34th Annual International Conference of the IEEE EMBS
San Diego, California USA, 28 August - 1 September, 2012

Decoding stimuli from multi-source neural responses

Lin Li, *Student Member, IEEE*, John S. Choi,
Joseph T. Francis, Justin C. Sanchez, *Member, IEEE,* and José C. Príncipe, *Fellow, IEEE*

***Abstract*— Spike trains and local field potentials (LFPs) are two different manifestations of neural activity recorded simultaneously from the same electrode array and contain complementary information of stimuli or behaviors. This paper proposes a tensor product kernel based decoder, which allows modeling the sample from different sources individually and mapping them onto the same reproducing kernel Hilbert space (RKHS) defined by the tensor product of the individual kernels for each source, where linear regression is conducted to identify the nonlinear mapping from the multi-type neural responses to the stimuli. The decoding results of the rat sensory stimulation experiment show that the tensor-product-kernel-based decoder outperforms the decoders with either single-type neural activities.**

I. INTRODUCTION

The rapid advance of micro-electrode arrays and electrophysiological recording techniques opens up new opportunities to precisely extract information of stimuli or behaviors from neural responses. Different types of neural activity are recorded simultaneously from the same electrode array, such as spike trains and local field potentials (LFPs), which encode complementary information of the stimuli or behaviors [1], [2]. In most recordings, the spike train is obtained by using a high-pass filter with the cutoff frequency about $300-500$ Hz, while the LFP is obtained by using a low-pass filter with the cutoff frequency about 300 Hz [3]. The spike train represents the single-unit neural activity with a fine temporal resolution. However, its stochastic properties induce considerable trial-to-trial variability, especially when the stimulation amplitude is small. In contrast, LFPs reflect the sum of all local currents near the surface of the electrode, which limits specification but provides robustness for characterizing the modulation induced by stimuli, even at low amplitudes. Therefore, an appropriate mergence of LFPs and spike trains in decoding models make it possible to more accurately decode the stimuli or behaviors from the neural response. For example, the decoder can coordinate LFP or spike patterns to tag particularly salient events or extract different stimulus features characterized by different type signals.

This work was supported in part by the U.S. NSF Partnerships for Innovation Program 0650161, and Darpa project N66001-10-C-2008.

Lin Li and J. C. Príncipe are with the Department of Electrical Engineering, University of Florida. (email: linli@cnel.ufl.edu; principe@cnel.ufl.edu)

John S. 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:jschoi831@gmail.com; joey199us@gmail.com)

J. C. Sanchez is with the Department of Biomedical Engineering, University of Miami. (email: jcsanchez@miami.edu)

The different signal properties between LFPs and spike trains make merging in the same model difficult. First, the underlying stochastic properties are totally different between the two representations. A spike train that is a set of spike timings can be interpreted as a realization of a point process [4], [5], while LFP is a continuous amplitude process. Moreover, the time scale of LFPs is significant longer than spike trains. In this paper, we propose a tensor-product-kernel-based regressor to decode the stimulus information from multi-type neural activities. The tensor product kernel allows modeling different types of signals individually and merge their information in the feature space defined by the tensor product of the individual kernels for each type signal [6].

Spike trains characterize stimulus information with a fine time resolution but a large variability. Most decoding techniques have been applied to discretized representations of spike trains [7], [8], [9], which has limited applicability for systems requiring a fine time resolution. With a small discretization size, the input space becomes sparse and the dimensionality of the input space also becomes a problem (curse of dimensionality). Instead, the set of spike time occurrences provides a more effective and accurate description of spike trains, but the space of spike trains is not a conventional L_2 functional space, i.e., operations such as addition and multiplication can not be applied. Therefore, this paper models spikes by using the Schoenberg kernel defined in the spike timing space [10], which decreases the computation time in contrast with the kernels based on the conventional rate representation and avoids sparse high-dimensional vectors corresponding to binned spike trains. In contrast, the LFPs, as a continuous process, are modeled by the Schoenberg kernel defined in the continuous space on the time structure (nonlinear correlation), which is able to enhance the robustness of stimulus estimation. The tensor product of the two kernels map data into a joint kernel feature space, where the kernel least mean square (KLMS) algorithm estimates the nonlinearly mapping from the neural response to the stimuli as necessary in somatosensory stimulation studies.

