Neural Networks 22 (2009) 1340–1351

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

2009 Special Issue

Nonlinear modeling of neural population dynamics for hippocampal prostheses

Dong Song[a,b,*], Rosa H.M. Chan[a,b], Vasilis Z. Marmarelis[a], Robert E. Hampson[d], Sam A. Deadwyler[d], Theodore W. Berger[a,b,c]

[a] Department of Biomedical Engineering, University of Southern California, Los Angeles, CA 90089, USA
[b] Center for Neural Engineering, University of Southern California, Los Angeles, CA 90089, USA
[c] Program in Neuroscience, University of Southern California, Los Angeles, CA 90089, USA
[d] Department of Physiology & Pharmacology, Wake Forest University, School of Medicine, Winston-Salem, NC 27157, USA

ARTICLE INFO

Article history:
Received 11 March 2009
Received in revised form 18 April 2009
Accepted 17 May 2009

Keywords:
Hippocampus
Spike
Spatio-temporal pattern
Volterra kernel
Feedback
Multiple-input multiple-output system

ABSTRACT

Developing a neural prosthesis for the damaged hippocampus requires restoring the transformation of population neural activities performed by the hippocampal circuitry. To bypass a damaged region, output spike trains need to be predicted from the input spike trains and then reinstated through stimulation. We formulate a multiple-input, multiple-output (MIMO) nonlinear dynamic model for the input–output transformation of spike trains. In this approach, a MIMO model comprises a series of physiologically-plausible multiple-input, single-output (MISO) neuron models that consist of five components each: (1) feedforward Volterra kernels transforming the input spike trains into the synaptic potential, (2) a feedback kernel transforming the output spikes into the spike-triggered after-potential, (3) a noise term capturing the system uncertainty, (4) an adder generating the pre-threshold potential, and (5) a threshold function generating output spikes. It is shown that this model is equivalent to a generalized linear model with a *probit* link function. To reduce model complexity and avoid overfitting, statistical model selection and cross-validation methods are employed to choose the significant inputs and interactions between inputs. The model is applied successfully to the hippocampal CA3–CA1 population dynamics. Such a model can serve as a computational basis for the development of hippocampal prostheses.

1. Introduction

One of the fundamental principles of cortical brain regions, including the hippocampus, is that information is represented in the ensemble firing of populations of neurons, i.e., spatio-temporal patterns of electrophysiological activity (Deadwyler & Hampson, 1995; Georgopoulos, Schwartz, & Kettner, 1986; Pouget, Dayan, & Zemel, 2003; Puchalla, Schneidman, Harris, & Berry, 2005; Schwartz, Kettner, & Georgopoulos, 1988). The hippocampus has long been known to be responsible for the formation of declarative, or fact-based, memories (Burgess, Maguire, & O'Keefe, 2002; Deadwyler, Bunn, & Hampson, 1996; Deadwyler & Hampson, 1995; Eichenbaum, 1999; Squire, 1992). Damage to the hippocampus disrupts the propagation of spatio-temporal patterns of activity through the hippocampal internal circuitry, resulting in a severe anterograde amnesia. Developing a neural prosthesis for the damaged hippocampus requires restoring this multiple-input, multiple-output (MIMO) transformation of spatio-temporal patterns of activity (Fig. 1(A)). Because the mechanisms underlying synaptic transmission and generation of electrical activity in neurons are inherently nonlinear, any such prosthesis must be based on a nonlinear MIMO model. We developed a nonlinear dynamic model to describe the spike train-to-spike train transformations in the brain. This model allowed the prediction of spatio-temporal patterns in a downstream brain region (e.g., hippocampal CA1) based on the spatio-temporal patterns in an upstream brain region (e.g., hippocampal CA3), and served as the computational basis for the development of hippocampal prostheses.

In this approach, a MIMO model consists of a series of multiple-input, single-output (MISO) models. Each MISO model can be considered as a modified Volterra kernel model describing the nonlinear dynamic transformation from the input (e.g., CA3) spike trains that one (CA1) neuron receives to the output spike train of this neuron. With progressively higher order Volterra kernel terms, this model can capture arbitrarily high order nonlinearities within each input and nonlinear interactions between inputs as they affect the output (Song et al., 2007). In a real biological

[*] Corresponding address: 403 Hedco Neuroscience Building, University of Southern California, Los Angeles, CA 90089, USA. Tel.: +1 213 740 8063; fax: +1 213 740 5687.

E-mail addresses: dsong@usc.edu (D. Song), homchan@usc.edu (R.H.M. Chan), vzm@bmsr.usc.edu (V.Z. Marmarelis), rhampson@wfubmc.edu (R.E. Hampson), sdeadwyl@wfubmc.edu (S.A. Deadwyler), berger@bmsr.usc.edu (T.W. Berger).

