although the gradient could be computed numerically, it is better to use the derivatives obtained directly by solving differential equations as explained in the preceding section. This is because numerical gradient computation is much slower and also increases the risk of singularities in the solution.

Finally, MATLAB parallel computation toolbox can speed up the optimization. We obtained a speedup by a factor of 5 even on a PC with a single Intel i7 six-core processor.

2.7. Procedural Information

In this section, we will summarize the overall procedure to show how the optimal design and parameter estimation algorithm works together in an automated loop. The first issue is the choice of the initial parameter values. In the beginning of an experiment (or simulation in our case), one usually has no idea about the true values of the network parameter vector θ, although one may have some prior information about plausible or reasonable ranges of acceptable parameter values. Since the optimal design method needs a current parameter estimate, one can assign a randomly chosen initial parameter vector from some prior distribution. In our simulations, we simply drew each parameter randomly from a uniform distribution between a lower bound and an upper bound of acceptable parameter values. One can find the details about the parametric bounds in **Table 3**. The stimulus amplitude coefficients [A_n in (14)] also need to be bounded because in the real world the total energy of the sound stimulus is finite. In simulations, very large amplitudes may lead to unrealistically large responses and instabilities which may break the optimal design procedures. An upper bound on A_n's can be found by a few initial simulations, and if no instabilities are detected one will be fine with the decided value of the bound.

The overall process of the algorithm is summarized below:

1. Set $i = 1$ (iteration count)
2. Set the current estimate $\hat{\theta}$ as a random vector between $\theta_{\min}$ and $\theta_{\max}$ (from **Table 3**)
3. Set $k = 1$ (parameter count)
4. Optimize $U_k(\mathbf{x}, \hat{\theta})$ (as in Equation 23) to generate stimulus $\hat{\mathbf{x}}$, which elicits a new response.
5. Update the current maximum likelihood estimate, $\hat{\theta}$, by including the new response data.
6. Set $k \to k + 1$. If $k > 8$ (total number of parameters) set $i \to i + 1$ and go to the next step, otherwise go to **Step 4** .
7. If $i > N_{\mathrm{itr}}$ stop and report the result as $\hat{\theta}$, otherwise go to **Step 3**

Here N_{itr} is the total number of iterations. The total number of stimuli generated by the procedure is equal to $8N_{\mathrm{itr}}$. The optimal design results in the next section were all obtained by this procedure.

TABLE 3 | True values, bounds, and estimates of network parameters.

Parameter	True value θ	Lower bound $\theta_{\min}$	Upper bound $\theta_{\max}$	Estimate $\hat{\theta}_{\text{opt}}$ (mean ± std)	Estimate $\hat{\theta}_{\text{rand}}$ (mean ± std)
β_{e}	50	0	100	49.94 ± 0.86	50.07 ± 1.63
β_{i}	25	0	100	25.18 ± 1.44	25.23 ± 1.96
w_{e}	1	0	2	1.002 ± 0.036	0.998 ± 0.039
w_{i}	0.7	0	2	0.698 ± 0.068	0.715 ± 0.100
w_{ee}	1.2	0	3	1.227 ± 0.069	1.247 ± 0.107
w_{ei}	2	0	3	2.069 ± 0.187	2.071 ± 0.227
w_{ie}	0.7	0	3	0.712 ± 0.095	0.784 ± 0.171
w_{ii}	0.4	0	3	0.456 ± 0.199	0.569 ± 0.374

The true values of the 8 parameters in the excitatory-inhibitory network in Equation (5) are shown, followed by the lower and upper bounds used during the maximum-likelihood optimization process. The means and the standard deviations of the maximum-likelihood estimates $\hat{\theta}_{opt}$ and $\hat{\theta}_{rand}$ were obtained from 100 repeated trials, where each trial had 120 optimally designed stimuli and 120 random stimuli, respectively.

3. SIMULATION RESULTS

In this section, we will summarize our computer simulation results based on the adaptive stimulus design and network parameter estimation algorithms as described in the preceding section. Besides the properties of the optimally designed stimuli and the accuracy of the parameter estimation, we will also examine how the errors of different parameter estimates are correlated, and how the correlations might be accounted for by parameter confounding formulas derived from the network dynamical equations.

3.1. Details of the Example Problem

All simulations of the excitatory-inhibitory network (**Figure 1B** and Equation 5) were based on the generating values (sometimes also referred to as "true values") of the parameters given in **Table 3**. The gain functions $g_{\text{e}}(V_{\text{e}})$ and $g_{\text{i}}(V_{\text{i}})$ require additional parameters, namely, $\Gamma_{\text{e}}, a_{\text{e}}, h_{\text{e}}, \Gamma_{\text{i}}, a_{\text{i}}$, and h_{i}. These gain functions are given by Equation (2) except that the subscript j should be replaced by either 'e' or 'i'. Because this research was aimed at the estimation of the network parameters only, the gain functions, representing the input-output properties of individual units, were assumed to be fixed with the parameter values $\Gamma_{\text{e}} = 100, a_{\text{e}} = 0.04, h_{\text{e}} = 70, \Gamma_{\text{i}} = 50, a_{\text{i}} = 0.04$, and $h_{\text{i}} = 35$. This assumption simplifies the parameter estimation problem because now we can focus on network parameter estimation only. This assumption is not absolutely required because our method could be readily extended to estimate these parameters as well. The simplification here is equivalent to assuming that we already know the input-output relations of each individual unit. In realistic situation, we may use the average input-output relationship of each neuronal type if it is known because each unit in the model could correspond to a subpopulation of neurons as describe before.

This set of parameters allows the network to have a unique equilibrium state for each stationary input. For a square wave stimulus (Figure 2A), the resultant excitatory and inhibitory neural membrane potential responses, namely, $V_{\text{e}}(t)$ and $V_{\text{i}}(t)$, show both a transient component (initial peak) and a sustained component (later plateau) (Figures 2B,C). The peak response of the inhibitory unit rises more slowly than the excitatory unit. The peak of the excitatory unit is more pronounced, which is especially clear in the firing rate $r_{\text{e}}(t) = g_{\text{e}}(V_{\text{e}}(t))$ as shown in Figure 2D.

The optimization of the stimuli requires that the maximum power level in a single stimulus is bounded. This is a precaution to protect the model from potential instabilities due to over-stimulation. In real experiments, the stimuli should also have bounded maximum energy and subject will also need to be protected from over-stimulations. Consider the Fourier series for the stimulus in (14). As the amplitude parameter is assumed to be nonnegative ($A_n \geq 0$), assigning an upper bound defined as $A_{\max}$ should be enough. This is applied to all stimulus amplitudes A_n. In this research, a fixed setting of $A_{\max} = 120$ is chosen. The lower bound is obviously $A_{\min} = 0$. The phase parameter ϕ_n is allowed to vary freely without any lower or upper bounds as the cosine function itself is already bounded ($-1 \leq \cos \leq 1$). The frequency ω_n of the stimulus component is the n-th harmonics of a base frequency f_{base}; that is, $\omega_n = 2\pi n f_{base}$. Since we assumed a simulation time of $T_{opt} = 3$ s, we made a reasonable choice of the base frequency as $f_{base} = \frac{10}{3}$ Hz or 3.33 Hz. So we have chosen an integer relationship between the stimulation frequency and simulation time. The number of stimulus components N is chosen as $N = 5$ which is found to be reasonable concerning speed and performance balance.

Optimization algorithms such as *fmincon* requires an initial guess of the optimum solution. To make the initial choice uninformative, we picked each initial value randomly with a uniform distribution between the lower bound and the upper bound. In the optimization of stimuli, the initial amplitudes were uniformly distributed between $[0, A_{\max}]$ and phases were uniformly distributed between $[-\pi, \pi)$. Although we did not have any constraints on the phase parameter, we limited the initial phase values to a safe assumed range. We follow a similar strategy for the network parameter estimation based on maximum likelihood method. The multiple initial guesses were chosen randomly from a set of values uniformly distributed between the lower and upper bounds defined in **Table 3**.

In section 2.3, one recall from (10) or (11) that the likelihood estimation should produce better results when the number of spike train samples, M, increases. Because of this fact, the likelihood function should always be evaluated with all the

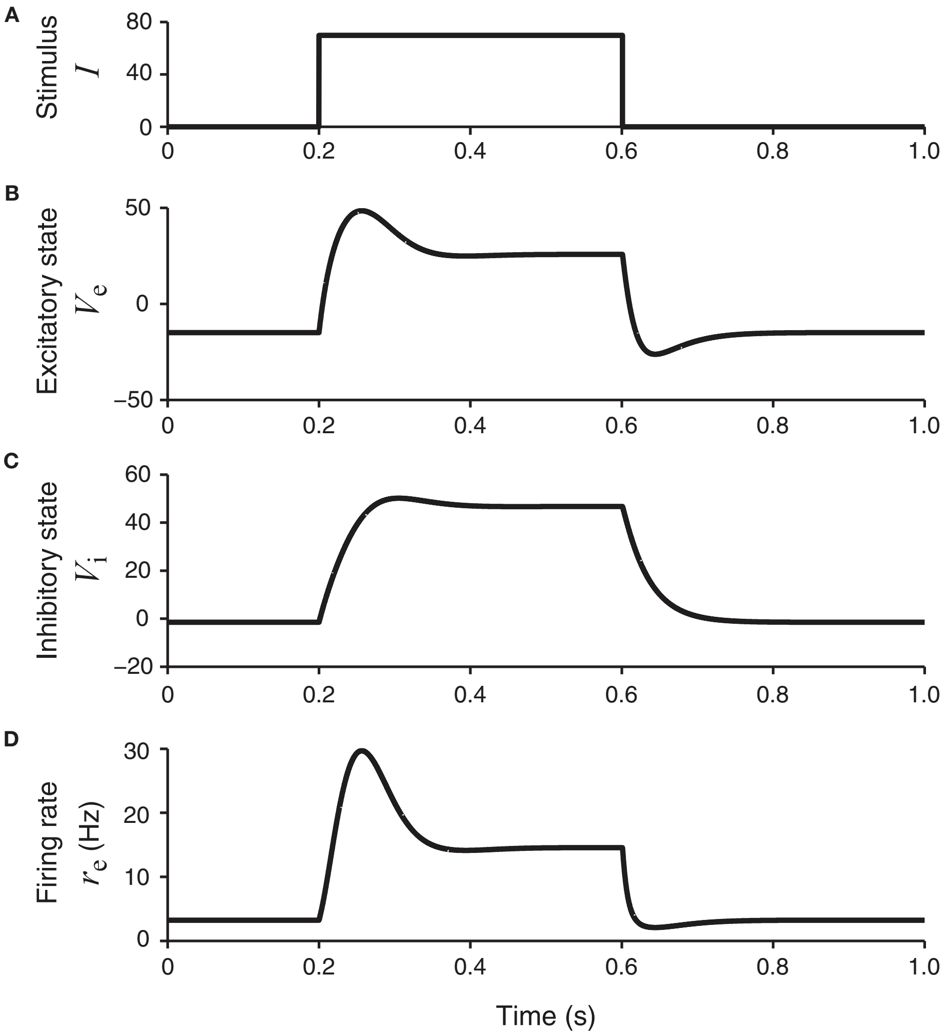

FIGURE 2 | The network model in **Figure 1B** in response to a square-wave stimulus. The states of the excitatory and inhibitory units, V_e and V_i, are shown, together with the continuous firing rate of the excitatory unit, $r_e = g_e(V_e)$. The firing rate of the excitatory unit (bottom panel) has a transient component with higher firing rates, followed by a plateau or sustained component with lower firing rates.

