An adaptive decoder from spike trains to micro-stimulation using kernel least-mean-square (KLMS) algorithm

Lin Li, Student Member, Il Memming Park, Sohan Seth, Student Member, John Choi, Joseph T. Francis, Justin C. Sanchez, Member, and José C. Príncipe, Fellow, IEEE

***Abstract*— This paper proposes a nonlinear adaptive decoder for somatosensory micro-stimulation based on the kernel least mean square (KLMS) algorithm applied directly on the space of spike trains. Instead of using a binned representation of spike trains, we transform the vector of spike times into a function in reproducing kernel Hilbert space (RKHS), where the inner product of two spike time vectors is defined by a nonlinear cross intensity kernel. This representation encapsulates the statistical description of the point process that generates the spike trains, and bypasses the tradeoff of dimensionality - resolution of the binned spike representations. We compare in synthetic experiment our method with two other methods based on bin data: GLM and KLMS. The results indicate that the KLMS based on RKHS for spike train is able to detect the timing, the shape and the amplitude of the biphasic stimulation with the best accuracy.**

I. Introduction

The rapid advance of micro-electrode arrays and electrophysiological recording techniques is promoting the simultaneous stimulation and recording of the spatiotemporal activities of hundreds of neurons. This opens up new opportunities to precisely predict stimulus or reconstruct movement by observing and modeling neural responses at the single neuron or population level. Quantifying the information contained in neural spike trains is conventionally called neural decoding, and it is a fundamental first step to design neural prosthetics and brain machine interfaces.

In order to effectively apply machine learning algorithms to neural decoding, a significant step is to define an appropriate input space for the decoder. A variety of machine learning techniques have been applied on the discrete representations of spike train (binned data), such as statistical methods [1], [2], kernel-based methods [3] etc. However, binned spike train representations impair the decoder effectiveness. Specifically, with a large bin size, a decoder will fail to reconstruct the stimuli or movement with the required time resolution. For example, the time resolution of the micro-stimulation signal is small (0.2ms), which requires a small bin size (less than the width of the stimulus pluses). With such small bin size, the input space becomes really sparse and dimensionality of the input space also becomes a problem. Alternatively, the vector of the spike times may be a more effective and accurate representation of spike trains. However, traditional machine learning algorithms cannot be directly implemented on the vector of spike times because it exists in a space that has no natural metric [4]. However it is still possible to define in the spike train space positive definite functions [4].

Based on the mathematical theory of reproducing kernel Hilbert space (RKHS), a suitable inner product between two spike trains such as the nonlinear cross intensity kernel, implicitly transforms a spike train into a continuous time function in the RKHS and opens up the possibility of applying any kernel-based machine learning algorithms directly on the spike time space.

Another challenge of neural decoding is the complexity and noise inherent in the spike train structure due to the organization of the neural circuits and the functional links between the neural assemblies and stimuli or movement [5]. It is essential that the decoder is capable of adaptively learn from the new data to track perturbations. Therefore, this paper proposes a nonlinear adaptive decoder with the KLMS algorithm [6] applied directly on the spike time input space, which obtains adaptive nonlinear regression with the linear sample-by-sample adaption algorithm in the RKHS without converging to local minima, and can effectively track perturbations.

In this paper, this adaptive KLMS-based decoder of spike train is applied to reconstruct continuous micro-stimulation on both synthetic and real data recorded from rat somatosensory cortex. Compared with two other methods with bin data: KLMS and Generalized Linear Model (GLM) [1], KLMS with spike kernel has the best performance for stimuli reconstruction.

II. Reproducing Kernel Hilbert Space (RKHS) for Spike Train

The kernel-based machine learning neural decoders are developed by defining a positive definite function in the spike train space that captures the probability law underlying the

This work was supported in part by the U.S. National Science Foundation under Grant CNS-0540304, NSF Partnerships for Innovation Program 0650161, and Darpa project N66001-10-C-2008.