0893-6080/$ – see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.neunet.2009.05.004

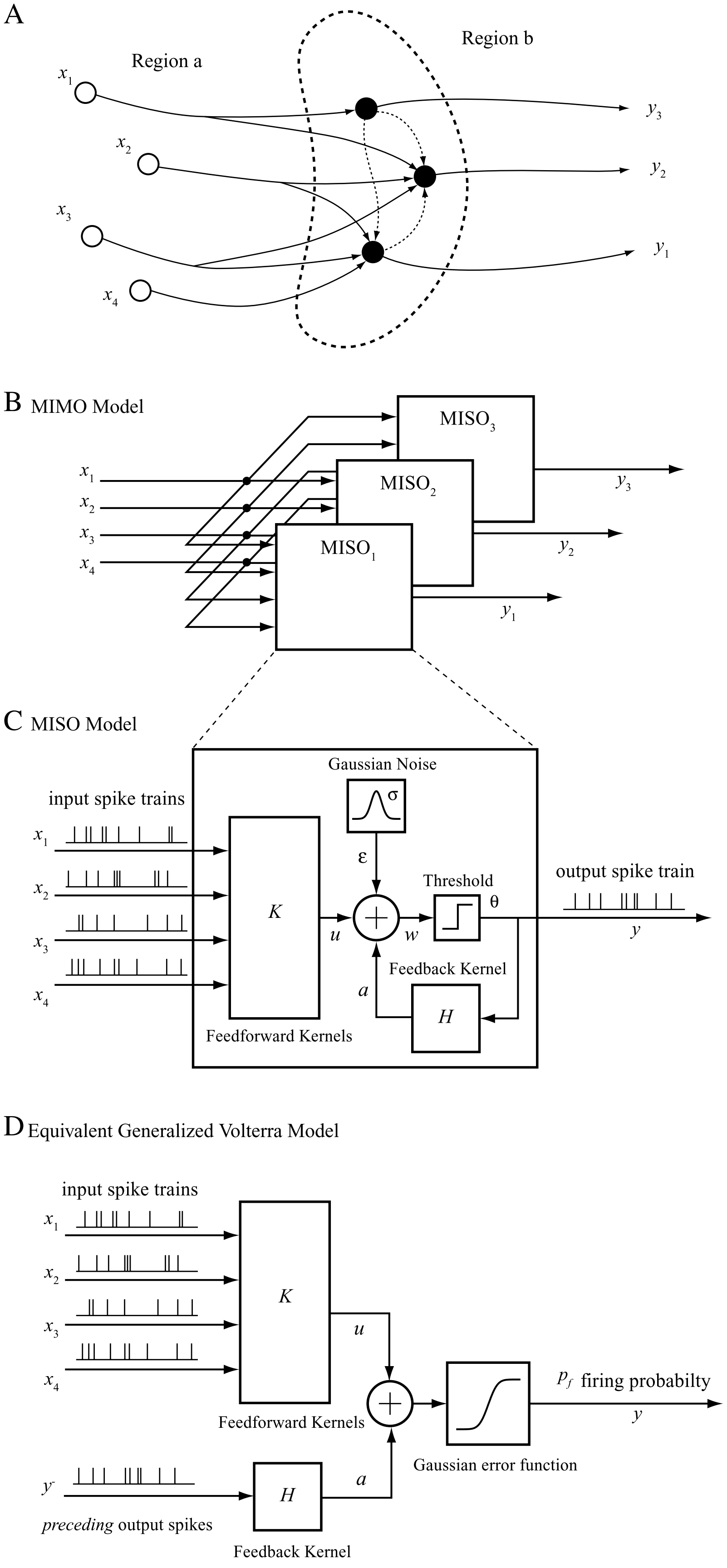

Fig. 1. Multiple-input multiple-output (MIMO) model for population neural dynamics. (A) schematic diagram of spike train propagation between two brain regions. (B) MIMO model as a series of multiple-input single-output (MISO) models. (C) structure of a MISO model. (D) MISO model is equivalent to a generalized Volterra model (GVM) with a *probit* link function.

Table 1
Number of Coefficient to be estimated in MISO models.

	k_0	k_1	k_{2s}	k_{2x}	h	Total
V	1	NM_k 12 000	$NM_k(M_k+1)/2$ 3 006 000	$N(N-1)M_kM_k/2$ 69 000 000	M_h 300	72 018 301
LV	1	NL 72	$NL(L+1)/2$ 144	$N(N-1)L^2/2$ 2 484	L 3	2 704
rLV	1	18	36	9	3	67

V: Volterra kernel model. LV: Laguerre expansion of Volterra kernel model. rLV: reduced LV model with only significant inputs and cross-terms. The numbers are calculated with: $N = 24$, $L = 3$, $M_k = 500$, $M_h = 300$. Numbers of rLV models are from an example with six significant inputs and one significant cross-term.

system, however, an output may not be affected by every input, and all inputs may not nonlinearly interact with each other. In such circumstances, the full Volterra kernel model is not the most efficient way of system representation. In addition, the number of coefficients to be estimated in a full Volterra kernel model grows rapidly with the number of inputs and the model order. Estimation of such a model, especially one with higher order terms, can easily become unwieldy. Furthermore, a model with too many open parameters (coefficients) eventually tends to fit the noise instead of the signal in the training data. A model with overfitted parameters results in (1) poor generalization of the training data and, (2) bad predictions for novel data. Interpretation of such a model becomes problematic.

To solve this problem, we further established a rigorous statistical method of reducing/optimizing the structure of a MIMO model of spike train transformation, and demonstrate how such an approach can be applied to the specific MIMO structure. Specifically, we applied statistical model selection methods to the configuration of our modified Volterra kernel models. The model complexity was reduced by (1) selecting the inputs, i.e., self-terms, that significantly contributed to the output, (2) selecting the interactions between inputs, i.e., cross-terms, that had significant contributions to the output. Results showed that the reduced kernel model, which contained much fewer coefficients, had maximal prediction power for novel data. The reduced kernels can be more reliably used for further analysis of the functional connections/interactions between neurons.