available spike trains generated since the beginning of the simulated trial. The total number of spike trains, M, is always equal to the total number of stimuli. If the optimal design process is iterated N_{itr} times, one will have M stimuli with $M = 8N_{\text{itr}}$ due to the fact that each iteration has 8 optimal designs sub-steps for each of the eight parameters in (8) (see section 2.4). In our simulations, we typically used 15 iterations $N_{\text{itr}} = 15$, so that we had $M = 120$ stimuli and the likelihood in (11) had 120 samples. This also means that optimal design and subsequent parameter estimation were iterated 120 times for each simulated trial. To obtain the statistics of the estimates, the trial was repeated 100 times for each case.

3.2. Statistics of Optimally Designed Stimuli

With all the necessary information from section 3.1, one can perform an optimal design experiment where stimuli are generated adaptively to elicit neural responses. An example of an optimally design stimulus, together with the elicited responses, is shown in **Figure 3**. It is noted that the optimally designed stimulus in the top panel of **Figure 3** is periodic in time because it is modeled by the Fourier series in (14).

In addition to the network responses, the bottom half of **Figure 3** shows the time courses of the parametric sensitivity derivatives $\frac{\partial r_e}{\partial \theta_i}$ which are generated by integrating (30) and then substituting to (33). The sensitivity derivatives varied greatly in their magnitudes or maximum absolute values. In **Figure 3**, the magnitudes of the derivatives with respect to the weight parameters (w_e, w_{ei}, etc.) were more than 10 times that of the time parameters (β_e and β_i). In particular, the maximum value of $\frac{\partial r_e}{\partial w_e}$ was over 100 times greater than that of $\frac{\partial r_e}{\partial \beta_e}$. This difference supports the idea that optimizing the Fisher Information Metric in (23) should be performed with respect to each single parameter separately as described at the end of section 2.4. Otherwise, if one only optimizes the sum in (24) as a whole, then a term with a low-sensitivity parameter could be easily overwhelmed by the

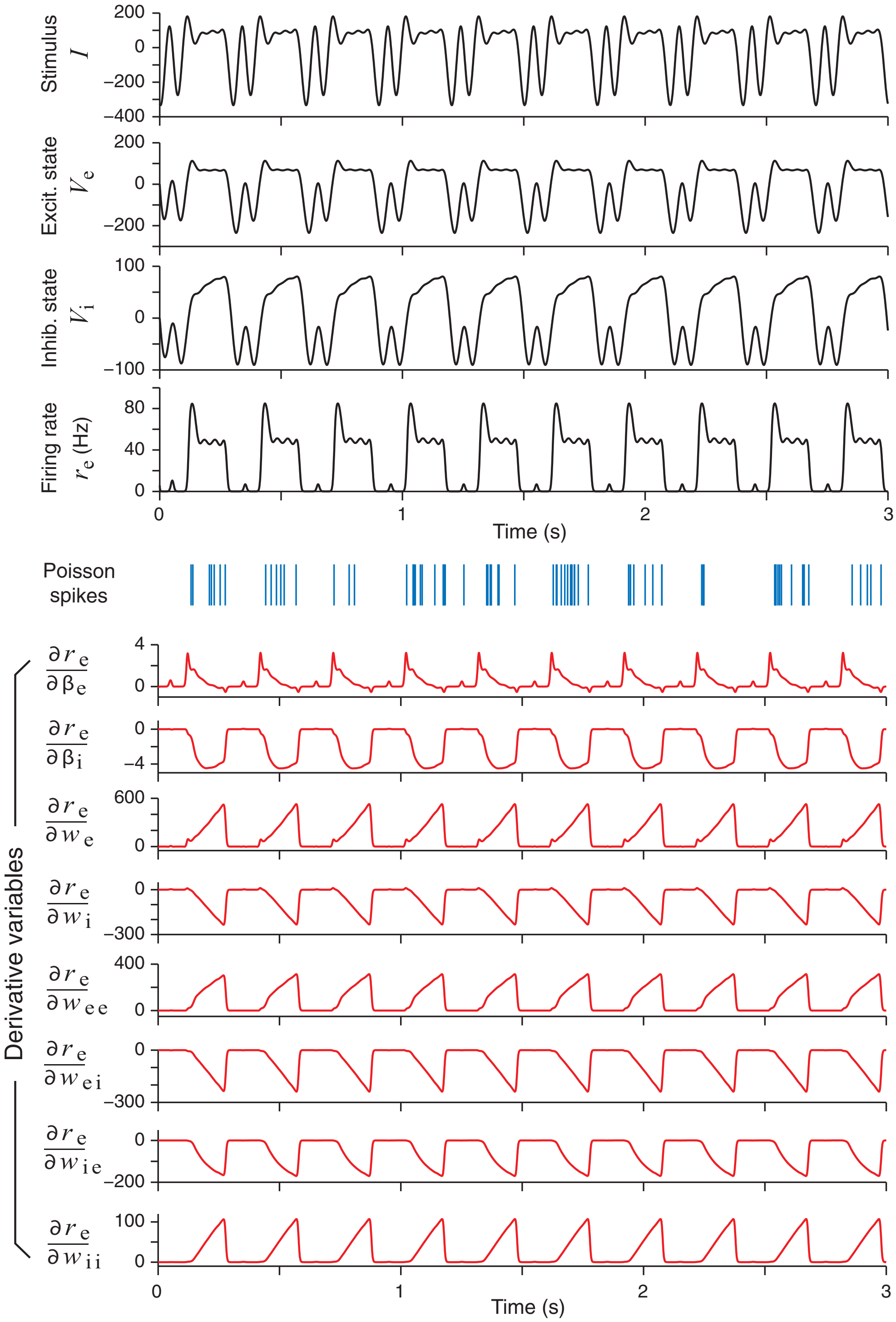

FIGURE 3 | An example of an optimally designed stimulus with a duration of 3 s (top panel). The responses of the excitatory and the inhibitory units in the network are shown below, followed by an example of spike trains (blue vertical bars) generated by an inhomogeneous Poisson process according to the continuous firing rate of the excitatory unit (r_e). In response to this stimulus, the time courses of eight derivatives variables, namely, the derivatives of the firing rate r_e with respect to all the network parameters, are shown as red curves. These derivatives were solved directly from the differential Equation (29).

fluctuations of a term with a high-sensitivity parameter. We also note that the times courses of some of the sensitivity derivatives looked very similar, while others looked like mirror images of one another. This means that small increments of these parameters affect the output firing rate r_e in either the same direction or the opposite direction, due to the intrinsic dynamics of the

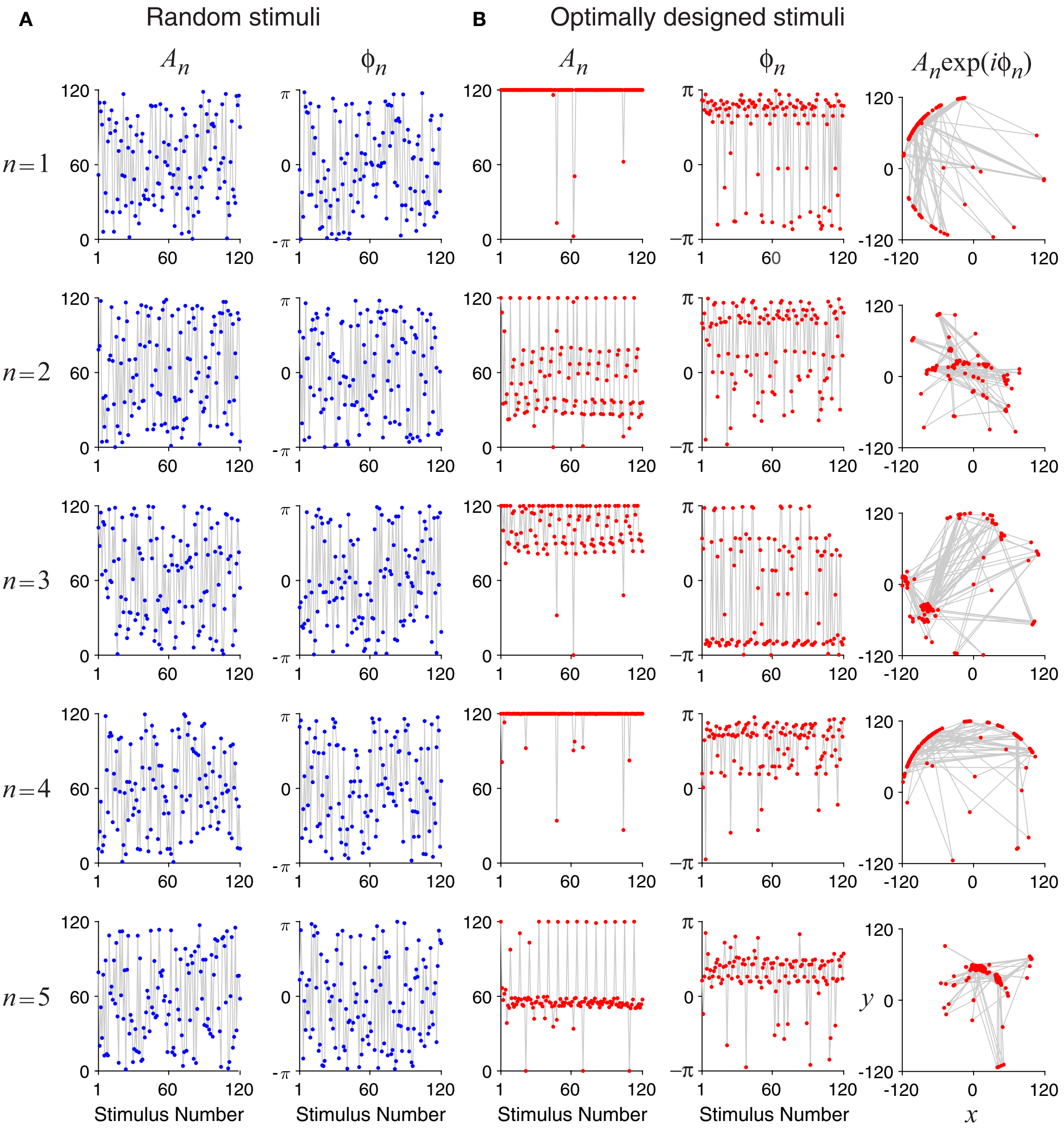

FIGURE 4 | Time evolution of the optimally designed stimuli in a simulated experiment with 120 iterations. **(A)** Random stimuli with uniformly distributed random Fourier amplitudes A_n and phases ϕ_n. **(B)** The optimally designed stimuli show specific patterns in the amplitudes A_n and the phases ϕ_n. The rightmost column is a vector representation of A_n and ϕ_n using complex number $x + iy = A_n \exp(i\phi_n)$.

network and how the excitatory unit and inhibitory unit are wired up.