II. TENSOR PRODUCT KERNEL

We utilize distinctive kernels to map the spike train $\mathbf{s}_i$ and LFP $\mathbf{x}_i$ into feature functions $\varphi(\mathbf{s}_i)$ and $\psi(\mathbf{x}_i)$ in two RKHSs defined by $\kappa_x(\mathbf{x}_i,\mathbf{x}_j)$ and $\kappa_s(\mathbf{s}_i,\mathbf{s}_j)$, respectively. In order to merge the information from LFPs and spike trains, the tensor product of two kernels incorporate two individual

978-1-4577-1787-1/12/$26.00 ©2012 IEEE

RHKSs into a joint kernel defined feature space defined by

$$\kappa(\mathbf{x}_i,\mathbf{s}_i,\mathbf{x}_j,\mathbf{s}_j)=\kappa_x(\mathbf{x}_i,\mathbf{x}_j)\kappa_s(\mathbf{s}_i,\mathbf{s}_j) \quad (1)$$

Since subsequent processing and estimation are conducted in the kernel space, there are no constrains on the format of the input signals and their corresponding kernel functions. Therefore, the kernels for spike trains and LFPs can be selected individually based on their own stochastic properties. For spike trains that can be interpreted as observations of a point process, the Schoenberg kernel [11] defined in the spike timing space is applied. LFPs are a continuous amplitude stochastic process and so we also embed their time structure over the lags in a Schoenberg kernel defined in the continuous space. Since the stimulus response durations of LFPs are longer than spike trains, the time scales of their input sample are specified individually. The tensor product kernel allows decoding stimuli from multi-type signals with different timescales.

A. Schoenberg kernel for spike trains

A spike train can be represented as a sequence of ordered spike times i.e $\mathbf{s}=\{t_m\in\mathscr{T}_s : m=1,\ldots,M\}$, in the interval $\mathscr{T}_s=[0,T_s]$, which can also be written by $s(t)=\sum_{m=1}^{M}\delta(t-t_m),\{t_m\in\mathscr{T}_s : m=1,\ldots,M\}$. This paper utilizes the Schoenberg kernel [11], [12] on spike trains, which interprets a spike train as a realization of an underlying point process and defines an injective mapping based on a strictly positive definite kernel between the conditional intensity functions of two point processes defined by [12].

$$\begin{aligned}\kappa(\lambda(t|H_t^i),\lambda(t|H_t^j)) &= \exp(-\frac{\|\lambda(t|H_t^i)-\lambda(t|H_t^j)\|^2}{\sigma^2})\\ &= \exp(-\frac{\int_\tau(\lambda(t|H_t^i)-\lambda(t|H_t^j))^2dt}{\sigma^2}),\end{aligned} \quad (2)$$

where σ is the kernel size and H_t^i is the history of the process up to t. The full spike train segment is mapped into a function in the RKHS.

A practical choice used in this paper estimates the conditional intensity function using a kernel smoothing approach [12], [13], which allows estimating the intensity function from a single realization. The estimated intensity function is obtained by simply convolving $\mathrm{s}(t)$ with the smoothing kernel $\mathrm{g}(t)$, yielding

$$\hat{\lambda}_s(t)=\sum_{m=1}^{M}\mathrm{g}(t-t_m),\{t_m\in\mathscr{T}_s : m=1,\ldots,M\}, \quad (3)$$

where the smoothing function $\mathrm{g}(t)$ integrates to 1. The rectangular and exponential functions [14], [12] are popular smoothing kernels, which guarantee injective mappings from the spike train to the estimated intensity function. In order to decrease the kernel computation complexity, the rectangular function $\mathrm{g}(t)=\frac{1}{T_s}(\mathrm{U}(t)-\mathrm{U}(t-T_s))$ ($T_s\gg$ the interspike interval) is used here, where $\mathrm{U}(t)$ is a Heaviside function. With the kernel smoothing estimation of the intensity function, the kernel $\kappa(\lambda(t|H_t^i),\lambda(t|H_t^j))$ becomes

$$\kappa_s(s_i,s_j)=\exp(-\frac{\int_{\mathscr{T}_s}(\hat{\lambda}_{s_i}(t)-\hat{\lambda}_{s_j}(t))^2\mathrm{d}t}{\sigma_s^2}), \quad (4)$$

The Schoenberg kernel defined in the spike timing space is a strictly positive definite kernel, so it defines a RKHS that is nonlinearly related to the intensity functions thereby preserving information beyond crosscorrelation.