2. Nonlinear model of population neural dynamics

2.1. Model configuration

A multiple-input, multiple-output (MIMO) model is broken down into a series of multiple-input, single-output (MISO) models (Fig. 1(B)). Within each MISO model, the output spike train is predicted based on all the input spike trains. The MISO model structure is inspired by the electrophysiological properties of single spiking neurons and consists of five components (Fig. 1(C)): (1) a feedforward block K transforming the input spike trains x to a continuous hidden variable u that can be interpreted as the synaptic potential, (2) a feedback block H transforming the *preceding* output spikes to a continuous hidden variable a that can be interpreted as an after-potential, (3) a noise term ε that captures the system uncertainty caused by both the intrinsic neuronal noise and the unobserved inputs, (4) an adder generating a continuous hidden variable w that can be interpreted as a pre-threshold potential, and (5) a threshold function generating output spike when the value of w crosses θ (Song et al., 2007).

The model can be expressed by the following equations:

$$w = u(k, x) + a(h, y) + \varepsilon(\sigma) \tag{1}$$

$$y = \begin{cases} 0 & \text{when } w < \theta \\ 1 & \text{when } w \geq \theta. \end{cases} \tag{2}$$

K and H can take any mathematical form as long as it sufficiently captures the nonlinear dynamic transformations between the variables, i.e., x to u and y to a. In our approach, K takes the form of a Volterra model, in which u is expressed in terms of the inputs x by means of the Volterra series expansion as:

$$\begin{aligned} u(t) = k_0 &+ \sum_{n=1}^{N} \sum_{\tau=0}^{M_k} k_1^{(n)}(\tau) x_n(t-\tau) \\ &+ \sum_{n=1}^{N} \sum_{\tau_1=0}^{M_k} \sum_{\tau_2=0}^{M_k} k_{2s}^{(n)}(\tau_1, \tau_2) x_n(t-\tau_1) x_n(t-\tau_2) \\ &+ \sum_{n_1=1}^{N} \sum_{n_2=1}^{n_1-1} \sum_{\tau_1=0}^{M_k} \sum_{\tau_2=0}^{M_k} k_{2x}^{(n_1,n_2)}(\tau_1, \tau_2) \\ &\times x_{n_1}(t-\tau_1) x_{n_2}(t-\tau_2) + \cdots. \end{aligned} \tag{3}$$

The zeroth order kernel, k_0, is the value of u when the input is absent. First order kernels $k_1^{(n)}$ describe the linear relation between the nth input x_n and u, as functions of the time intervals (τ) between the present time and the past time. Second self-kernels $k_{2s}^{(n)}$ describe the second order nonlinear relation between the nth input x_n and u. Second order cross-kernels $k_{2x}^{(n_1,n_2)}$ describe the second order nonlinear interactions between each unique pair of inputs (x_{n_1} and x_{n_2}) as they affect u. N is the number of inputs. M_k denotes the memory length of the feedforward process. Higher order kernels, e.g., third order and fourth order kernels, are not shown in this equation.

Similarly, H takes the form of a first order Volterra model as in:

$$a(t) = \sum_{\tau=1}^{M_h} h(\tau) y(t-\tau) \tag{4}$$

where h is the linear feedback kernel. M_h is the memory length of the feedback process. (Note that τ starts from 1 instead of 0 to avoid predicting the current output with itself.)

One of the major challenges in Volterra modeling is the large number of open parameters (coefficients) to be estimated. The total number of open parameters increases exponentially with input dimension and model order. The MISO model in this study involves 2D (time and index of the input neurons) input and second order nonlinearity. The number of parameters easily becomes unwieldy even in a moderately large model (e.g., the 24-input MISO model shown in Table 1). To solve this problem, we employed (1) Laguerre expansion of Volterra kernel (LEV), and (2) statistical model selection techniques (see Section 2.3).

Using LEV, both k and h are expanded with orthonormal Laguerre basis functions b (Marmarelis, 1993, 2004; Song, Wang, Marmarelis, & Berger, 2009; Song & Wang, et al., 2009; Zanos et al., 2008). With input and output spike trains x and y convolved with b:

$$v_j^{(n)}(t) = \sum_{\tau=0}^{M_k} b_j(\tau) x_n(t-\tau), \tag{5}$$

$$v_j^{(h)}(t) = \sum_{\tau=1}^{M_h} b_j(\tau) y(t-\tau). \tag{6}$$

Eqs. (3) and (4) can be rewritten into:

$$u(t) = c_0 + \sum_{n=1}^{N}\sum_{j=1}^{L} c_1^{(n)}(j)v_j^{(n)}(t) + \sum_{n=1}^{N}\sum_{j_1=1}^{L}\sum_{j_2=1}^{j_1} c_{2s}^{(n)}(j_1,j_2)v_{j_1}^{(n)}(t)v_{j_2}^{(n)}(t) + \sum_{n_1=1}^{N}\sum_{n_2=1}^{n_1-1}\sum_{j_1=1}^{L}\sum_{j_2=1}^{L} c_{2x}^{(n_1,n_2)}(j_1,j_2)v_{j_1}^{(n)}(t)v_{j_2}^{(n)}(t) + \cdots \quad (7)$$

$$a(t) = \sum_{j=1}^{L} c_h(j)v_j^{(h)}(t). \quad (8)$$

$c_1^{(n)}$, $c_{2s}^{(n)}$, $c_{2x}^{(n_1,n_2)}$, and c_h are the sought Laguerre expansion coefficients of $k_1^{(n)}$, $k_{2s}^{(n)}$, $k_{2x}^{(n_1,n_2)}$, and h, respectively (c_0 is simply equal to k_0). Since the number of basis functions (L) can be made much smaller than the memory length (M_k and M_h), the number of open parameters is greatly reduced by the expansion.

The noise term ε is modeled as a Gaussian white noise with standard deviation σ.