Unlike random stimuli generated by Fourier series with random amplitudes and phases, the optimally designed stimuli seemed to have some inherent structures, which can be observed in the stimulus sequence generated one by one in a simulated experiment (**Figure 4**) as well as in the histograms obtained from repeated experiments (**Figure 5**). The appearance of oscillations in the sequence of optimally designed stimuli in **Figure 4B** is probably due to the spontaneous switching by the optimal design algorithm to probe different parameters (DiMattina and Zhang, 2011). The random stimuli were used as a control in our study. Their amplitudes A_n and the phases ϕ_n showed flat distributions in the histograms (**Figure 5A**). This is expected because the stimuli were generated by randomly drawing the amplitudes and phases from a uniform distribution. By contrast, the amplitudes A_n and the phases ϕ_n of the optimally designed stimuli were distributed in more complex manners (**Figure 5B**). For instance, the distribution of amplitude A_5 (bottom panel) had a major peak in the middle, besides additional concentrations at either the lower bound or the upper bound. The lower and upper bounds mean the minimum and the maximum energy allowed for each

Fourier component. The distributions of the amplitudes showed a tendency to cluster at the upper bound, which occurred in about half of the cases; the amplitudes lied between the lower and upper bounds also in about half of the cases, while the lower bound was reached in less than 2% of the cases. Our Fisher information maximization procedure yielded phase parameters with specific distributions. In fact, none of the phase parameters (rightmost column in **Figure 5B**) had a flat distribution; instead, they all seemed to have distributions with multiple peaks, indicating that the phases were not irrelevant variables for the optimal design.

The tendency for optimally designed stimuli to cluster around the topological boundary of the set of allowable stimuli was reported for feedforward network models (DiMattina and Zhang, 2011). In that situation, both the feedforward network and the stimuli were stationary without involving time. Here we found similar topological boundary property for time-varying stimuli in a dynamic recurrent network. We suspect there might exist deeper mathematical reasons underlying this phenomenon although a full and rigorous analysis is beyond the scope of this paper.

3.3. Maximum-Likelihood Estimates of Network Parameters

Given a dataset consisting of stimulus-response pairs, we can always use maximum-likelihood estimation to fit the network model to the data to recover all the parameters. Maximum-likelihood estimation is known to be asymptotically efficient in the limit of large data size, in the sense that the estimation is asymptotically unbiased (i..e, average of the estimates approaches the true value) and has minimal variance (i.e., the variance of the estimates approaches the Cramér-Rao lower bound).

The accuracies of the maximum-likelihood estimates from the optimally designed stimuli, as well as from the random stimuli, are shown in **Table 3**. Both types of stimuli yielded reasonably accurate estimates of the network parameters although some parameters appeared harder to estimate than others. The maximum-likelihood estimates based on optimally designed stimuli were closer to the true values, on average, than that based on random stimuli. Furthermore, the standard deviations of the estimates based on optimally designed stimuli were also smaller, indicating more consistent estimates in repeated trials.

A further comparison of the accuracies of individual parameter estimates is shown in **Figure 6**. As expected, the estimates of all parameters tended to be more accurate when the number of stimuli increased, both for the random stimuli and for the optimally designed stimuli. For every individual parameter, the error for the optimally designed stimuli was always smaller than that for the random stimuli, across all total numbers of stimuli we tested. The statistical significance of the differences was evaluated by Wilcoxon rank-sum test (Mann and Whitney, 1947; Gibbons and Chakraborti, 2011; Hollander et al., 2014). Most of the differences reached statistical significance at $p = 0.05$, as indicated by the asterisk ($*$) above the bars in **Figure 6**. That is, in all cases with an asterisk, the estimates from optimal design was significantly better than that from random stimuli.

With the total number of stimuli $M = 120$ (the largest sample size in our tests in **Figure 6**), the p-values of Wilcoxon rank-sum test for all the parameters estimated by the two methods were as follows: $p\left(\beta_{\mathrm{e}}\right) = 4.5 \times 10^{-5}$, $p\left(\beta_{\mathrm{i}}\right) = 9.1 \times 10^{-3}$, $p\left(w_{\mathrm{e}}\right) = 0.989$, $p\left(w_{\mathrm{i}}\right) = 6.9 \times 10^{-3}$, $p\left(w_{\mathrm{ee}}\right) = 2.0 \times 10^{-3}$, $p\left(w_{\mathrm{ei}}\right) = 5.6 \times 10^{-3}$, $p\left(w_{\mathrm{ie}}\right) = 2.2 \times 10^{-5}$, and $p\left(w_{\mathrm{ii}}\right) = 2.9 \times 10^{-8}$. Thus the optimally design stimuli yielded significantly more accurate estimates than the random stimuli for 7 out of 8 parameters at the significance level $p = 0.05$. The only exception is the third parameter, w_{e}, which was not significant regardless of the sample size M (**Figure 6**). This parameter was among the easiest to estimate because it had low errors relative to the true value for both methods (**Table 3** and **Figure 6**). Although the optimal design method performed slightly better across all sample sizes, the small differences between the two methods did not reach statistical significance for our sample sizes (**Figure 6**). The error of parameter w_{e} might also be associated with the parameter confounding phenomenon, which will be discussed in section 3.5. The most significant difference, or the smallest p-value, was achieved for the parameter w_{ii}, which was the hardest parameter to estimate because it had the largest errors relative to the true value for both methods (**Table 3** and **Figure 6**). Parameter w_{ii} tended to have larger errors because the data recording was performed directly on the excitatory unit whose activity was affected only indirectly by the inhibitory unit self-connection weight. For the parameter which was the most difficult to estimate, the advantage of the optimal design over random stimuli was the most significant.

Besides means and standard deviations, we now examine the actual distributions of the likelihood values for the two types of estimates. We found that maximum likelihood obtained from the optimally design stimuli was typically much better than that obtained from the random stimuli (**Figure 7A**). A proper comparison should be based on the same number of stimuli. because for each estimation method, the likelihood value increased with the number of stimuli (**Figure 7B**), For a given number of stimuli, the optimally designed stimuli always yielded much greater likelihood value than the random stimuli. The minimum difference between the two sets of likelihood values (i.e., between the maximum from the random stimulus samples and the minimum from the optimal design samples) was typically at least two times greater than the standard deviation of either estimate except for the case with 24 samples ($M = 24$ with $N_{\mathrm{itr}} = 3$). Even in this case, this violation occurred only for one outlier data point.

As expected from the log likelihood formula (11), the log likelihood value should increase approximately linearly with the number of stimuli, M, assuming that each stimulus, on average, contributes about the same to the log likelihood function. Although this assumption is not exactly true in further analysis, it is a good enough approximation that is compatible with the linear relationship. The linear relationship was confirmed by the numerical results in **Figure 7B**. Based on the regression lines, the values of the log likelihood have the following empirical formulas:

$$l_{\mathrm{opt}} = 257.1 M_{\mathrm{opt}} + 76.2 \tag{41}$$

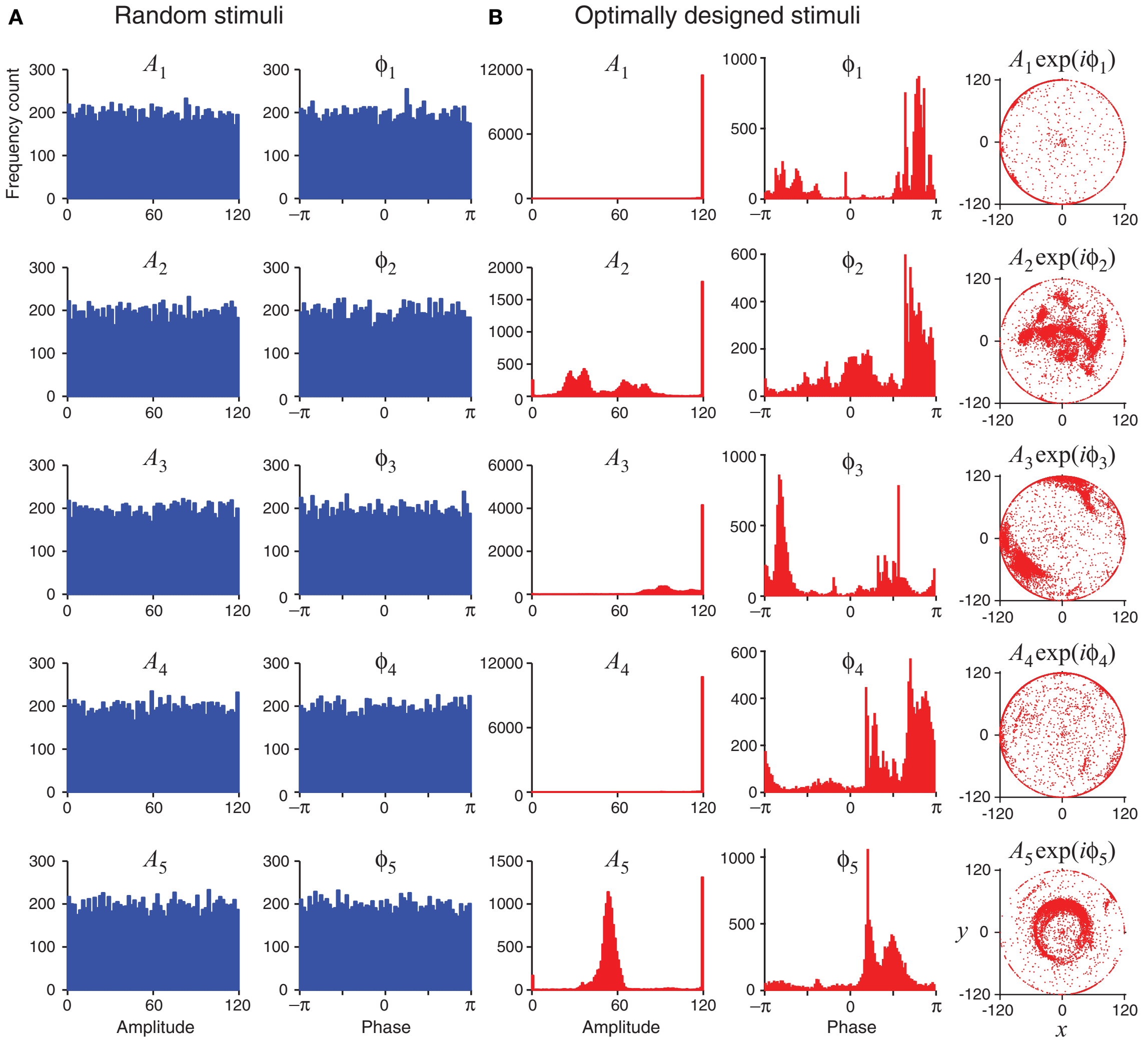

FIGURE 5 | Statistics of random stimuli and optimally designed stimuli, showing distributions of their Fourier amplitudes $A_1, \ldots, A_5$ and phases $\phi_1, \ldots \phi_5$. Each subplot is based on 12,000 stimuli, either random or optimally designed. **(A)** Random stimuli were generated by choosing their Fourier amplitudes and phases randomly from a uniform distribution. **(B)** The optimally designed stimuli showed some structures in the distributions of the Fourier amplitudes and phases. The rightmost column is a vector representation of A_n and ϕ_n using complex number $x + iy = A_n \exp(i\phi_n)$.

for M_{opt} optimally designed stimuli, and

$$l_{\text{rand}} = 174.5 M_{\text{rand}} + 31.1 \tag{42}$$

for M_{rand} random stimuli. Both regression lines path through the origin point (0, 0) approximately (**Figure 7B**). This means that the ratio of the log likelihood values in the two cases is approximately a constant, as long as the two methods have the same number of stimuli. That is, when $M_{\text{opt}} = M_{\text{rand}}$, it follows from Equations (41) and (42) that the log likelihood ratio is approximately:

$$\frac{l_{\text{opt}}}{l_{\text{rand}}} \approx \frac{257.1}{174.5} = 1.47 \tag{43}$$

where the small constant terms in Equations (41) and (42) are ignored. In other words, the two lines have a slope ratio of approximately 1.47. There is another way to look at the empirical formulas. For achieving the same level of likelihood value ($l_{\text{opt}} = l_{\text{rand}}$), it follows from Equations (41) and (42) that

$$\frac{M_{\text{opt}}}{M_{\text{rand}}} \approx \frac{174.5}{257.1} = 0.68 \tag{44}$$

where once again the small constant terms in Equations (41) and (42) are ignored. Thus the optimal design saves the number of stimuli by about 32% or 1/3. Although this saving seems moderate, it may become useful in situations where it is extremely valuable to keep the duration of an experiment as short as possible.

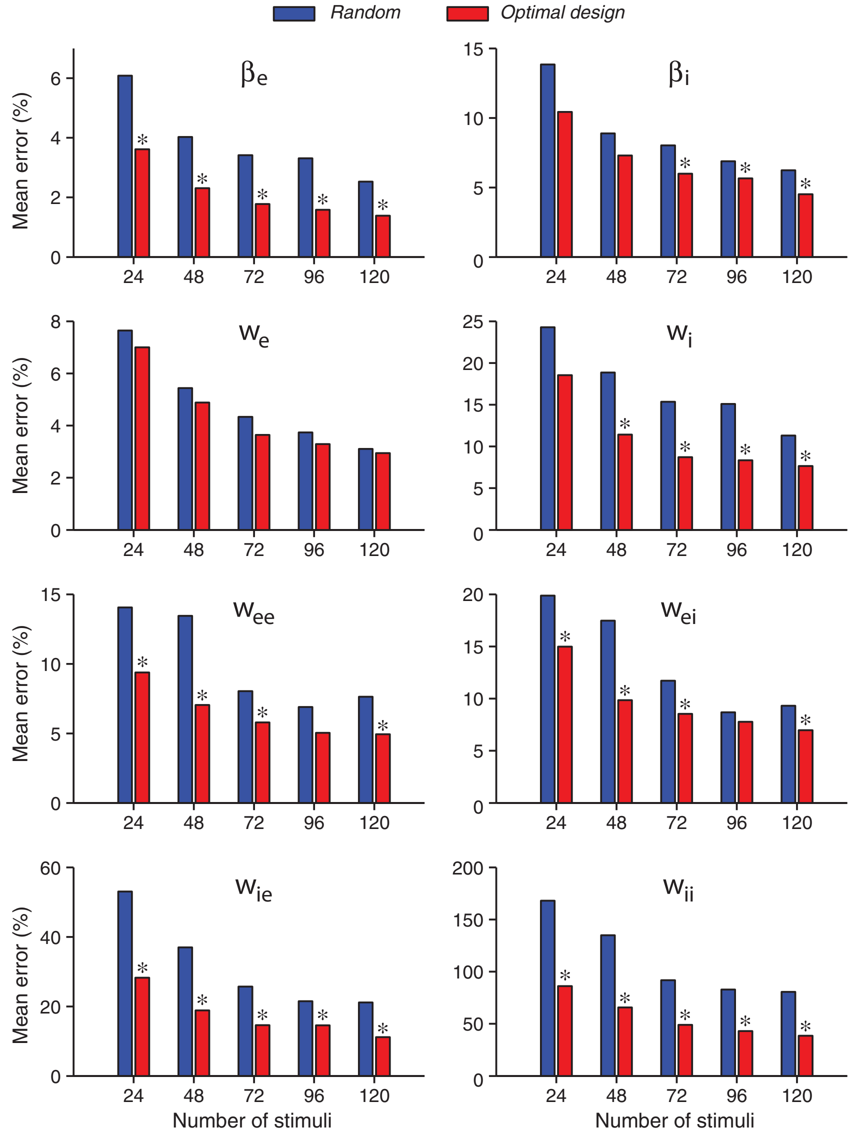

FIGURE 6 | Errors of individual network parameters obtained by maximum likelihood estimation tended to decrease as the number of stimuli increased. Compared with the random stimuli, the optimally design stimuli yielded smaller errors for all network parameters in all the cases tested, although not all differences reached statistical significance. An asterisk (∗) at the top of an optimal design bar indicates that the difference from the neighboring bar for random stimuli was statistically significant at the level $p < 0.05$ in the Wilcoxon rank-sum test. The mean error was calculated as the absolute value of the relative percentage of the difference between the estimate and the true value, averaged over 100 repeated trials.

3.4. Problem of Local Maxima During Optimization

We used optimization in two places: maximum likelihood estimation of parameters, and optimal design of stimuli. In the optimization process, local algorithms such as the constrained optimization function *fmincon* in Matlab need an initial guess to start the iterative procedure, but not all of the initial guesses will eventually converge to the true value. Generally speaking, as the number of initial guesses increases, it becomes more and more likely for one to find the global optimum rather get stuck in a local optimum. In the following, we perform an analysis to assess how likely our optimization processes were able to find the global optimum in repeated trials.

We emphasize that there is no need to always find the absolutely best solution in the actual optimization process. A good enough solution is probably good enough. Our purpose in this section is to perform a simple test to assess how hard our optimization problem really is. In repeated runs if all solutions end up to be approximately the same, it means that the optimization problem is easy, probably with a single global optimum, which makes repeated runs unnecessary. On the other hand, if there are many local optima, it is harder to hit on the

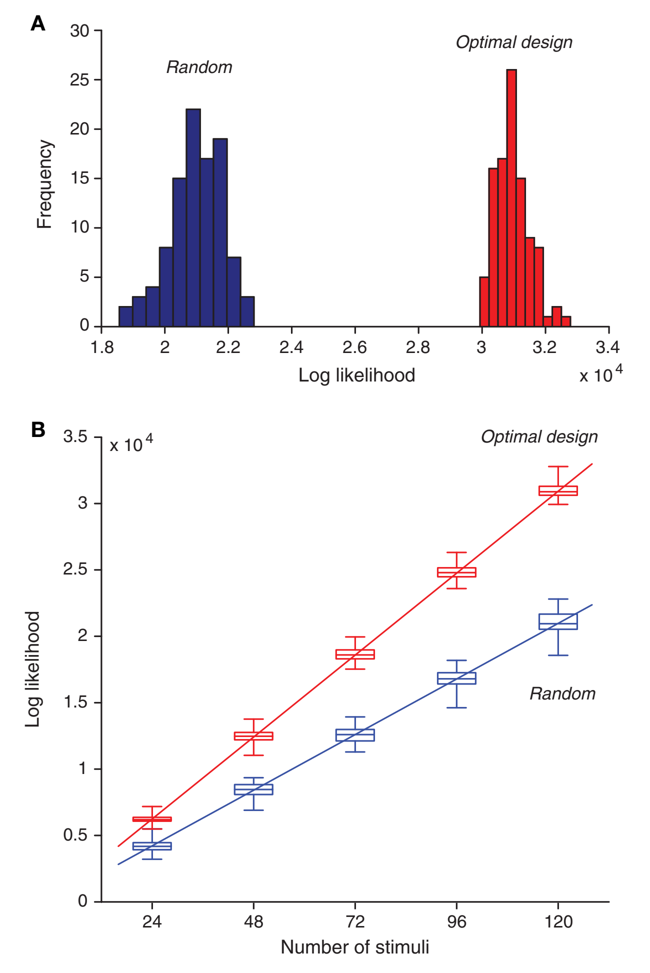

FIGURE 7 | Optimally designed stimuli yield greater likelihood values for maximum-likelihood parameter estimation compared to random stimuli. **(A)** Histogram of optimized likelihood values for 100 trials with random stimuli is compared with that for 100 trails with optimally designed stimuli. Note that even the best value from the random trials was much worse than the worst value from the optimal design trials. Here each trial contained a sequence of 120 stimuli, generated either randomly or by optimal design. Each likelihood value was obtained by maximizing the likelihood function in Equation (11) using the response data elicited by all 120 stimuli. **(B)** As the number of stimuli increased, the likelihood function also increased, following an approximate linear relationship. The optimal design yielded better likelihood values than random stimuli regardless of the number of stimuli. The boxplot shows 25%, 50% (median), and 75% percentiles, with the whiskers outside of the boxes indicating the minimum and maximum values. The straight lines were obtained by linear regression on the median values [Equations (41) and (42)].

best solution in a single run, and repeated runs become useful. In this situation it is also helpful to allow multimodal distribution of the parameter estimates so as to discourage early commitment to incorrect parameter values (see section 4).