Lin Li and S. Seth are with the Department of Electrical Engineering, University of Florida, Gainesville, Florida 32611. (email: linli@cnel.ufl.edu; sohan@cnel.ufl.edu)

Il Memming Park is with the Department of Biomedical Engineering, University of Florida, Gainesville, Florida 32611. (email: memming@cnel.ufl.edu)

John Choi and Joseph. T. Francis are with The Robert F. Furchgott Center for Neural & Behavioral Science And program in biomedical engineering at SUNY Downstate and NYU-Poly. (email: Joe.Francis@downstate.edu)

J. C. Sanchez is with the Department of Pediatrics, Division of Neurology, University of Florida, Gainesville, Florida 32611. (email: jcs77@ufl.edu)

J. C. Príncipe is with the Department of Electrical Engineering and Biomedical Engineering, University of Florida, Gainesville, Florida 32611. (principe@cnel.ufl.edu)

point process that generate the spike trains. Since a positive definite function also defines an inner product space the method also exploits the mathematical theory of reproducing kernel Hilbert space (RKHS). The inner product induces a nonlinear mapping on the space of the functional representation of spike trains.

A. Functional representation of spike trains

Spike trains are a representation of neural activity, which neglect the stereotypical shape of an action potential, but preserves only the time instants at which they occur (i.e., spike times) [7]. A spike train $\mathbf{s}$ is a sequence of ordered spike times i.e $\mathbf{s} = \{t_n \in \mathcal{T} : n = 1, \ldots, N\}$, in the interval $\mathcal{T} = [0, T]$. A spike train can be interpreted as a realization of an underlying stochastic point process [8]. With the assumption of Poisson process [9], the intensity function can be estimated from single realization, which transforms the space of spike trains to the space of integrable continuous functions with finite number of discontinuities [10]. According to kernel smoothing intensity estimation, a spike train $\mathbf{s}_i$ comprising of spike times $\{t^i_m \in \{m = 1, \ldots, N_i\}\}$ can be functionally represented by the estimated intensity function:

$$\hat{\lambda}_{\mathbf{s}_i}(t) = \sum_{m=1}^{N_i} h(t - t^i_m) \tag{1}$$

where h is the stereotypical evoked post-synaptic potential. Here

$$h(t - t^i_m) = \exp\left(-\frac{t - t^i_m}{\tau}\right) U(t - t^i_m). \tag{2}$$

where $U(t - t^i_m)$ represents the *Heaviside step function.*

B. The inner product of spike trains

The inner product between two spike trains is defined on the functional representation of spike times [10], which incorporates the statistics of the point processes directly. One possible inner product in the space of intensity function is defined by a nonlinear cross intensity kernel:

$$\kappa_\sigma(\mathbf{s}_i, \mathbf{s}_j) = \exp\left(-\frac{\|\lambda_{\mathbf{s}_i} - \lambda_{\mathbf{s}_j}\|^2}{\sigma^2}\right) \tag{3}$$

where σ is the kernel size and $\|\lambda_{\mathbf{s}_i} - \lambda_{\mathbf{s}_j}\|^2$ is the distance between two intensity functions defined by:

$$\begin{aligned}\|\lambda_{\mathbf{s}_i} - \lambda_{\mathbf{s}_j}\|^2 &= \langle \lambda_{\mathbf{s}_i} - \lambda_{\mathbf{s}_j}, \lambda_{\mathbf{s}_i} - \lambda_{\mathbf{s}_j} \rangle \\ &= \langle \lambda_{\mathbf{s}_i}, \lambda_{\mathbf{s}_i} \rangle + \langle \lambda_{\mathbf{s}_j}, \lambda_{\mathbf{s}_j} \rangle - 2\langle \lambda_{\mathbf{s}_i}, \lambda_{\mathbf{s}_j} \rangle \end{aligned} \tag{4}$$

where

$$\begin{aligned}\langle \lambda_{\mathbf{s}_i}, \lambda_{\mathbf{s}_j} \rangle &= \int_T \lambda_{\mathbf{s}_i} \lambda_{\mathbf{s}_j} dt \\ &= \sum_m^{N_i} \sum_n^{N_j} \left(\frac{-\tau}{2}\right) \exp\left(-\frac{|t^i_n - t^j_m|}{\tau}\right) \end{aligned} \tag{5}$$

Although the kernel is defined in the space of intensity functions, it is in fact a function of spike time vector and thus is a kernel function in the space of spike time. This kernel defines a RKHS for spike trains and induces a nonlinear mapping on the space of the intensity functions.