Individual LFP or spike train channels are a relatively poor decoders of stimulus or behavior on a single-trial basis. Therefore, we defined the kernel for multi-channel spike trains as follows [6]

$$\kappa_s(\mathbf{s}_i(t),\mathbf{s}_j(t))=\sum_{n=1}^{N}\kappa_s(s_i^n(t),s_j^n(t)) \quad (5)$$

Where N is the number of channels of spike trains.

B. Schoenberg kernel for LFP

Since LFP is a continuous signal, the Schoenberg kernel can also be defined in continuous space to map the correlation time structure of the LFP $x(t)$ into a function in RKHS in basically the same way,

$$\begin{aligned}\kappa_x(x_i(t),x_j(t)) &= \exp(-\frac{\|x_i(t)-x_j(t)\|^2}{\sigma_x^2})\\ &= \exp(-\frac{\int_{\mathscr{T}_x}(x_i(t)-x_j(t))^2\mathrm{d}t}{\sigma_x^2})\end{aligned} \quad (6)$$

where $\mathscr{T}_x=[0\ T_x]$. Similarly to spike trains, the kernel for multi-channel LFPs is defined by

$$\kappa_x(\mathbf{x}_i(t),\mathbf{x}_j(t))=\sum_{n=1}^{N}\kappa_x(x_i^n(t),x_j^n(t)) \quad (7)$$

Where N is the number of channels of LFPs. The time scale of the analysis for LFPs and spikes is very important and needs to be defined by the characteristic of each signal as will be explained below.

C. Kernel least mean square (KLMS)

In order to form regression models in the tensor product kernel space we utilize the kernel least mean square (KLMS) algorithm developed [15]. The great appeal of kernel-based filters is the usage of the linear structure of RKHS to implement well-established linear adaptive algorithms and to obtain a nonlinear filter in the input space that leads to universal approximation capability without the problem of local minima.

The basic idea behind KLMS is to transform the data $[\mathbf{s}_i,\mathbf{x}_i]$ given by $\mathbf{s}_i=\{t_m-(i-1)\tau,t_m\in[(i-1)\tau,(i-1)\tau+T_s] : m=1,\ldots,M\}$ (the ith window of the spike time sequence obtained by sliding T_s-length window with step τ) and $\mathbf{x}_i(t)=\{x(t),t\in[(i-1)\tau,(i-1)\tau+T_x\}$ (the ith window of the LFP obtained by sliding the T_x-length window with step τ) from the input space to a high dimensional feature space $\phi(\mathbf{s}_i,\mathbf{x}_i)$, where the inner products can be computed using a positive definite kernel function satisfying Mercer's condition $\kappa(\mathbf{s}_i,\mathbf{x}_i,\mathbf{s}_j,\mathbf{x}_j))=<\phi(\mathbf{s}_i,\mathbf{x}_i),\phi(\mathbf{s}_j,\mathbf{x}_j)>$ [15]. For

our work the RKHS of input signal is defined by the tensor product kernel. The linear least-mean-square (LMS) algorithm is directly applied in the RKHS defined by the product kernel $\kappa(\mathbf{s}_i,\mathbf{x}_i,\mathbf{s}_j,\mathbf{x}_j)$. Let Ω be the filter weight function (which can be considered an infinite dimensional vector) in RKHS and $[\mathbf{s}_i\mathbf{x}_i]$ be the input and $\phi(\mathbf{s}_i,\mathbf{x}_i)$ the transformed input function in the RKHS, then the regressor output in the input space is $y=\langle\Omega,\phi(\mathbf{s}_i,\mathbf{x}_i)\rangle$. During online adaptation Ω becomes $\Omega(n)$ at time n. Following the stochastic gradient LMS update, the KLMS in the kernel feature space using a stochastic instantaneous estimate of the gradient vector, yields