The basic idea of our test is very simple. Let p be the probability of finding the "best" solution (tentative global optimum) in an individual run with a random initial guess. Suppose there are n repeats or starts from different initial values. Then in n repeated runs, the probably that at least one run will lead to the "best" solution is

$$\text{Prob (“best”)} = 1 - (1 - p)^n \tag{45}$$

A rough estimate of probability p can be obtained in two different ways, as explained below.

First, we consider the optimization process for maximum-likelihood estimation of parameters. We estimated the value of Prob ("best") in (45) by starting the optimization from different initial guesses and then checking the number of solutions which stay in a relative error bound of fraction η for each individual parameter with respect to the best solution with the highest likelihood value. In other words, to pass the test the following criterion should be satisfied for all individual parameters $(\theta_1, \cdots, \theta_8)$ in (8):

$$\frac{\left|\hat{\theta}_i - \hat{\theta}_i^{\text{best}}\right|}{\theta_i} \le \eta \qquad (i = 1, \cdots, 8) \tag{46}$$

where θ_i is the true value, $\hat{\theta}_i$ is the estimate from a particular run, $\hat{\theta}_i^{\text{best}}$ is the best solution having the highest likelihood value in the repeated runs, and η is a constant. For simplicity we used $\eta = 10\%$ in all simulations reported in this paper; in other words, the error tolerance was set to be one order of magnitude smaller than the true parameter value. One may choose other values as well. To make the criterion less arbitrary, one could use Fisher information to estimate the asymptotic value of the error. In (46) we used $\hat{\theta}_i^{\text{best}}$ instead of the true value θ_i in the numerator of because even the true global maximum of the likelihood function could still be substantially different from the true parameter value. If the above was satisfied for all $\theta_1, \cdots, \theta_8$, this result was counted as one pass of the test. By counting this way, we obtained an estimate of Prob ("best"), which was then inserted into (45) to solve for the probability p. The final result was $p \approx 0.85$, which was based on 200 runs with 10 multiple initial guesses per 20 different stimuli configurations, and each configuration had $M = 120$ stimuli. In conclusion, in our maximum-likelihood parameter estimation, we might have a high probability ($p \approx 0.85$) of obtaining a global maximum of the likelihood function in a single run. It implies that to get a 99% correct rate we would only need $n = 3$ repeats of the optimization procedure.

Next, we consider the optimization process for optimal design of stimuli. This optimization problem turned out to be much harder than the maximum-likelihood optimization described above because there were more local optima. Recall that the Fisher Information measure $U_k(\mathbf{x}, \theta)$ is computed with respect to each parameter θ_k as shown in (23). We need to consider how $U_k(\mathbf{x}, \theta)$ is maximized by the stimulus parameters $\mathbf{x} = [x_1, \cdots, x_{2N}] = [A_1, \cdots, A_N, \phi_1, \cdots, \phi_N]$ as given in (15). Like in the case of likelihood analysis, we used repeated runs to estimate Prob ("best") first and then computed p from (45). Similar to (46), the criterion for passing the test was $\left|\hat{x}_i - \hat{x}_i^{\text{best}}\right| / \hat{x}_i^{\text{best}} \le 10\%$ for all i, where $\hat{x}_i$ is the value of parameter x_i obtained by maximizing $U_k(\mathbf{x}, \theta)$ in a particular run, and $\hat{x}_i^{\text{best}}$ is the parameter value that yielded the largest value of $U_k(\mathbf{x}, \theta)$ in the repeated runs. Here $\hat{x}_i^{\text{best}}$ is used in the denominator as well as in the numerator because unlike (46), there was no such as concept as a true stimulus parameter value.

Our simulation indicated that the maximum of the Fisher Information measure $U_k\,(\mathbf{x}, \theta)$ tended to depend more sensitively on the stimulus Fourier amplitudes A_n than the stimulus Fourier phases ϕ_n. This difference might be related to the fact that the optimized amplitudes had a tendency to cluster at the upper bound, as described in section 3.2. In the following analysis, the criterion of finding the best solution was applied to the amplitudes only. The largest probability of finding the best solution for the amplitudes was $p = 0.86$, which was obtained for $U_1\,(\mathbf{x}, \theta)$, corresponding to the parameter $\theta_1 = \beta_{\mathrm{e}}$. The smallest probability $p = 0.175$ was obtained for $U_5\,(\mathbf{x}, \theta)$, corresponding to the parameter $\theta_5 = w_{\mathrm{ee}}$. The second and third smallest values $p = 0.275$ and $p = 0.35$ were obtained for $U_3\,(\mathbf{x}, \theta)$ and $U_6\,(\mathbf{x}, \theta)$, corresponding to $\theta_3 = w_{\mathrm{e}}$, and $\theta_6 = w_{\mathrm{ei}}$, respectively. It is worth mentioning that those three parameters were confounded or had strong correlations with at least one other network parameter (see section 3.5). Even in the worst case ($p = 0.175$ for $U_5\,(\mathbf{x}, \theta)$), one would find the best solutions for the amplitudes with a 0.99 probability after $n = 23$ repeated runs with random initial guesses for the optimization procedure.

In summary, repeated runs with random initial guesses greatly increased the chance of finding the global maximum, although in our simulations this method appeared more effective for maximum-likelihood parameter estimation than stimulus optimization which suffered much more from the local optima problem. Fortunately, for practical applications, the existence of many more local maxima in the stimulus optimization process might not present a big problem because even partially optimized stimuli would still be better than nonadaptive random stimuli. There is a tradeoff between the quality of the stimulus optimization and the computational time, which could be reduced further in the future by the increasing power of parallel computing because repeated runs are readily parallelizable.

3.5. Parameter Confounding

In our simulations, the errors of different parameter estimates were often correlated (**Figure 8**). Some of the correlations may be explained by parameter confounding. The basic idea is that different parameters may compensate each other such that the output of the network behaves essentially in the same way, even when the parameter values are configured differently. It is known that in individual neurons, different ion channels may be regulated such that diverse configurations may lead to neurons with similar neuronal behaviors in their electrical activation patterns (Prinz et al., 2004). Parameter confounding also exists at the network level, for example, in multilayer perceptrons (DiMattina and Zhang, 2010).

We will examine the original dynamical equations and demonstrate how approximate parameter confounding might arise. We emphasize that different parameters in the excitatory-inhibitory network are distinct and independent, and no strict confounding exists. All the parameter confounding problems considered here are approximate in nature.

Based on the analysis in **Figures 8A,B**, three pairs of parameters stand out with the strongest correlations. These pairs are $(\beta_{\mathrm{e}},\, w_{\mathrm{e}})$, $(\beta_{\mathrm{i}}, w_{\mathrm{ei}})$ and $(w_{\mathrm{ee}}, w_{\mathrm{ei}})$. We will use the idea of parameter confounding to offer an intuitive heuristic explanation of why these three pairs tend to be correlated.

Example 1: Confounding of the Parameter Pair (β_{e}, w_{e})

We first rewrite the dynamical Equations (3) and (4) in the following form:

$$\dot{V}_{\mathrm{e}} = -\beta_{\mathrm{e}} V_{\mathrm{e}} + \beta_{\mathrm{e}} \left\{ w_{\mathrm{ee}} g_{\mathrm{e}}(V_{\mathrm{e}}) - w_{\mathrm{ei}} g_{\mathrm{i}}(V_{\mathrm{i}}) \right\} + \beta_{\mathrm{e}} w_{\mathrm{e}} I \quad (47)$$

$$\dot{V}_{\mathrm{i}} = -\beta_{\mathrm{i}} V_{\mathrm{i}} + \beta_{\mathrm{i}} \left\{ w_{\mathrm{ie}} g_{\mathrm{e}}(V_{\mathrm{e}}) - w_{\mathrm{ii}} g_{\mathrm{i}}(V_{\mathrm{i}}) \right\} + \beta_{\mathrm{i}} w_{\mathrm{i}} I \quad (48)$$

The external stimulus I drives the first Equation (47) through the weight $\beta_{\mathrm{e}} w_{\mathrm{e}}$. If this product is the same, the drive would be the same, even though the individual parameters are different. For example, if β_{e} is increased by 25% from its true value while w_{e} is decreased by 20% from its true value, then their product $\beta_{\mathrm{e}} w_{\mathrm{e}}$ stays the same, so that the external input I provides the same drive to (47). Of course, any deviation from the true parameter values also leads to other differences elsewhere in the system. Therefore, the confounding relation is only approximate and not strict. This heuristic argument gives an empirical formula:

$$\beta_{\mathrm{e}} w_{\mathrm{e}} = \hat{\beta}_{\mathrm{e}} \hat{w}_{\mathrm{e}} \quad (49)$$

where β_{e} and w_{e} refer to the true values of these parameters, whereas $\hat{\beta}_{\mathrm{e}}$ and $\hat{w}_{\mathrm{e}}$ refer to the estimated values.