2.2. Parameter estimation

With recorded input and output spike trains x and y, model parameters can be estimated using maximum-likelihood method. The negative log-likelihood function L can be expressed as:

$$L(y|x,k,h,\sigma,\theta) = -\sum_{t=0}^{T} \ln P(y|x,k,h,\sigma,\theta) \quad (9)$$

where T is the data length, and P is the probability of generating the recorded output y:

$$P(y|x,k,h,\sigma,\theta) = \begin{cases} \text{Prob}(w \geq \theta|x,k,h,\sigma,\theta) & \text{when } y = 1 \\ \text{Prob}(w < \theta|x,k,h,\sigma,\theta) & \text{when } y = 0. \end{cases} \quad (10)$$

Since ε is assumed to be Gaussian, the conditional firing probability intensity function P_f (the conditional probability of generating a spike, i.e., $\text{Prob}(w \geq \theta|x,k,h,\sigma,\theta)$ in Eq. (10)) at time t can be calculated with the Gaussian error function (integral of Gaussian function) erf:

$$P_f(t) = 0.5 - 0.5\, erf\left(\frac{\theta - u(t) - a(t)}{\sqrt{2}\sigma}\right) \quad (11)$$

where

$$erf(s) = \frac{2}{\sqrt{\pi}}\int_0^s e^{-t^2}\mathrm{d}t. \quad (12)$$

P at time t then can be calculated as:

$$P(t) = \begin{cases} p_f(t) & \text{when } y = 1 \\ 1 - p_f(t) & \text{when } y = 0 \end{cases} \quad (13)$$

or,

$$P(t) = 0.5 - [y(t) - 0.5]\, erf\left(\frac{\theta - u(t) - a(t)}{\sqrt{2}\sigma}\right). \quad (14)$$

Model coefficients c then can be estimated by minimizing the negative log-likelihood function L:

$$\tilde{c} = \arg\min(L(c)). \quad (15)$$

It is instructive to point out that the aforedescribed model is mathematically equivalent to a Generalized Linear Model (GLM) with y as a dependent variable, the convolutions of Laguerre basis functions with inputs x (v in Eqs. (7) and (8)) as well as the products of the these convolutions (vv in Eq. (7)) as independent variables, and c as unknown parameters (Fig. 1(D)). The GLM link function is the *probit* function (inverse cumulative distribution function of the normal distribution) since the latter is defined as:

$$probit(y) = \sqrt{2}erf^{-1}(2y - 1). \quad (16)$$

Given this important equivalence, model coefficients c and their covariance matrices can be estimated using the iterative re-weighted least-squares method, the standard algorithm for fitting GLMs (McCullagh & Nelder, 1989; Truccolo, Eden, Fellows, Donoghue, & Brown, 2005). For the same reason, this model can be termed as the Generalized Volterra Model (GVM) (Song et al., 2008). Since u, a and n are dimensionless variables, without loss of generality, both θ and σ can be set to unity; only c are estimated. σ is later restored and the θ value remains unity (see Section 2.4).

2.3. Model selection

Theoretically, the aforedescribed method can be used to estimate arbitrary MISO models. However, in practice, model complexity often needs to be reduced by selecting an optimal subset of model parameters (coefficients). This procedure, termed model selection, is particularly necessary and desirable in modeling the population neural dynamics for the following reasons: first, neurons are often sparsely connected. In a brain region, an output neuron is seldom affected by all the input neurons. The full Volterra kernel model as described in Eq. (7) is not the most efficient or interpretable way of representing such a system. More importantly, the number of coefficients to be estimated in a full Volterra kernel model grows rapidly with the number of inputs and the model order. Estimation of such a model, especially the higher order ones, can easily become unwieldy (Table 1). Furthermore, a model with too many open parameters (coefficients) tends to fit the noise instead of the signal in the training data. An overfitting model would result in poor generalization of the training data and bad predictions of the novel data. Consequently, interpretation of such a model becomes problematic. To solve this problem, the following statistical model selection method is applied to the configuration and estimation of the GVMs.

Before model selection, the input–output dataset is partitioned into two subsets. One subset (training set) is used for model estimation. The other subset (testing set) is retained for validation of the results from the training set. Results from the two subsets are called in-sample and out-of-sample results, respectively.

The model starts from the zeroth order. A zeroth order model only contains c_0, which is equivalent to the standard deviation of the pre-threshold Gaussian noise (see Section 2.4). It essentially model the system output as a homogeneous Poisson process (constant firing probability intensity). The minimal negative likelihood (L) of the zeroth order model provides a starting point for the model selection. In the second step, feedback terms (as described by Eq. (8)) are added to the model. Output spike train is predicted by the preceding output spikes without considering any input. If L decreases in both the in-sample and out-of-sample results, the feedback term is then added into the model. In the third step, inputs are selected using a forward step-wise selection procedure (Kutner, Nachtsheim, Neter, & Li, 2004). Self-terms involving first order and second order kernels are constructed for all inputs. With the zeroth order term and feedback term (if selected in the previous step) included in the model, the values of L with and without each input are calculated. The input that decreases L the most is then added into the model. With the newly selected input included into the model, selection is then performed on the remaining inputs. Repeating this procedure, inputs are sequentially added into the model. The selection is stopped when the out-of-sample L starts to increase (in-sample L always decreases with more terms included in the model), indicating the occurrence of overfitting. In the fourth step, cross-terms involving cross-kernel are selected. Cross-terms are first constructed for every unique pairs of the selected inputs and then selected following the aforedescribed forward step-wise and cross-validation procedures.