$$\Omega(n)=\eta\sum_{i=0}^{n-1}e_i\phi(\mathbf{s}_i,\mathbf{x}_i)\ \text{with}\ \Omega(0)=\mathbf{0} \tag{8}$$

where $e_i=d_i-y_i$ and d_i is the desired signal. Given Ω and the input $\phi(\mathbf{s}_n,\mathbf{x}_n)$, the output is given by

$$\begin{aligned} y_n &=<\Omega(n),\phi(\mathbf{s}_n,\mathbf{x}_n)> \\ &=\eta\sum_{i=1}^{n-1}e_i<\phi(\mathbf{s}_i,\mathbf{x}_i),\phi(\mathbf{s}_n,\mathbf{x}_n)> \\ &=\eta\sum_{i=1}^{n-1}e_i\kappa(\mathbf{s}_i,\mathbf{x}_i,\mathbf{s}_n,\mathbf{x}_n) \end{aligned} \tag{9}$$

The KLMS algorithm is intrinsically regularized by the step size, therefore this parameter should be carefully determined because it also affects the convergence rate [15].

III. Experiment

A. Rat data

Three female Long-Evans rats (Hilltop, Scottsdale, PA) were implanted with 32 channel Michigan Probes (NeuroNexus Inc.) in the hand region of primary somatosensory cortex (S1). Neural recordings were made using a multichannel acquisition system (Tucker Davis). The rat was placed into a small chamber with a mesh floor which was suspended above a table. The apparatus helped keep it calm and stationary even though they remained awake. Spike and field potential data was preamplified 1000x (filter cutoffs at 0.7 and 8.8kHz) and digitized at 25kHz. LFPs were further filtered from 1 to 300Hz using a 3rd order Butterworth filter. Spike sorting was achieved using k-means clustering of the first 2 principal components of the detected waveforms.

The experimental procedure involved delivering 30-40 short 100ms tactile touches to the rat's fingers (repeated for digit pads 1-4) using a hand-held probe. The rat remained still for the recording durations, and trials were canceled if the rat changed orientation. The applied force was measured using a lever attached to the probe that pressed against a thin-film resistive force sensor (Trossen Robotics) when the probe tip contacted the rat's body. The resistive changes were converted to voltage using a bridge circuit and were filtered and digitized in the same way as described above. The digitized waveforms were then de-meaned and filtered at 1 to 60Hz using a 3rd order Butterworth filter. The first derivative of this signal is used as the desired stimulation signal in this paper, which are shown in Figure 1.

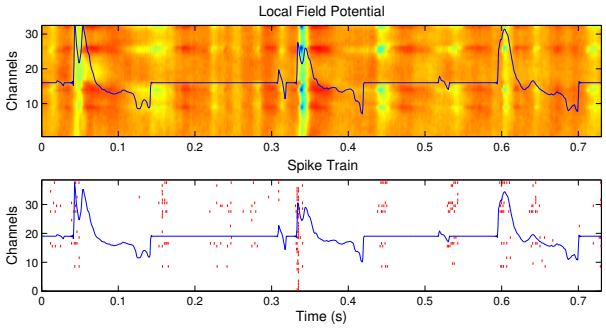

Fig. 1. Data set of the rat sensory stimulation experiment

B. Time scale estimation

The tensor product kernel (1) allows modeling the sample from different sources individually. Therefore, the time scales of LFPs and spike trains can be specified based on their own properties. In order to find reasonable time scales, we estimate the autocorrelation coefficients of LFPs and spike trains, which indicate the response duration induced by the stimulation. For this purpose, spike trains are binned with binsize 1ms. The local field potential are also resampled with sampling rate 1000Hz. The autocorrelation coefficients of the each signal average over channels are calculated by

$$\hat{\rho}_h=\frac{\sum_{t=h+1}^{T}(y_t-\bar{y})(y_{t-h}-\bar{y})}{\sum_{t=1}^{T}(y_t-\bar{y})^2}. \tag{10}$$