The spike train provides a precise description of neural firing pattern. However, collecting a single spike train from a neural systems is less robust and thus the signal structure has a large variability. Therefore, it is essential to utilize multichannel spike trains as inputs to enhance the decoder robustness. The kernel for multi-channel spike trains is defined here by the product kernel:

$$\kappa_\sigma(\mathbf{s}_i, \mathbf{s}_j) = \prod_{k=1}^{K} \exp\left(-\frac{\|\lambda_{\mathbf{s}^k_i} - \lambda_{\mathbf{s}^k_j}\|^2}{\sigma^2}\right) \tag{6}$$

where $\mathbf{s}^k_i$ is the spike train recorded from neuron k in window i.

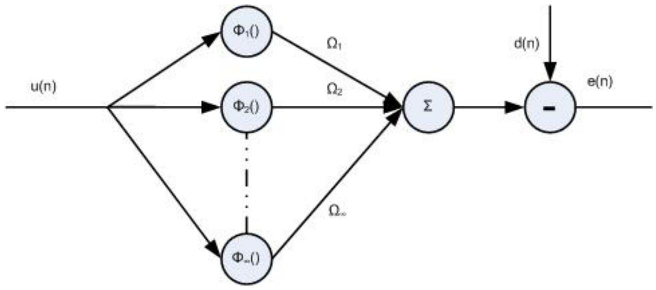


Fig. 1. **Network topology of kernel- least-mean-square algorithm.**

III. Kernel least mean square

In order to quantify the neural system dynamics, we utilize the kernel least mean square (KLMS) [6] algorithm to reconstruct the continuous stimulation. KLMS is an adaptive kernel-based algorithm developed in reproducing kernel Hilbert space (RKHS). The great appeal of kernel-based filters in RKHS is the usage of the linear structure of this space to implement well-established linear adaptive algorithms and to obtain nonlinear filter in the input space that leads to universal approximation capabilities without the problem of local minima.

The basic idea is to transform the data $\mathbf{s}_i$ from the input space to a high dimensional feature space of feature vectors $\Phi(\mathbf{s}_i)$, where the inner products can be computed using a positive definite kernel function satisfying Mercer's condition [11] : $\kappa(\mathbf{s}_i, \mathbf{s}_j) = \langle \Phi(\mathbf{s}_i), \Phi(\mathbf{s}_j) \rangle$. In this paper, the inner product of spike trains is defined by the nonlinear cross intensity kernel that induces an infinite dimensional feature space.

For KLMS, the linear least-mean-square (LMS) algorithm is directly applied on RKHS with an emphasis on the general methodology to formulate linear filters and gradient descent algorithms in terms of inner products defined by the kernel $\kappa(\mathbf{s}_i, \mathbf{s}_j)$. As shown in Figure 1, $\Phi(\mathbf{s}_i)$ can be considered as an infinite dimensional vector. Let Ω be the weight vector in this space such that the output is $y(n) = \langle \Omega(n), \Phi(\mathbf{s}_i) \rangle$, where $\Omega(n)$ is Ω at time n. The LMS in the kernel feature space using stochastic instantaneous estimate of the gradient vector, yields

$$\Omega(0) = \mathbf{0}$$
$$\Omega(n+1) = \Omega(n) + \eta e(i)\Phi(\mathbf{s}_i)$$
$$\Omega(n) = \eta \sum_{i=0}^{n-1} e(i)\Phi(\mathbf{s}_i) \quad (7)$$

where $e(i) = d(i) - y(i)$. Given Ω and the input $\Phi(\mathbf{s}_i)$, the output is given by

$$\begin{aligned} y(n) &= < \Omega(n), \phi(\mathbf{s}_i) > \\ &= \eta \sum_{i=1}^{n-1} e(i) < \phi(\mathbf{s}_i), \phi(\mathbf{s}_n) > \\ &= \eta \sum_{i=1}^{n-1} e(i)\kappa(\mathbf{s}_i, \mathbf{s}_n) \end{aligned} \quad (8)$$