2.4. Kernel reconstruction and interpretation

The final coefficients $\hat{c}$ and $\hat{\sigma}$ can be obtained from estimated Laguerre expansion coefficients, $\tilde{c}$, with a simple normalization/conversion procedure:

$$\hat{c}_0 = 0, \tag{17}$$

$$\hat{c}_1^{(n)} = \frac{\tilde{c}_1^{(n)}}{1-\tilde{c}_0}, \tag{18}$$

$$\hat{c}_{2s}^{(n)} = \frac{\tilde{c}_{2s}^{(n)}}{1-\tilde{c}_0}, \tag{19}$$

$$\hat{c}_{2x}^{(n_1,n_2)} = \frac{\tilde{c}_{2x}^{(n_1,n_2)}}{1-\tilde{c}_0}, \tag{20}$$

$$\hat{c}_h = \frac{\tilde{c}_h}{1-\tilde{c}_0}, \tag{21}$$

$$\hat{\sigma} = \frac{1}{1-\tilde{c}_0}. \tag{22}$$

Feedforward and feedback kernels then can be reconstructed as:

$$\hat{k}_0 = 0, \tag{23}$$

$$\hat{k}_1^{(n)}(\tau) = \sum_{j=1}^{L} \hat{c}_1^{(n)}(j) b_j(\tau), \tag{24}$$

$$\hat{k}_{2s}^{(n)}(\tau_1,\tau_2) = \sum_{j_1=1}^{L}\sum_{j_2=1}^{j_1} \frac{\hat{c}_{2s}^{(n)}(j_1,j_2)}{2} \times \left[b_{j_1}(\tau_1)b_{j_2}(\tau_2) + b_{j_2}(\tau_1)b_{j_1}(\tau_2)\right], \tag{25}$$

$$\hat{k}_{2x}^{(n_1,n_2)}(\tau_1,\tau_2) = \sum_{j_1=1}^{L}\sum_{j_2=1}^{L} \hat{c}_{2x}^{(n_1,n_2)}(j_1,j_2) b_{j_1}(\tau_1) b_{j_2}(\tau_2), \tag{26}$$

$$\hat{h}(\tau) = \sum_{j=1}^{L} \hat{c}_h(j) b_j(\tau). \tag{27}$$

Threshold θ is equal to one in this normalized representation.

The normalized kernels provide an intuitive representation of the system input–output nonlinear dynamics. Single-pulse and paired-pulse response functions (r_1 and r_2) of each input can be derived as (Song, Marmarelis, & Berger, 2009):

$$r_1^{(n)}(\tau) = \hat{k}_1^{(n)}(\tau) + \hat{k}_{2s}^{(n)}(\tau,\tau), \tag{28}$$

$$\text{and} \quad r_2^{(n)}(\tau_1,\tau_2) = 2\hat{k}_{2s}^{(n)}(\tau_1,\tau_2); \tag{29}$$

$r_1^{(n)}$ is simply the response in u elicited by a single spike from the nth input neuron; $r_2^{(n)}$ describes the joint nonlinear effect of pairs of spikes from the nth input neuron in addition to the summation of their first order responses, i.e., $r_1^{(n)}(\tau_1) + r_1^{(n)}(\tau_2)$. $\hat{k}_{2x}^{(n_1,n_2)}(\tau_1,\tau_2)$ represents the joint nonlinear effect of pairs of spikes with one spike from neuron n_1 and one spike from neuron n_2. h represents the output spike-triggered after-potential on u (Fig. 2).

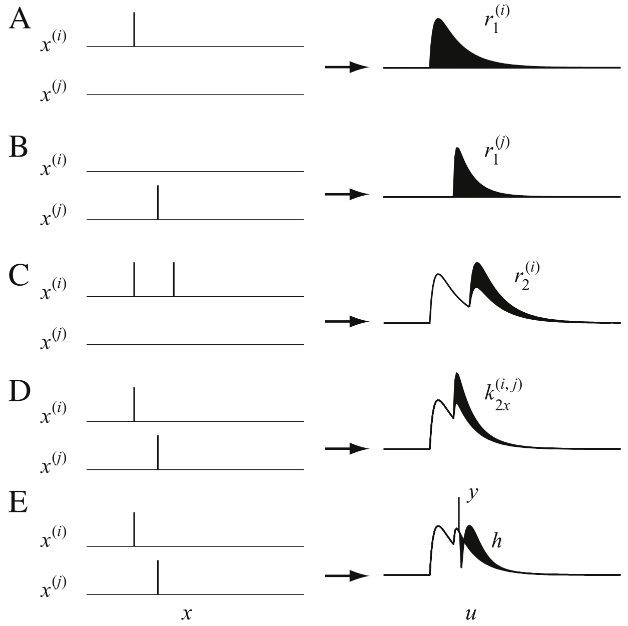

Fig. 2. Interpretations of the feedforward and feedback kernels. $r_1^{(i)}$ is the response in u elicited by a single spike from the ith input neuron; $r_2^{(i)}$ describes the joint nonlinear effect of pairs of spikes from the ith input neuron in addition to the linear summation of their first order responses. $k_{2x}^{(i,j)}$ represents the joint nonlinear effect of pairs of spikes from neurons i and j. h represents the output spike-triggered after-potential on u. Black areas: effect of each kernel on u.