Its 90% confidence bounds of the hypothesis that the autocorrelation coefficient is effectively zero are approximately estimated by $\pm 2SE\rho$, where

$$SE\rho=\sqrt{(1+2\sum_{i=1}^{h-1}\rho_i^2)/N}. \tag{11}$$

The average confidence bounds for LFP and spike trains are $[-0.032\ 0.032]$ and $[-0.031\ 0.031]$, respectively. The autocorrelation coefficients of LFP fall into the confidence interval after 20ms, while the autocorrelation coefficients of spike trains die out after 9 ms, as shown in Figure 2. Therefore, the decoder inputs built from spike train and LFP are obtained by sliding the window with window size T_s =9ms for spike train and T_x =20 ms for LFP.

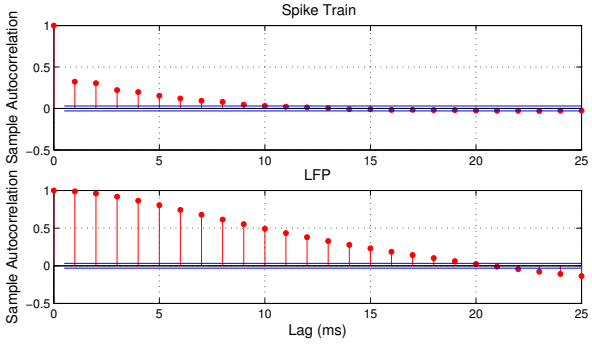

Fig. 2. Autocorrelation of LPFs and spike trains

C. Decoding results

We now present the results of decoding the tactile stimulus waveform using KLMS updates operating on the tensor

product kernel. For comparison, we also apply the kernel-based decoder on only one type of signals to find out what is the performance enhancement gained by using multi-type signals. Time discretization is 5 ms. The learning rates for each decoder are determined by the best results of test data after scanning the parameters. The kernel size σ_s and σ_x are determined by the average distance in RKHS of each pair of training samples. The normalized mean square error (NMSE) between the estimated stimulation ($\mathbf{y}$) and the desired stimulation ($\mathbf{d}$) is utilized as an accuracy criteria.

NMSEs are obtained across 8 trials data sets. For each trial, we use 20s data to train the decoders and compute an independent test error on the remaining 2.5s data. The results are shown in Table 1. the LFP&spike decoder out-performs both the LFP decoder and the spike decoder.

TABLE I

COMPARISON AMONG NEURAL DECODERS.

Property \ input	LFP&spike	LFP	spike
NMSE (mean/STD)	0.48/0.05	0.55/0.03	0.63/0.11

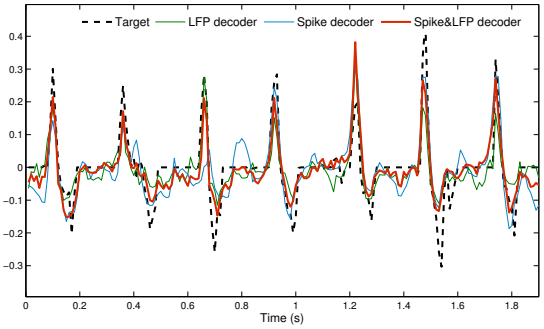

Fig. 3. Results of the LFP&spike decoder of the first stimulation trial.

In order to illustrate the details of the decoding performance, the test results of the first trial are illustrated in Figure 3. It is observed that the output of the spike decoder fluctuates enormously and misses some pulses, i.e., around 0.65 s, because of the sparsity and variability of the spike train. In contrast, the output estimated by LFP is smooth, because the LFP reflects the sum of all local currents on the surface of the electrodes, which causes the robustness but the limitation of the specification. The LFP&spike decoder preforms better than LFP decoder by gaining the precise pulse timing information from spike trains. The learning curves are also estimated by calculating the testing NMSE after the model parameter updated after each new sample entered. Three learning curves of spike decoder, LFP decoder and LFP&spike decoder are shown in Figure 4, respectively.