KLMS topology can be regarded as a growing Radial basis function (RBF) network, which allocates a new unit for the new sample $\mathbf{s}_i$ as the center and the scaled prediction error $\eta e(i)$ as the output weight coefficient. The coefficients and the centers are stored in memory during training. The algorithm is summarized below (Algorithm 1). KLMS on spike trains obtains adaptive nonlinear regression with the sample-by-sample adaption linear algorithm in RKHS without converging to local minima, which enable better tracking of neural system perturbations. Note further that this RKHS is very different from the RKHS defined by the Gaussian kernel used in machine learning, since the kernel here embodies the statistics of the spike trains captured in the nonlinear transformation of the intensity function (eq. 6).

Algorithm 1. The kernel least-mean-square algorithm

Initialization

choose step-size parameter η and kernel size σ,

$e(1) = d(1)$, $\mathcal{C}(1) = \mathbf{s}(1)$, $f_i = \eta e(1)\kappa(\mathbf{s}(1), \cdot)$

Computation

While $\{\mathbf{s}(i), \mathbf{d}(i)\}$ available **do**

% compute the output

$y_{i-1}(\mathbf{s}(i)) = \sum_{j=1}^{i-1} \eta e(i-1)\kappa(\mathbf{s}(i), \mathbf{s}(j))$

% compute the error

$e(i) = d(i) - y_{i-1}(\mathbf{s}(i))$

% store the new center

$\mathcal{C}(i-1) = \{\mathcal{C}(i-1), \mathbf{s}(i)\}$

end while

IV. Synthetic Experiment

A. Neural circuit

In the experiment, the plant is simulated by *a neural Circuit SIMulator* (CSIM). The network consists of 10 fully connected neurons, as shown in Figure 2. All neurons are modeled as Leaky Integrate-and-Fire (LIF) units [12] with the following parameters: membrane time constant 30ms, absolute refractory period 2ms (excitatory neurons), threshold 15mV (for a resting membrane potential assumed to be 0), reset voltage 14.3mV, constant nonspecific background current 13.5nA, input resistance 1 MΩ and input noise 9nA. The postsynaptic current is modeled as an exponential decay $\exp(-t/\tau_s)$ with $\tau_s = 3$ms. All neurons fire at a reasonable firing rate range (1Hz to 10Hz) in rest state. This neural circuit has one current input and two output neurons, whose spike trains are recorded as the observed neural responses.

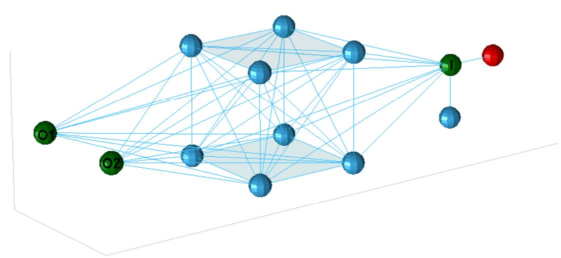


Fig. 2. The stimulated network of 10 neurons. The spike trains of neuron o1 and o2 are recorded as the observed neural responses.

B. Stimulation pattern

A periodic current stimulation at the input of the neural circuit generates the target spike train. The stimulation repeats every 2s with varying amplitude. In each 2s period, there are eight biphasic current pulses with different amplitudes separated by 250ms, as shown in Figure 3.

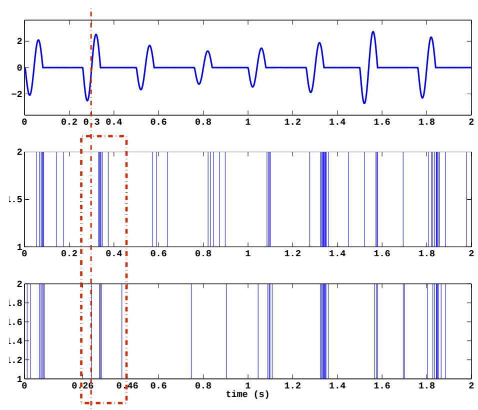


Fig. 3. **The upper plot shows the desired periodical stimulation within one periodic. Each stimulus is a biphasic pulse with different amplitude. The bottom plot shows the corresponding spike train produced at the output. The red block shows the windowed spike train, which is the input vector correspond to the stimulation signal label with the red line.**