2.5. Model validation and prediction

Two methods are used to evaluate the goodness-of-fit of the estimated GVM models. The first method directly evaluates the continuous firing probability intensity predicted by the model with the recorded output spike train. According to the time-rescaling theorem, an accurate model should generate a conditional firing intensity function P_f that can rescale the recorded output spike train into a Poisson process with unit rate (Brown, Barbieri, Ventura, Kass, & Frank, 2002; Song et al., 2007). By further variable conversion, inter-spike intervals should be rescaled into independent uniform random variables on the interval (0, 1). The model goodness-of-fit then can be assessed with a Kolmogorov–Smirnov (KS) test, in which the rescaled intervals are ordered from the smallest to the largest and then plotted against the cumulative distribution function of uniform density. If the model is correct, all points should lie on the 45° line of the KS plot within the 95% confidence bounds.

The second method quantifies the similarity between the recorded output spike train y and the predicted output spike train $\hat{y}$ after a smoothing process. First, $\hat{y}$ is realized through simulation: u is calculated with inputs x and the estimated feedforward kernels. This forms the deterministic part of the pre-threshold potential w. A Gaussian random sequence with standard deviation $\hat{\sigma}$ is then generated and added to u. This operation renders w stochastic. At each time t, if w crosses the threshold ($\theta = 1$), a spike is generated and added to $\hat{y}$, and a feedback process a is triggered and added to the future values of w. The calculation then moves on to time $t+1$ with updated w until it reaches the end of the trace. In the second step, point-process signals $\hat{y}$ and y are smoothed to continuous signals $\hat{y}_{\sigma_g}$ and y_{σ_g}, by convolving with a Gaussian kernel with standard deviation σ_g. Correlation coefficients r are then calculated as:

$$r(\sigma_g) = \left(\sum_{t=0}^{T} \hat{y}_{\sigma_g}(t) y_{\sigma_g}(t)\right) \Bigg/ \sqrt{\left(\sum_{t=0}^{T} y_{\sigma_g}(t) y_{\sigma_g}(t)\right)\left(\sum_{t=0}^{T} \hat{y}_{\sigma_g}(t) \hat{y}_{\sigma_g}(t)\right)}. \tag{30}$$

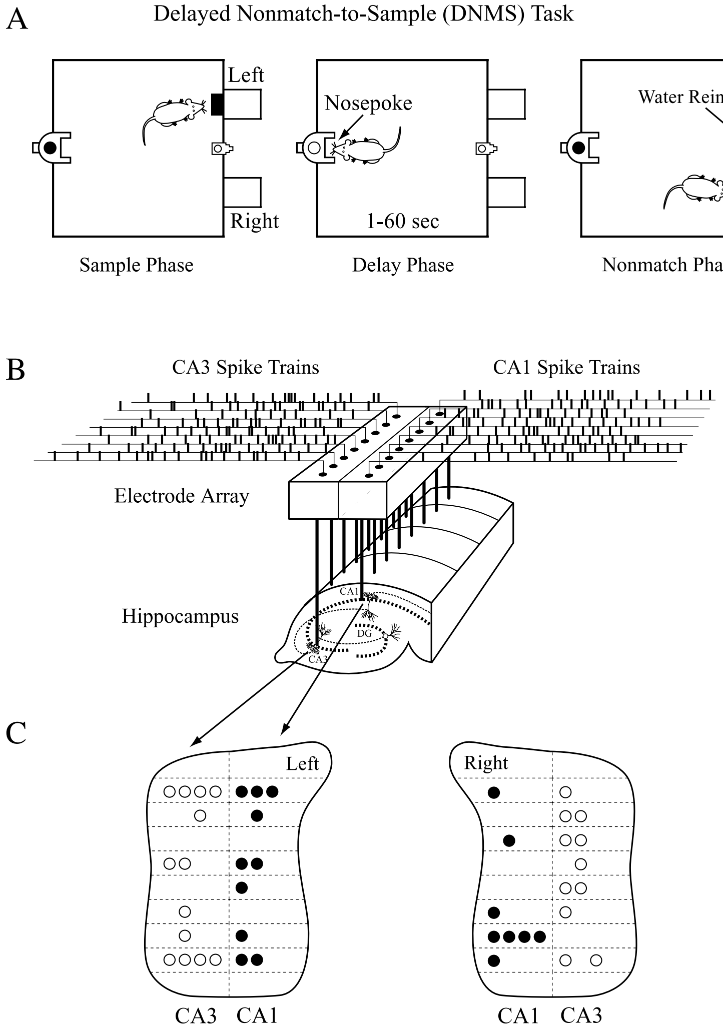

Fig. 3. Input and output spike trains recorded from hippocampal CA3 and CA1 regions during a delayed nonmatch-to-sample (DNMS) behavior task. (A) schematic diagram of the DNMS task. (B) CA3 and CA1 spike trains are recorded using a multi-electrode array during the task. (C) anatomical locations of input (CA3) and output (CA1) neurons indicated on a foldout map of the hippocampus. For this sample MIMO dataset, there are 24 inputs (white symbols) and 18 outputs (black symbols).

Because $\hat{y}_{\sigma_g}$ and y_{σ_g} are both positive vectors, r is a quantity between 0 and 1 that measures the similarity between $\hat{y}$ and y as a function of the "smoothness parameter" σ_g. σ_g essentially determines the temporal resolution used in comparing the predicted spike train with the actual spike train. A large value means low temporal resolution and a small value means high temporal resolution. This parameter does not influence the model estimation since the latter is done by maximizing the likelihood function defined with a fixed 2 ms bin size. In this study, σ_g varies from 2 to 100 ms. Mean and standard deviation of r are estimated with 32 trials of simulation.