Example 2: Confounding of the Parameter Pair ($w_{\mathrm{ei}}, \beta_{\mathrm{i}}$)

These two parameters appear separately in different equations, namely, w_{ei} appearing only in (47) while β_{i} appearing only in (48). To combine them, we need to consider the interaction of these two equations. To simplify the problem, we consider a linearised system around the equilibrium state:

$$\dot{V}_{\mathrm{e}} = -\beta_{\mathrm{e}} V_{\mathrm{e}} + \beta_{\mathrm{e}} \left\{ w_{\mathrm{ee}} k_{\mathrm{e}} V_{\mathrm{e}} - w_{\mathrm{ei}} k_{\mathrm{i}} V_{\mathrm{i}} \right\} + \beta_{\mathrm{e}} w_{\mathrm{e}} I + C_{\mathrm{e}} \quad (50)$$

$$\dot{V}_{\mathrm{i}} = -\beta_{\mathrm{i}} V_{\mathrm{i}} + \beta_{\mathrm{i}} \left\{ w_{\mathrm{ie}} k_{\mathrm{e}} V_{\mathrm{e}} - w_{\mathrm{ii}} k_{\mathrm{i}} V_{\mathrm{i}} \right\} + \beta_{\mathrm{i}} w_{\mathrm{i}} I + C_{\mathrm{i}} \quad (51)$$

where k_{e} and k_{i} are the slopes of the gain functions, and C_{e} and C_{i} are extra terms that depend on the equilibrium state and other parameters. Note that V_{i} appears in (50) only once, in the second term in the curly brackets. Since V_{i} also satisfies (51), we solve for V_{i} in terms of $\dot{V}_{\mathrm{i}}$ from (51) and find a solution of the form: $V_{\mathrm{i}} = c\dot{V}_{\mathrm{i}}/\beta_{\mathrm{i}} + a$ where $c = -1/(1 + w_{\mathrm{ii}} k_{\mathrm{i}})$ is a constant and $a = (\beta_{\mathrm{i}} w_{\mathrm{i}} I + C_{\mathrm{i}}) / \left(\beta_{\mathrm{i}}(1 + w_{\mathrm{ii}} k_{\mathrm{i}})\right)$ which depends on input I. Substitution into (50) eliminates V_{i}, yielding an equation of the following form:

$$\beta_{\mathrm{e}}^{-1} \dot{V}_{\mathrm{e}} = (w_{\mathrm{ee}} k_{\mathrm{e}} - 1) V_{\mathrm{e}} - c k_{\mathrm{i}} (w_{\mathrm{ei}}/\beta_{\mathrm{i}}) \dot{V}_{\mathrm{i}} - k_{\mathrm{i}} w_{\mathrm{ei}} a + w_{\mathrm{e}} I + C_{\mathrm{e}}/\beta_{\mathrm{e}} \quad (52)$$

Note that the parameter combination $w_{\mathrm{ei}}/\beta_{\mathrm{i}}$ is a factor that scales how strongly $\dot{V}_{\mathrm{i}}$ influences this equation. We use this combination as the basis for the following heuristic confounding relation:

$$w_{\mathrm{ei}}/\beta_{\mathrm{i}} = \hat{w}_{\mathrm{ei}}/\hat{\beta}_{\mathrm{i}} \quad (53)$$

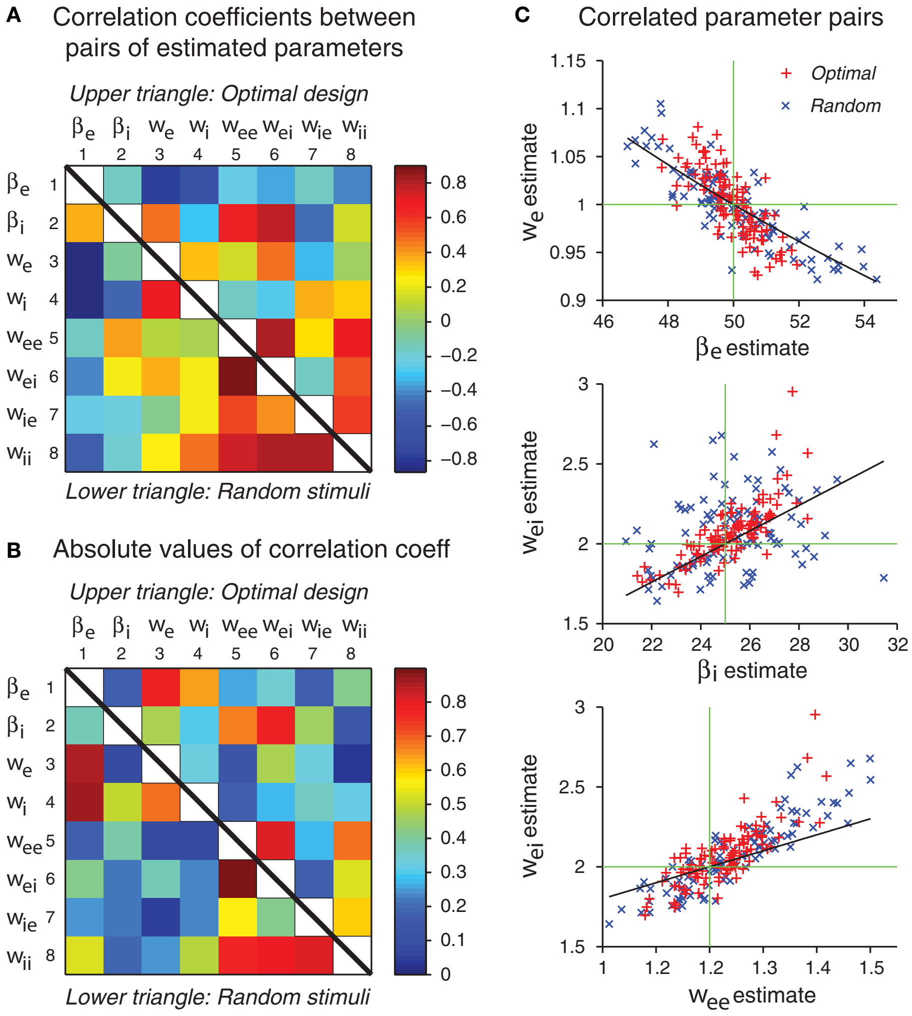

FIGURE 8 | Some network parameters are approximately confounded, in the sense that a change of one parameter can be compensated by a proper change of another parameter, such that the stimulus-response relation of the whole network is kept approximately the same. **(A)** Correlation coefficient matrix of all possible pairs of network parameters obtained by maximum likelihood estimation from either optimal design data or random stimulus data. Since each matrix is symmetric, only a half needs to be shown. Here data from optimal design trials are shown in the upper triangle, whereas data from the random trials are shown in the lower triangle. Each data point was based on 100 repeated trials each containing a sequence of 120 stimuli. **(B)** Same data as in panel A, except that the absolute values are shown. **(C)** The three pairs of parameters with the highest correlation coefficients in the upper triangle in panel B are shown in the scatter plots. Data from both optimal design and random trials are shown. Black curves are theoretical predictions according to Equations (49), (53), and (54). Green crosshairs are centered at the true parameter values.

Example 3: Confounding of the Parameter Pair (w_{ee}, w_{ei})

These two parameters both appear in the curly brackets in (47). We have a heuristic confounding relation:

$$w_{ee}g_e(\bar{V}_e) - w_{ei}g_i(\bar{V}_i) = \hat{w}_{ee}g_e(\bar{V}_e) - \hat{w}_{ei}g_i(\bar{V}_i) \qquad (54)$$

where $\bar{V}_e$ and $\bar{V}_i$ are the equilibrium states. If this equation is satisfied, we expect that the term in the curly brackets in (47) would be close to a constant [the right-hand side of (54)] whenever the state V_e and V_i are close to the equilibrium values. When the state variables vary freely, we expect this relation to hold only as a crude approximation.

The theoretical curves shown in **Figure 8C** are based on Equations (49), (53), and (54), and we emphasize that there is no free parameter. These three approximate confounding relations can qualitatively account for the data for optimally designed stimuli (**Figure 8C**). The data for random stimuli also appear to follow the same pattern (**Figure 8C**) although they seem to have more scattering than the optimal design counterpart. The theoretical slopes are somewhat smaller than the slopes of the empirical correlations, suggesting that the heuristic theory only accounts for a portion of the correlation. It is possible that the empirical correlations contain contributions from other parameters, beyond these three pairs and pair-wise correlations. The static multilayer perceptrons already show various types of parameter confounding behaviors (DiMattina and Zhang, 2010). The dynamic recurrent network models, which formally subsume the multilayer perceptrons as special equilibrium states, are expected to have more complicated behaviors of parameter confounding. In sum, our results indicate that the errors of different parameter estimates have a tendency to be correlated in specific manners, and some of the correlation patterns can be explained by an argument of approximate parameter confounding or compensation based on the differential equations of the underlying dynamics.

4. DISCUSSION

We have implemented an adaptive design algorithm for generating dynamic stimuli that can efficiently probe a recurrent network with interacting excitatory and inhibitory units so as to optimize its parameter estimation. The network was a standard rate model with continuous dynamics which could reflect the average dynamics of a group of neurons with similar response properties. Since single-unit recordings in neurophysiological experiment yield spike train data, our optimal design process was based on artificial spike trains generated by an inhomogeneous Poisson process whose rate was determined by the continuous network state. Our simulated recording experiment was performed on the excitatory unit in the simplest excitatory-inhibitory model while the activity of the inhibitory unit was not directly known (**Figure 1B**). This situation was not unlike most neurophysiological experiments where there were always some relevant neurons whose activities were hidden from view but contributed indirectly to the recorded data. The time-varying stimuli were parameterized by a Fourier series whose parameters were determined by maximizing a utility function based on the Fisher information matrix; in other words, the stimuli were designed to elicit responses that would reveal the most information about the values of the network parameters. Each stimulus was fed to the network to elicit a Poisson spike train, and from the available stimulus-response data we inferred the values of all the network parameters, including the time constants and all the connection weights, by maximum-likelihood estimation, and the updated parameter values were then used to design the next stimulus, and so on, mimicking a closed-loop experiment (Benda et al., 2007; DiMattina and Zhang, 2013; Potter et al., 2014; El Hady, 2016). The optimization process was facilitated by knowing the derivatives of the network states with respect to the network parameters, and we found it convenient to compute the time evolutions of the derivatives by directly solving differential equations derived from the original system (**Figure 3**).

We confirmed that optimally designed stimuli elicited responses with much better likelihood values for parameter estimation than nonadaptive random stimuli (**Figure 7**), and every single parameter was recovered more accurately by the optimally designed stimuli (**Figure 6**). The statistical significance of the differences tended to be more pronounced for parameters that were harder to estimate with larger relative errors (section 3.3), and this observation underscores the value of the optimal design method for difficult estimation problems. Following the approximately linear relationship between the log likelihood and the number of stimuli (Equations 41, 42), we estimated that the optimal design would cut down the number of stimuli by about 1/3 to achieve the same level of parameter estimation accuracy as the nonadaptive random stimuli. Although this saving seems to be moderate, it still could potentially be important in real situations where time is extremely valuable, such as when recording time is seriously limited in experiments. We also noted that the Fourier amplitudes and phases of the optimally designed stimuli were not uniformly distributed but had specific features (**Figure 5**). Applying the features of optimally designed stimuli (e.g., the tendency for amplitude saturation) to bias the random stimuli could potentially lead to better parameter estimation (DiMattina and Zhang, 2011). In this paper we simply compare the optimal design against unbiased random stimuli because this is a fair comparison when one only knows the bounds of the parameters without any other information.

We found that the errors of different parameter estimates were correlated in specific ways, and some of the correlation patterns were rather predictable (**Figure 8**). We have derived heuristic formulas to account for some of the most prominent correlation patterns by an approximate theory of parameter confounding where the effect of changing one parameter was compensated by changing another parameter in a particular direction with an appropriate amount. This finding suggests that it would be useful in future studies to examine not only the errors of different parameter estimates but also their correlations, because they may reflect the underlying network connectivity and dynamics as predicted by the parameter confounding theory. It would be an interesting future research topic to study how to take advantage of the error correlation phenomenon to further improve the optimal design method.