C. Results

The decoder input is built from spike time vectors, which is obtained by sliding the window every 2ms (window size: [-20ms 80ms]) as shown in Figure 3. Here each window also includes the previous 20ms spike train which contains the information of the previous stimulation and helps to cancel the lasting influence of previous stimulus on the current neural firing pattern. The estimation of kernel size σ is based on the average pairwise distance $\|\lambda_{\mathbf{s}_i} - \lambda_{\mathbf{s}_j}\|$ of the training sample in the RKHS. The learning rate and τ are determined by the best results after scanning the parameters.

The main goal of decoding is to reconstruct the stimulation that can produce the neural response indistinguishable from the desired stimulation. Therefore, the performance evaluation is based on the following two criteria:

1 The normalized mean square error between the reconstructed stimulation and the desired stimulation.

$$E^2 = \frac{\frac{1}{N}\sum(d(n) - y(n))^2}{var(d(n))} \tag{9}$$

2 The log-likelihood of observing the entire target spike train elicited by the output stimulation with a Poisson process assumption [7], which evaluates the similarity between the target neural response and the elicited neural response of M output neurons.

$$L = \sum_{m=1}^{M}\left(\sum_{n} sp_n^m \log \lambda_n^m - \Delta \sum_{n} \lambda_n^m\right) \tag{10}$$

where λ_n^m is the intensity of the Poisson process estimated by the firing rates of neuron m produced by the output stimulation, and $\mathbf{sp}^m$ represents the target spike train of neuron m defined as a vector of spike counts binned at a brief time resolution $\Delta = 1ms$ and indexed by time n.

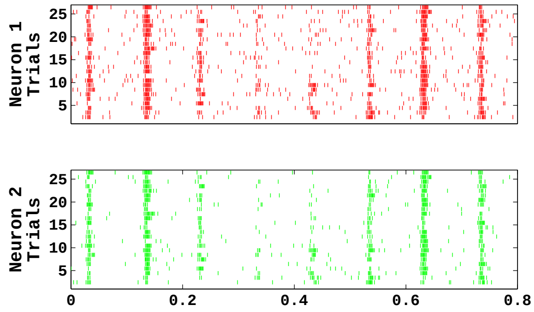
The simulation results demonstrate that the spike-based decoder is able to reconstruct the timing, the shape and the amplitude pattern of the biphasic stimulation with normalized mean square error 0.13. Figure 4(b) shows the one-trial output of decoder. In addition, the comparison between the input spike train raster plot (Figure 4(a)) and the elicited spike train raster plot (Figure 4(c)) also illustrates the accuracy of the decoded stimulation producing the spike train similar to that from the desired stimulation with log-likelihood 5.31×10^3.

TABLE I

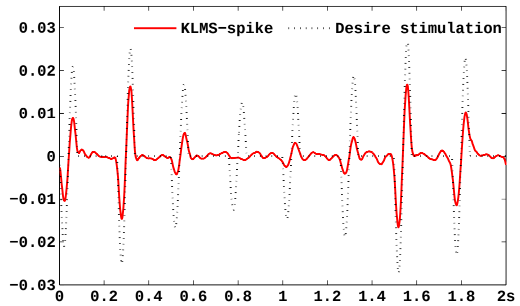
COMPARISON OF DECODING PERFORMANCE AMONG SPIKE-BASED KLMS, BINDATA-BASED KLMS AND BINDATA-BASED GLM

Method	Normalized-MSE	Log-likelihood
Spike-based KLMS	0.13	5.31×10^3
Bindata-based KLMS	0.27	4.78×10^3
Bindata-GLM	0.35	4.39×10^3