3. Application on hippocampal CA3–CA1 population activity

3.1. Behavioral task

Male Long–Evans rats were trained to criterion on a two-lever, spatial delayed-nonmatch-to-sample (DNMS) task with randomly occurring variable delay intervals (Deadwyler et al., 1996; Hampson, Simeral, & Deadwyler, 1999). Animals performed the task by pressing (sample response) a single lever presented in one of the two positions in the sample phase (left or right); this event is called the "sample response". The lever was then retracted and the delay phase initiated; for the duration of the delay phase, the animal was required to nose-poke into a lighted device on the opposite wall. Following termination of the delay the nose-poke light was extinguished, both levers were extended and the animal was required to press the lever opposite to the sample lever; this act is called the "nonmatch response". If the correct lever was pressed, the animal was rewarded and the trial was completed (Fig. 3(A)).

3.2. Data preprocessing

Spike trains were obtained with multi-site recordings from different septo-temporal regions of the hippocampus of rats performing the DNMS task (Fig. 3(B)). For each hemisphere of the brain, an array of electrodes (microwires) was surgically implanted into the hippocampus, with 8 electrodes in the CA3 (input) region and 8 electrodes in the CA1 (output) region. Each electrode had the capacity of recording as many as 4 discriminable units (Fig. 3(C)). Spikes were sorted and timestamped with a 25 μs resolution. Datasets from 25 rats were analyzed. One to four sessions of recordings were selected from each rat. A session included approximately 100 successful DNMS tasks that each of which consisted of two of the four behavioral events, i.e., right sample (RS) and left nonmatch (LN), or left sample (LS) and right nonmatch (RN).

Spike trains were pre-screened based on mean firing rate and peri-event histogram. Neurons with mean firing rates in the

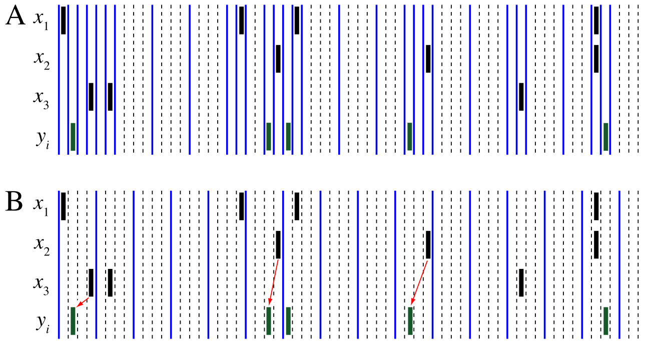

Fig. 4. Comparison of inter-spike binning and conventional binning for spike trains. Black bars: input spikes; green bars: output spikes. (A) with inter-spike binning (blue lines), the causal relations between input spikes and output spikes are retained. (B) with conventional binning (blue lines), future inputs may influence present outputs in analysis and cause spurious causal relation between inputs and outputs (red arrows). Dashed lines in A and (B) 2 ms bin.

range of 0.5 to 15 Hz and identifiable peri-event histogram were included in further analyses. Low and high mean rate recordings were rejected since they could represent artifacts or mixtures of action potentials. Both presumed principal (pyramidal) neurons and interneurons were included in the datasets analyzed. Peri-event (−2 s to +2 s) spike trains of the four behavioral events were extracted from each session and then concatenated to form the datasets.

First, the timestamp data are discretized with a 2 ms bin size. For such bin width, each bin contains a maximum of one spike event. To facilitate model estimation, the data length is further reduced with an inter-spike binning (ISB) method (Fig. 4). In ISB, bins that contain spikes are retained; bins within inter-spike intervals are merged with a bigger bin size. This method preserves the accurate timing of every spike while reducing the data length. In order to preserve the causal relation between input and output spikes, bins with an output spike are checked to see whether it also contains input spike(s). If there is no other spike, no operation is performed; if there is, the 2 ms bin is further divided between the input spike and the output spike. (The latter case is extremely rare due to the low firing rate of hippocampal neurons.) For most of the datasets, inter-spike bin size can be as large as 40 ms without introducing noticeable change to the coefficient estimates.

3.3. CA3–CA1 MIMO model

MIMO datasets consist of 6–32 CA3 spike trains and 5–24 CA1 spike trains, depending on the number of neurons recorded from each animal and the number of recorded neurons meeting the selection criteria. Each MIMO dataset is divided into MISO datasets for MISO model estimation. Thirty-two MIMO datasets from 25 rats were analyzed. Fig. 3(C) shows the anatomical locations of neurons included in one representative MIMO dataset (circles).

With the preprocessed dataset, the MISO model is obtained using the aforedescribed estimation and selection method (Sections 2.2–2.5 and 3). Fig. 5 illustrates the selection of inputs and cross-terms of a MISO model. As described in Section 2.3, the model starts from the zeroth order. The feedback term is then added into the model since it decreases L in both training and testing datasets (Fig. 5(A)). With the zeroth order and feedback terms, inputs are added into the model in a forward step-wise fashion (Fig. 5(B)). The in-sample L decreases monotonically as more terms are added into the model. However, the out-of-sample L starts to increase from the 7th input. Six inputs are then selected based on this cross-validation results (Fig. 5(A)). With the 6 selected inputs, 15 cross-terms in total are then constructed and added into the model (Fig. 5(C)). Again, in-sample L decreases monotonically; out-of-sample L starts to increase from the 2nd cross-term—so only one cross-term is selected by cross-validation (Fig. 5(A)). Table 1 summarizes the number of coefficients in a Volterra model, a Laguerre expansion of Volterra (LEV) model, and the reduced LEV after model selection. It is evident that the model selection procedure greatly reduces the model complexity and thus allows reliable estimation of the model.