Our stimulus design is based on maximizing an objective function or utility function, which may contain multiple local maxima. The existence of these local maxima makes it harder to find the global maximum. One related problem is early commitment to incorrect parameters during the iterative procedure. If the initial parameter estimates happen to be very wrong, then subsequent iterations may push the parameter estimation further down the wrong path, eventually getting stuck in an incorrect local maximum of the objective function. To assess how likely this may happen, we run the optimization procedure repeatedly with randomly chosen initial parameters and examined how often the final parameter estimates ended up around the putative global maximum (section 3.4). Although the problem of local maxima appeared manageable for the simple model studied in this paper, we expect that the problem may become more serious as model complexity increases. To better address the problem of early commitment to incorrect parameters, we should extend the theoretical framework to allow a potentially multimodal posterior distribution of the parameter values rather than only keeping the current best estimates. In a preliminary experiment of stimulus optimal design for auditory cortical neurons (Feng et al., 2012; Feng, 2013), four independent sets of parameter estimates were maintained throughout the sequential procedure so that a set of seemingly bad estimates at the initial stage could still have a chance to win the competition at a later time. In this work the stimuli as well as the model were stationary rather than time-varying. In the future it would be useful to develop our method for time-varying stimuli by incorporating a potentially multimodal posterior distribution of the parameters.

Speed of computation is an important issue when applying optimal design method to a real neurophysiological experiments. There is a basic tradeoff between the computational time and the quality of optimization in the optimal design. In our simulation, the optimization of a single 3-s long stimulus took about 15 min on a single PC with a 6-core Intel Core i7 processor. The number of steps required to converge to an optimum depended on multiple factors such as the current values of the network parameter estimates, the values of the objective function and the gradients, constraint violation, and step size. The optimization of the maximum likelihood parameter estimation had an additional problem: it tended to slow down as the spikes accumulate [see Equation (11)] leading to an objective function with gradually increasing complexity. An average value for the observed duration of the maximum likelihood optimization was about 38 min. As a result, optimization of one stimulus and subsequent maximum likelihood estimation took approximately 1 h. So one complete run of a simulated experiment with $M = 120$ optimally design stimuli took over 100 h.

To bring down the computation time for each 3-s stimulus to less than 3 s as required for real online experiments, we would need a 10^3-fold speedup. There are several ways to speed up the computation although we did not try to implement them here since our focus in this paper was on offline simulation to prove the concept. In our simulation we integrated the equations using a time step of 0.001-s, which could be increased to a level as high as 0.01-s to achieve a 10-fold saving of computation steps. If an efficient and stable numerical differentiation algorithm can be employed, another optimality criterion such as D or E optimality could be used in the computation of the Fisher Information metric which might help reduce the number of steps. Knowing the fact that the optimal stimulus amplitudes tend to reach the upper boundary, with a certain probability we could set the amplitudes directly to the boundary as the starting point for optimization. The optimization algorithms could be further fine-tuned and streamlined to save time. It is also possible to save time by early stopping of the optimization process and not optimizing the parameters completely. Taking into account of all these factors considered above, we could conceivably achieve a speedup factor of 10 or more.

Several key computational procedures are highly parallelizable, such as multiple initial guesses for optimization. The stimuli could also be generated in parallel on multiple, independent processors as a block instead of one by one on a single processor. Generating the stimuli sequentially on multiple processors using past response data as if they were moving on an assembly line could also save time by a factor of 10 or more. Employment of larger cluster computing systems with faster processors and a large number of compute cores could potentially speed up the computation but this option is very expensive. A much more economical alternative is to take advantage of GPU computing which has been widely employed in the field of machine learning. If we could achieve a speedup factor of 10 or more by parallel computing and block design, the speed of our optimal design method would start to become relevant for real experiments.

Although the simulations reported here were focused on the simplest model which was adequate for our main conclusions within the scope of this paper, we emphasize that our method is readily generalizable to more complex network models. It should be straightforward to extend our derivations of the utility functions and the derivative equations to more complex networks with rate-based dynamics (Equation 1) because the method itself does not have an inherent limitation in this regard. We noted that even for such a simple network, some parameters, especially those related to the inhibitory unit which was not directly observed, were already quite hard to estimate, and the relative errors remained sizable even with the help of optimal design which worked significantly better than nonadaptive random stimuli. This result confirms that network parameter estimation is intrinsically a hard problem even for a simple network. For complex networks with more free parameters, it is reasonable to expect that the parameter estimation problem could become much harder. For models with more complex dynamics than the rate model, such as conductance-based models, the optimal design method needs to be adapted. The key modification would be the likelihood function of the spike train (Equation 11), which depends crucially on how noise is introduced into the model. Our method becomes irrelevant if the situation involves only deterministic spike trains. On the other hand, if the spikes can be well approximated by an inhomogeneous Poisson process, then our method could still be useful. Moreover, the method of gradient computation by solving differential equations is still a useful strategy because in all cases considered above the dynamics can always be described by a system of ordinary differential equations.

Advances in the technology for simultaneous recording from many neurons could increase the amount of information available about the underlying networks (Buzsáki, 2004; Cai et al., 2016; Jun et al., 2017) and thus potentially alleviate the parameter estimation problem in future online experiments. As mentioned before, optimal design methods have already been used in experiments on auditory neurons in the inferior colliculus (Dekel, 2012; Tam, 2012) and the auditory cortex (Feng et al., 2012; Feng, 2013). Those experiments were based on feedforward networks without temporal dynamics, and the spectral amplitudes of stationary sound stimuli were optimized to improve parameter estimation. The dynamic network models considered in this paper could be used to accommodate the transient response properties of auditory neurons such as the stereotypical time course with both transient and sustained response components (de la Rocha et al., 2008) (see also **Figure 2**). The computational study reported in this paper presents a necessary first step for extending the optimal design method to this type of dynamic network models for sensory neurons. In future studies it would be useful to apply the optimal design method to networks with a larger number of neurons in situations where the response data are collected simultaneously from multiple neurons, including both excitatory neurons and inhibitory neurons. More complex stimulus structures could be utilized. Besides the estimation of network parameters such as connection weights and the time constants, one may also estimate intrinsic parameters such as firing thresholds and slopes. Additional realistic details such as plasticity and adaptation could also be included. Although these generalizations require more computational power, the steady increase of accessible computational capacities over the years especially in parallel computing should make the optimal design method increasingly more feasible for real online experiments.

AUTHOR CONTRIBUTIONS

RD: performed the computations, prepared the tables, and wrote the first draft of manuscript. KZ: contributed to the theoretical development, polished up the figures, and finalized of the manuscript.

ACKNOWLEDGMENTS

Supported by Turkish Scientific and Technological Research Council (TÜBİTAK) 2219 Research Program and NIH grant R01 DC013698. All the computer codes in Matlab for the simulations in this paper are freely available upon request for scientific research and education or any noncommercial use.

REFERENCES

Amari, S.-I. (1972). Characteristics of random nets of analog neuron-like elements. *IEEE Trans. Syst. Man Cybern.* SMC-2, 643–657. doi: 10.1109/TSMC.1972.4309193

Atkinson, A., and Donev, A. (1992). *Optimum Experimental Designs*. Oxford University Press.

Beer, R. D. (1995). On the dynamics of small continuous-time recurrent neural networks. *Adapt. Behav.* 3, 469–509. doi: 10.1177/105971239500300405

Benda, J., Gollisch, T., Machens, C. K., and Herz, A. V. (2007). From response to stimulus: adaptive sampling in sensory physiology. *Curr. Opin. Neurobiol.* 17, 430–436. doi: 10.1016/j.conb.2007.07.009

Bleeck, S., Patterson, R. D., and Winter, I. M. (2003). Using genetic algorithms to find the most effective stimulus for sensory neurons. *J. Neurosci. Methods* 125, 73–82. doi: 10.1016/S0165-0270(03)00040-2

Bölinger, D., and Gollisch, T. (2012). Closed-loop measurements of iso-response stimuli reveal dynamic nonlinear stimulus integration in the retina. *Neuron* 73, 333–346. doi: 10.1016/j.neuron.2011.10.039

Brown, E. N., Barbieri, R., Ventura, V., Kass, R. E., and Frank, L. M. (2002). The time-rescaling theorem and its application to neural spike train data analysis. *Neural Comput.* 14, 325–346. doi: 10.1162/08997660252741149

Buzsáki, G. (2004). Large-scale recording of neuronal ensembles. *Nat. Neurosci.* 7, 446–451. doi: 10.1038/nn1233

Cai, D. J., Aharoni, D., Shuman, T., Shobe, J., Biane, J., Song, W., et al. (2016). A shared neural ensemble links distinct contextual memories encoded close in time. *Nature* 534, 115–118. doi: 10.1038/nature17955

Carlson, E. T., Rasquinha, R. J., Zhang, K., and Connor, C. E. (2011). A sparse object coding scheme in area v4. *Curr. Biol.* 21, 288–293. doi: 10.1016/j.cub.2011.01.013

Cavagnaro, D. R., Myung, J. I., Pitt, M. A., and Kujala, J. V. (2010). Adaptive design optimization: a mutual information-based approach to model discrimination in cognitive science. *Neural Comput.* 22, 887–905. doi: 10.1162/neco.2009.02-09-959

Chambers, A. R., Hancock, K. E., Sen, K., and Polley, D. B. (2014). Online stimulus optimization rapidly reveals multidimensional selectivity in auditory cortical neurons. *J. Neurosci.* 34, 8963–8975. doi: 10.1523/JNEUROSCI.0260-14.2014

Chichilnisky, E. (2001). A simple white noise analysis of neuronal light responses. *Network* 12, 199–213. doi: 10.1080/713663221

deCharms, R. C., Blake, D. T., and Merzenich, M. M. (1998). Optimizing sound features for cortical neurons. *Science* 280, 1439–1444. doi: 10.1126/science.280.5368.1439

de la Rocha, J., Marchetti, C., Schiff, M., and Reyes, A. D. (2008). Linking the response properties of cells in auditory cortex with network architecture: cotuning versus lateral inhibition. *J. Neurosci.* 28, 9151–9163. doi: 10.1523/JNEUROSCI.1789-08.2008

DeGroot, M. H. (1970). *Optimal Statistical Decisions*. New York, NY: McGraw-Hill Company.