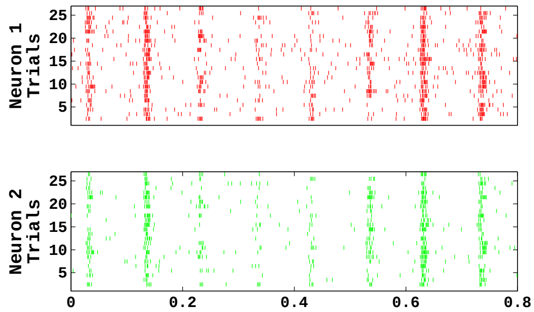
We also compare the performance of KLMS with RKHS for spike train with other two methods based on the discrete representation of spike trains (binning): GLM [1] with the link function $f = \exp(\cdot)$ and KLMS [6] with the Gaussian kernel for bin data. Although all three methods are able to detect the time of the stimuli, the spike KLMS wins the best decoding performance with respect to normalized mean square error of the output stimulation and log-likelihood of elicited spike trains, as shown in Table 1.



(a) The raster plot of the neural response recorded from two output neurons, which is produced by the desired stimulation.



(b) The reconstructed stimulation



(c) The raster plot of the neural response recorded from two output neurons, which is produced by the reconstructed stimulation.

Fig. 4. The testing results of the adaptive KLMS decoder in RKHS of the spike time vectors.

D. Perturbation Tracking

Considering the neural activity dynamics, the decoder is required to follow the time-varying transfer function. Therefore, we also investigate the control system ability to track the perturbation of the neural circuit by looking at the learning curves, as shown in Figure 5. In order to induce perturbations in the neural circuit, we change randomly the synaptic weight and the delay after 50 trials. In order to obtain a good performance of the decoding of a dynamic neural system, it is essential to select an appropriate learning rate that can balance the tradeoff between the tracking ability and the algorithm generality. Figure 5 illustrates the learning curves with three different learning rates. With the large learning rate (e.g. 0.5), the decoder converges fast at the beginning but with poor generality over the trails. However, the learning curve with a small learning rate (e.g. 0.01) has less vibrations which indicates the better generality over trials, but which causes a slow converge. In contrast, the decoder with learning rate 0.1 shows the best performance (the smallest mean square error) over all trails. The learning curve of the plant model decreases fast in the beginning demonstrating the ability of the decoder to learn progressively with the arrival of new data. The perturbation of the neuron circuit induces a fluctuation of the learning curve. However, the decoder is able to learn this variation by adaptation and converges to a new state within 5 trials.

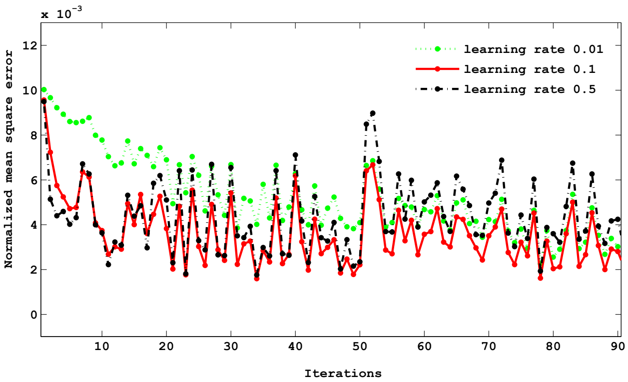


Fig. 5. **The learning curves of kernel-least-mean-square (KLMS) decoder with three different learning rates reproducing kernel Hilbert space (RKHS) of spike time vector. The vertical dot line shows when the neural circuit perturbation appears.**

V. Rat experiment

We apply the proposed method to reconstruct the electrical micro-stimulation imprinted in ventral posterolateral nucleus (VPL) of thalamus by decoding the elicited neural response recorded from somatosensory regions (S1).

A. Experimental data

The neural data used in this analysis was collected from a single chronically implanted rat with two 16-channel tungsten micro-wire arrays (Tucker Davis). Neuronal activity was recorded from arrays in S1 and in ventral posterolateral nucleus (VPL) of the thalamus using the Plexon multichannel acquisition processor. Action potentials were detected using a constant threshold and were sorted by a semi-automated clustering procedure (SortClient) using the first 3 principal components of the detected waveforms. Prior to the recording session, anesthesia was induced by isofluorane followed by a Nembutal injection and maintained with isofluorane.