The estimated and normalized kernels provide intuitive representations of the system input–output properties. Fig. 6 illustrates the first order and second order response functions for the 6 selected inputs, one second order cross-kernel for inputs No. 6 and No. 9, and the feedback kernel. These functions quantitatively describe how the synaptic potential is influenced by a single spike and pairs of spikes from a single input neuron or pairs of input neurons. The noise standard deviation is estimated to be 0.418, which reflects a 4.19 Hz spontaneous (baseline) firing rate of this CA1 neuron (see Eq. (11) for calculation) while the mean output firing rate is 7.44 Hz.

The estimated model is validated using the out-of-sample KS test. The summation of synaptic potential u and after-potential a is calculated using the input/output spike trains of the testing set, and the kernels estimated from the training set (Fig. 7(A), First row). Firing probability intensity P_f is then calculated using the Gaussian error function (Fig. 7(A), Second row). The KS plot shows that all data points are within the 95% confidence bound (Fig. 7(C)).

Output spike train is also predicted using the recursive simulation method described in Section 2.4. Input spike trains of the testing set and the kernels of the training set are used to predict and then compared with the output spike train of the testing set. Fig. 7(B) shows one realization of the output spike train with a Gaussian random noise path. The similarities between the predicted output spike train and the actual output spike train are quantified with correlation coefficients r. Fig. 7(D) illustrates the mean and standard deviation of r as functions of the standard deviation of the smoothing Gaussian kernel (σ_g). It shows that the model can generate output highly correlated with the actual output for a large range of σ_g.

Finally, all individually estimated and validated MISO models are concatenated to form the MIMO model for the hippocampal CA3–CA1 population neural dynamics. Fig. 8 compares the actual CA1 spatio-temporal pattern with the spatio-temporal pattern predicted by a MIMO model. Despite the difference in fine details (mostly due to the stochastic nature of the system), the MIMO model replicates the salient features of the actual CA1 spatio-temporal pattern.

4. Conclusions and discussion

We have formulated a Volterra kernel-based MIMO model for the population neural dynamics and applied it successfully to the modeling of hippocampal CA3–CA1 spike train transformation. The model inherits the capability of modeling nonlinear dynamic systems from the ordinary Volterra model while having several critical modifications/improvements made specific for the modeling of neural population activity. With the feedforward Volterra kernels, the model can capture the single-pulse and paired-pulse (nonlinear) dynamic effects of the input spikes to the output neuron and explicitly represents them in kernel functions. Different from an ordinary Volterra model (Song & Marmarelis, et al., 2009; Song & Wang, et al., 2009), the model includes hidden variables representing the internal states of the system. Point-process outputs (spikes) are considered as realizations of a firing probability intensity function determined by those hidden variables. This configuration allows simultaneous estimation of all model parameters, i.e., kernel coefficients and noise variance, from the input and output spike

Fig. 5. Selecting significant inputs and cross-terms of a MISO model. (A) the model starts from zeroth order (first data points). Feedback kernels are then selected and included in the model (second data points). Six inputs (first and second order self-terms) and one cross-term are selected based on cross-validation. Open circles: in-sample results; Closed circles: out-of-sample results; Top: absolute negative log-likelihood (L); Bottom: normalized L. (B) selection path of the inputs. (C) selecting path of the cross-terms. The 15 cross-terms are constructed with the selected 6 inputs only.

trains. The noise term and the threshold function can be considered as a "soft" threshold function, since they map to and generate output spikes based on the estimated continuous probability intensity function and thus avoid dichotomizing such continuous variable with a cut-point that highly depends on the relative frequencies of spikes (1s) and non-spikes (0s) in the outputs (Kutner et al., 2004). The resulting model structure is equivalent to a generalized linear model (GLM) with a *probit* link function and a binomial distribution function. Taking advantages of the concave likelihood function (L) and the well-established estimation methods of GLM, model parameters can be reliably (avoiding local minima) and efficiently estimated employing an iterative re-weighted least-squares method (Paninski, Pillow, & Simoncelli, 2004; Truccolo et al., 2005). Furthermore, a feedback kernel (h) is added to the model. This auto-regressive component captures the effects of the preceding output activities to the current output. Numerous evidence has shown that the output spike-triggered after-potential can profoundly influence the neuron's spiking activity (Alger & Nicoll, 1982; Berger, Chauvet, & Sclabassi, 1994; Gerstner & Kistler, 2002; Jolivet, Lewis, & Gerstner, 2004; Keat, Reinagel, Reid, & Meister, 2001; Paninski et al., 2004; Song, Wang, & Berger, 2002; Storm, 1987). As shown in our results (Fig. 5), the feedback component can account for a significant portion of the output variance. In our model selection scheme, the feedback term is not only included but also selected prior to the selection of inputs and cross-terms. This is akin to the Granger causality analysis, in which the moving-averages of inputs are statistically tested with the auto-regressive component built in (Nedungadi, Rangarajan, Jain, & Ding, 2009). Such a scheme differentiates the output-dependent dynamics from the input-dependent dynamics and thus provides more certain assessments of the causal relations between inputs and output.

The feedforward Volterra model is chosen to be second order in this study. However, with the expense of more open parameters and computational cost, the model can also include higher order terms, e.g., third order self-kernels and cross-kernels. Our previous analysis showed that adding third order self-kernels only marginally improved the model performance (Song et al., 2007).

The model described in this paper was developed within the context of hippocampal prostheses and CA3–CA1 population dynamics. For example, to replace a CA1 cell field selectively damaged in association with stroke – a common consequence of even brief periods of anoxia – the prosthetic device has to reinstate