Dekel, E. (2012). *Adaptive On-line Modeling in the Auditory System: In vivo Implementation of the Optimal Experimental Design Paradigm*. Master's thesis, Department of Biomedical Engineering, Johns Hopkins University, Baltimore, MD.

DiMattina, C., and Zhang, K. (2010). How to modify a neural network gradually without changing its input-output functionality. *Neural Comput.* 22, 1–47. doi: 10.1162/neco.2009.05-08-781

DiMattina, C., and Zhang, K. (2011). Active data collection for efficient estimation and comparison of nonlinear neural models. *Neural Comput.* 23, 2242–2288. doi: 10.1162/NECO_a_00167

DiMattina, C., and Zhang, K. (2013). Adaptive stimulus optimization for sensory systems neuroscience. *Front. Neural Circuits* 7:101. doi: 10.3389/fncir.2013.00101

Doruk, R. O., and Zhang, K. (2018). Fitting of dynamic recurrent neural network models to sensory stimulus-response data. *J. Biol. Phys.* 44, 449–469. doi: 10.1007/s10867-018-9501-z

Eden, U. T. (2008). "Point process models for neural spike trains," in *Neural Signal Processing: Quantitative Analysis of Neural Activity*, ed P. Mitra (Washington, DC: Society for Neuroscience), 45–51.

El Hady, A. (2016). *Closed Loop Neuroscience*. London: Academic Press.

Fedorov, V. V., and Leonov, S. L. (2013). *Optimal Design for Nonlinear Response Models*. Boca Raton, FL: CRC Press.

Feng, L. (2013). *Spectral Integration and Neural Representation of Harmonic Complex Tones in the Primate Auditory Cortex*. PhD thesis, Department of Biomedical Engineering, Johns Hopkins University, Baltimore, MD.

Feng, L., Dekel, E., Zhang, K., and Wang, X. (2012). Online adaptive stimulus design for studying non-linear spectral integration in auditory cortex. *Soc. Neurosci. Abstr.* 564.10.

Flila, S., Dufour, P., Hammouri, H., and Nadri, M. (2010). "A combined closed loop optimal design of experiments and online identification control approach," in *Proceedings of the 29th Chinese Control Conference* (Beijing: IEEE), 1178–1183.

Gibbons, J. D., and Chakraborti, S. (2011). *Nonparametric Statistical Inference*. Boca Raton, FL: CRC Press.

Gollisch, T., and Herz, A. V. (2012). The iso-response method: measuring neuronal stimulus integration with closed-loop experiments. *Front. Neural Circuits* 6:104. doi: 10.3389/fncir.2012.00104

Hancock, K. E., Davis, K. A., and Voigt, H. F. (1997). Modeling inhibition of type ii units in the dorsal cochlear nucleus. *Biol. Cybern.* 76, 419–428. doi: 10.1007/s004220050355

Hancock, K. E., and Voigt, H. F. (1999). Wideband inhibition of dorsal cochlear nucleus type iv units in cat: a computational model. *Ann. Biomed. Eng.* 27, 73–87. doi: 10.1114/1.150

Harth, E., and Tzanakou, E. (1974). Alopex: a stochastic method for determining visual receptive fields. *Vis. Res.* 14, 1475–1482. doi: 10.1016/0042-6989(74)90024-8

Hodgkin, A. L., and Huxley, A. F. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. *J. Physiol.* 117:500. doi: 10.1113/jphysiol.1952.sp004764

Hollander, M., Wolfe, D. A., and Chicken, E. (2014). *Nonparametric Statistical Methods, 3rd Edn*. Hoboken, NJ: John Wiley & Sons.

Hopfield, J. J. (1984). Neurons with graded response have collective computational properties like those of two-state neurons. *Proc. Natl. Acad. Sci. U.S.A.* 81, 3088–3092. doi: 10.1073/pnas.81.10.3088

Horwitz, G. D., and Hass, C. A. (2012). Nonlinear analysis of macaque v1 color tuning reveals cardinal directions for cortical color processing. *Nat. Neurosci.* 15, 913–919. doi: 10.1038/nn.3105

Jun, J. J., Steinmetz, N. A., Siegle, J. H., Denman, D. J., Bauza, M., Barbarits, B., et al. (2017). Fully integrated silicon probes for high-density recording of neural activity. *Nature* 551, 232–236. doi: 10.1038/nature24636

Ledoux, E., and Brunel, N. (2011). Dynamics of networks of excitatory and inhibitory neurons in response to time-dependent inputs. *Front. Comput. Neurosci.* 5:25. doi: 10.3389/fncom.2011.00025

Lewi, J., Butera, R., and Paninski, L. (2009). Sequential optimal design of neurophysiology experiments. *Neural Comput.* 21, 619–687. doi: 10.1162/neco.2008.08-07-594

Lewis, P. A., and Shedler, G. S. (1979). Simulation of nonhomogeneous poisson processes by thinning. *Naval Res. Logist. Q.* 26, 403–413. doi: 10.1002/nav.3800260304

Machens, C. K., Gollisch, T., Kolesnikova, O., and Herz, A. V. (2005). Testing the efficiency of sensory coding with optimal stimulus ensembles. *Neuron* 47, 447–456. doi: 10.1016/j.neuron.2005.06.015

Mann, H. B., and Whitney, D. R. (1947). On a test of whether one of two random variables is stochastically larger than the other. *Ann. Math. Stat.* 18, 50–60. doi: 10.1214/aoms/1177730491

Marmarelis, P. Z., and Marmarelis, V. Z. (1978). *Analysis of Physiological Systems: The White-Noise Approach*. New York, NY: Plenum Press.

Miller, K. D., and Fumarola, F. (2012). Mathematical equivalence of two common forms of firing rate models of neural networks. *Neural Comput.* 24, 25–31. doi: 10.1162/NECO_a_00221

Myung, I. J. (2003). Tutorial on maximum likelihood estimation. *J. Math. Psychol.* 47, 90–100. doi: 10.1016/S0022-2496(02)00028-7

Nelken, I., Prut, Y., Vaadia, E., and Abeles, M. (1994). In search of the best stimulus: an optimization procedure for finding efficient stimuli in the cat auditory cortex. *Hear. Res.* 72, 237–253. doi: 10.1016/0378-5955(94)90222-4

O'Connor, K. N., Petkov, C. I., and Sutter, M. L. (2005). Adaptive stimulus optimization for auditory cortical neurons. *J. Neurophysiol.* 94, 4051–4067. doi: 10.1152/jn.00046.2005

Paninski, L. (2005). Asymptotic theory of information-theoretic experimental design. *Neural Comput.* 17, 1480–1507. doi: 10.1162/0899766053723032

Pillow, J. W., Shlens, J., Paninski, L., Sher, A., Litke, A. M., Chichilnisky, E. J., et al. (2008). Spatio-temporal correlations and visual signalling in a complete neuronal population. *Nature* 454, 995–999. doi: 10.1038/nature07140

Potter, S. M., El Hady, A., and Fetz, E. E. (2014). Closed-loop neuroscience and neuroengineering. *Front. Neural Circuits* 8:115. doi: 10.3389/fncir.2014.00115

Prinz, A. A., Bucher, D., and Marder, E. (2004). Similar network activity from disparate circuit parameters. *Nat. Neurosci.* 7, 1345–1352. doi: 10.1038/nn1352

Pukelsheim, F. (1993). *Optimal Design of Experiments*. Philadelphia, PA: SIAM.

Shadlen, M. N., and Newsome, W. T. (1994). Noise, neural codes and cortical organization. *Curr. Opin. Neurobiol.* 4, 569–579. doi: 10.1016/0959-4388(94)90059-0

Simoncelli, E. P., Paninski, L., Pillow, J., and Schwartz, O. (2004). "Characterization of neural responses with stochastic stimuli," in *The Cognitive Neurosciences III*, ed M. Gazzaniga (Cambridge, MA: MIT press), 327–338.

Snyder, D. L., and Miller, M. I. (1991). *Random Point Processes in Space and Time, 2nd Edn*. New York, NY: Springer-Verlag.

Sugiyama, M., and Rubens, N. (2008). A batch ensemble approach to active learning with model selection. *Neural Netw.* 21, 1278–1286. doi: 10.1016/j.neunet.2008.06.004

Tam, W. (2012). *Adaptive Modeling of Marmoset Inferior Colliculus Neurons in Vivo*. PhD thesis, Department of Biomedical Engineering, Johns Hopkins University, Baltimore, MD.

Telen, D., Logist, F., Van Derlinden, E., and Van Impe, J. (2012a). "Approximate robust optimal experiment design in dynamic bioprocess models," in *20th Mediterranean Conference on Control & Automation* (Barcelona: IEEE), 157–162.

Telen, D., Logist, F., Van Derlinden, E., and Van Impe, J. F. (2012b). "Robust optimal experiment design: a multi-objective approach," in *MATHMOD 7th Vienna International Conference on Mathematical Modelling,* eds I. Troch and F. Breitenecker (Vienna: ARGESIM), 689–694.

Theunissen, F. E., David, S. V., Singh, N. C., Hsu, A., Vinje, W. E., and Gallant, J. L. (2001). Estimating spatio-temporal receptive fields of auditory and visual neurons from their responses to natural stimuli. *Network* 12, 289–316. doi: 10.1080/net.12.3.289.316

Van Hemmen, J. L., and Sejnowski, T. J. (2005). *23 Problems in Systems Neuroscience*. Oxford, UK: Oxford University Press.

Wang, X., Lu, T., Snider, R. K., and Liang, L. (2005). Sustained firing in auditory cortex evoked by preferred stimuli. *Nature* 435, 341–346. doi: 10.1038/nature03565

Westwick, D. T., and Kearney, R. E. (2003). *Identification of Nonlinear Physiological Systems*. Piscataway, NJ: IEEE Press.

Wilson, H. R., and Cowan, J. D. (1972). Excitatory and inhibitory interactions in localized populations of model neurons. *Biophys. J.* 12, 1–24. doi: 10.1016/S0006-3495(72)86068-5

Wu, M. C., David, S. V., and Gallant, J. L. (2006). Complete functional characterization of sensory neurons by system identification. *Annu. Rev. Neurosci.* 29, 477–505. doi: 10.1146/annurev.neuro.29.051605.113024

Yamane, Y., Carlson, E. T., Bowman, K. C., Wang, Z., and Connor, C. E. (2008). A neural code for three-dimensional object shape in macaque inferotemporal cortex. *Nat. Neurosci.* 11, 1352–1360. doi: 10.1038/nn.2202

Zhang, K. (2014). How to compress sequential memory patterns into periodic oscillations: general reduction rules. *Neural Comput.* 26, 1542–1599. doi: 10.1162/NECO_a_00618

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.