Bipolar microstimulation (AM Systems Model 2200 Isolator) was applied to two adjacent electrodes in the thalamic array. The pulse current amplitude were varied among three distinct levels ($50\mu A$,$75\mu A$,$100\mu A$), with 140 responses from each level randomly permuted throughout the recording.

B. Results

We implemented a Multi-Input-Single-Output (MISO) KLMS decoder on rat data (VPL-S1). Spike trains are a more precise representation of the firing pattern of a neural circuit than local field potentials. However, spike trains have a large variability, therefore, it is essential to utilize multichannel spike trains to enhance the decoder robustness. However, if all channels are used as input, the computation cost increase drastically and the neuron channels that are not associated with stimuli introduces noise into the decoder. Therefore, based on the dependence analysis results [13], as shown in Figure 6, only four channels (10, 11, 12 and 15) that are the most highly dependent with the desired stimulation are selected as our input channels.

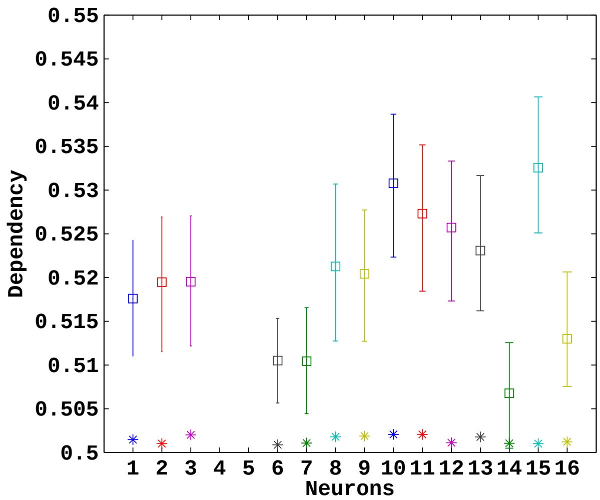


Fig. 6. **Dependency analysis based on the generalized association between the spike train and the stimulation over 16 neurons. The stars represent the dependency value of the surrogate test.**

The square wave stimulations are also preprocessed by passing through a *sinc* lowpass filter with Hamming window and a cutoff frequency of 500Hz. Regression is performed from the four spiking channels to smoothed stimulation waveform (low pass filtering the current magnitude waveform). The input is again the vector of spike times, which is obtained by sliding the window every 0.1ms (window size: 25ms).

The MISO decoder is able to reconstruct the shape and amplitude pattern of the desired stimulation with an amount of variance, as shown in Figure 7. Because of the variability of the recorded spike trains, the large variance exists in input space with respect to the same desired stimulation, which induces the output variance of decoder. In addition, the output variance is increased with decrease the stimulation current amplitude, since the lower signal-noise-ratio cause the larger variance of the input when the stimulation amplitude is low.