IV. Conclusion

Spike trains and LFPs encode complementary aspects of stimulus and behavior. However, the two signals have vastly different stochastic properties and are often modeled separately with different algorithms. Merging the two signals effectively is important for enhancing decoding performance. In this paper, we use a tensor product to merge two separate Schoenberg kernels: one defined in spike timing space

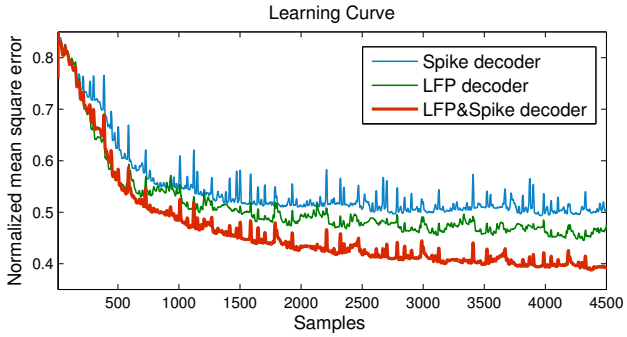

Fig. 4. Learning curves (NMSE of the test set) of the LFP&spike decoder of the first stimulation trial.

for spikes and one defined in continuous space for LFPs. The decoder also allows modeling LFPs and spikes with different time-scale parameters, which can be determined experimentally. We applied this algorithm to decoding tactile stimulations in the rat. In this setting, using both spikes and LFPs for decoding enhances performance over using each signal separately.

References

[1] R. Q. Quiroga and S. Panzeri, "Extracting information from neuronal populations: information theory and decoding approaches," *Nature Reviews Neuroscience*, vol. 10, pp. 173–185, 2009.

[2] J. R. Huxter, T. J. Senior, K. Allen, and J. Csicsvari, "Theta phasespecific codes for two-dimensional position, trajectory and heading in the hippocampus," *Nature Neuroscience*, vol. 11, pp. 587 – 594, 2008.

[3] M. J. Rasch, A. Gretton, Y. Murayama, W. Maass, and N. K. Logothetis, "Inferring spike trains from local field potentials," *Journal of Neurophysiology*, vol. 99, no. 3, pp. 1461–1476, 2008.

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

[5] R. E. Kass, V. Ventura, and E. N. Brown, "Statistical issues in the analysis of neuronal data," *Journal of Neurophysiology*, vol. 94, no. 1, pp. 8–25, 2005.

[6] B. Scholkopf and A. J. Smola, *Learning with kernels : support vector machines, regularization, optimization, and beyond*, 1st ed. The MIT Press, Dec. 2002.

[7] J. Pillow, Y. Ahmadian, and L. Paninski, "Model-based decoding , information estimation , and change-point detection in multineuron spike trains," *Statistics*, pp. 1–27, 2006.

[8] F. W. Volterra, R. Kelly, and T. S. Lee, "Decoding v1 neuronal activity using particle," in *Advances in Neural Information Processing Systems*, 2003, pp. 15–1359.

[9] L. Shpigelman, Y. Singer, R. Paz, and E. Vaadia, "Spikernels: Predicting arm movements by embedding population spike rate patterns in inner-product spaces," *Neural Computation*, vol. 17, no. 3, pp. 671–690, March 2005.

[10] L. Li, I. M. Park, S. Seth, J. S. Choi, J. T. Francis, J. C. Sanchez, and J. C. Principe, "An adaptive decoder from spike trains to micro-stimulation using kernel least-mean-squares (klms)," in *Machine Learning for Signal Processing (MLSP)*, sept. 2011, pp. 1–6.

[11] I. M. Park, S. Seth, M. Rao, and J. C. Principe, "Strictly positive definite spike train kernels for point process divergences," *Neural Computation*, accepted.

[12] 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.

[13] H. Ramlau Hansen, "Smoothing counting process intensities by means of kernel functions," *The Annals of Statistics*, vol. 11, no. 2, pp. pp. 453–466, 1983.

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

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