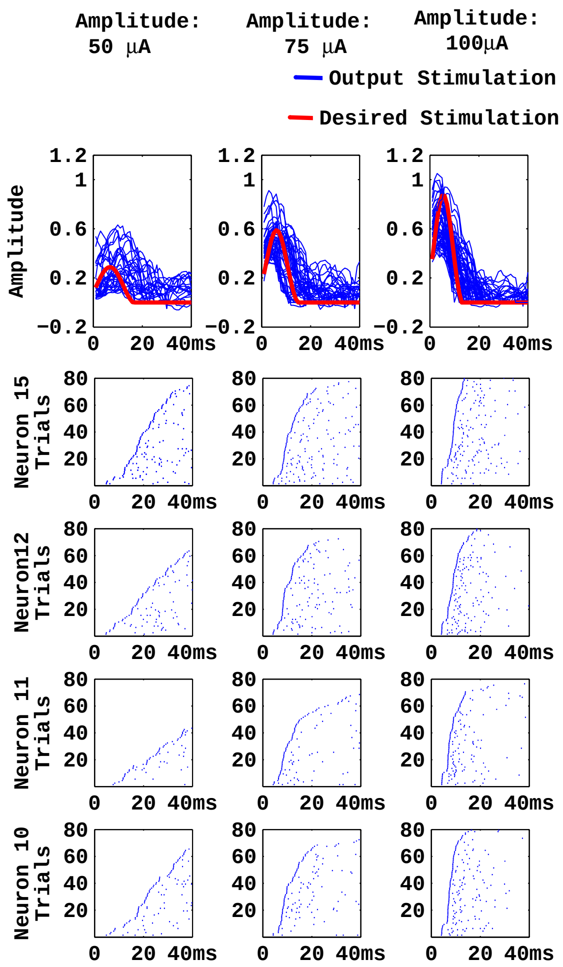


Fig. 7. **The upper plots show the test results of reconstructing the micro-stimulation in rat thalamus. The other plots represent input spike train rasters recoded from 4 channels in somatosensory cortical area (S1), which are sorted by the first spike time in order to show the spike train variability.**

VI. Conclusions

The paper proposes a Multiple-Input-Single-Output (MISO) adaptive decoder to reconstruct the continuous current stimulation from the observed neural response from multiple channels, which applies the kernel least mean square algorithm on the spike time space based on the definition of the inner product of the spike time vectors in reproducing kernel Hilbert space (RKHS). The decoder is applied directly on the input space of the spike time vector, instead of discrete representation of the spike time (bin data) that is widely used in neural signal processing, which bypass the limitations of bin data, such as time resolution and the sparseness. The comparison with bindata-based methods: KLMS and GLM indicate that this decoder is able to reconstruct the micro-stimulation with the best accuracy and the elicited neural response from the decoder output is the most similar with the target spike train. In addition, the adaptation is able to adjust the decoder to successfully track the perturbation in neural system.

References

[1] L. Paninski, J. Pillow, and J. Lewi, “Statistical models for neural encoding, decoding, and optimal stimulus design,” in *Computational Neuroscience: Progress in Brain Research*. Elsevier, 2006.

[2] E. N. Brown, L. M. Frank, D. Tang, M. C. Quirk, and M. A. Wilson, “A statistical paradigm for neural spike train decoding applied to position prediction from ensemble firing patterns of rat hippocampal place cells,” *The Journal of Neuroscience*, vol. 18, pp. 7411–7425, 1998.

[3] J. Eichhorn, A. Tolias, E. Zien, M. Kuss, C. E. Rasmussen, J. Weston, N. Logothetis, and B. Schlkopf, “Prediction on spike data using kernel algorithms,” in *In NIPS 16*. MIT Press, 2004, pp. 1367–1374.

[4] I. M. Park, “Capturing spike train similarity structure:a point process divergence approach,” Ph.D. dissertation, University of Florida, 2010.

[5] A. Arieli, A. Sterkin, A. Grinvald, and A. Aertsen, “Dynamics of ongoing activity: Explanation of the large variability in evoked cortical responses,” *Science*, vol. 273, pp. 1868–1871, 1996.

[6] W. Liu, J. C. Príncipe, and S. Haykin, *Kernel adaptive filtering*, S. Haykin, Ed. john wiley sons, Inc., 2010.

[7] P. Dayan and L. F. Abbott, *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*. Cambridge, MA, USA: MIT Press, 2001.

[8] D. L. Snyder and M. I. Miller, *Random Point Processes in Time and Space*. Springer-Verlag, 1991.

[9] R.E.Kass, V. Ventura, and E.N.Brown, “Statistical issues in the analysis of neuronal data,” *Journal of Neurophysiology*, vol. 94, pp. 8–25, 2005.

[10] A. R. C. Paiva, I. Park, and J. C. Prncipe, “A reproducing kernel hilbert space framework for spike train signal processing,” *Neural Computation*, vol. 21, pp. 424–449, 2009.

[11] N. Aronszajn, “Theory of reproducing kernels,” *Transactions of The American Mathematical Society*, vol. 68, pp. 337–337, 1950.

[12] W. Maass, T. Natschläger, and H. Markram, “Real-time computing without stable states: A new framework for neural computation based on perturbations,” *Neural Computation*, vol. 14, pp. 2531–2560, 2002.

[13] S. Seth, A. J. Brockmeier, J. S. Choi, M. Semework, J. T. Francis, and J. C. Principe, “Evaluating dependence in spike train metric spaces,” in *International Joint Conference on Neural Networks*, 2011.