A Monte Carlo Sequential Estimation of Point Process Optimum Filtering for Brain Machine Interfaces

Yiwen Wang, *Student Member*, António R. C. Paiva, *Student Member,* José C. Príncipe, *Fellow,* Justin C. Sanchez, *Member*

***Abstract*—The previous decoding algorithms for Brain Machine Interfaces are normally utilized to estimate animal's movement from binned spike rates, which loses spike timing resolution and may exclude rich neural dynamics due to single spikes. Based on recently proposed Monte Carlo sequential estimation algorithm on point process, we present a decoding framework to reconstruct the kinematic states directly from the multi-channel spike trains. Starting with analysis on the differences between the simulation and real BMI data, neural tuning properties are modeled to encode the movement information of the experimental primate as the pre-knowledge for Monte-Carlo sequential estimation for BMI. The preliminary kinematics reconstruction shows better results when compared with Kalman filter.**

I. Introduction

BRAIN Machine Interface (BMI) is a framework in which the understanding of the spatial and temporal structure of neural activity is used to control a prosthetic device with the intention of movement. In the experiments [1][2], the microelectrode arrays were implanted into multiple cortical areas of a primate's brain to collect different signals of neural activity, such as local field potentials and single unit activities, while the primate was performing a 3-D food reaching task, or a 2-D target-tracking task.

On these recording results, several signal-processing approaches have been applied toward extracting the functional relationship between the neural recordings and the animal's kinematic trajectories. The inputs to these models are usually multi-channel neuronal binned spike rates collected from selected regions of a primate's brain. The outputs of these models are the predicted movements, and control a prosthetic robot arm to coordinate the intended movements.

One of the best well known methods is the population vector algorithm proposed by Georgopoulos et al. [3]. In this method the movement direction is predicted from all cells preferred direction vectors appropriately weighted according to each cell tuning property. An alternative decoding methodology uses binned spike trains to predict movement based on linear or nonlinear optimal filters. These methods avoids the need for explicit knowledge of the neurological dynamic encoding model of the neural receptive field, and standard linear or nonlinear regression is used to fit the relationship directly into the decoding operation. The Wiener filter or time delay neural network (TDNN) was designed to predict the 3D hand using neuronal binned spike rates embedded by a 10-tap delay line [1]. In addition to this forward model, a recursive multilayer perceptrons (RMLP) model was proposed by Sanchez et al. [4]. Subsequently, Kim et al. [5] proposed the development of switching multiple linear models combined with a nonlinear network to increase prediction performance in food reaching.

This work was supported by NSF grants ECS-0422718, CISE-0541241 and CNS-0540304 and the Children's Miracle Network. A. R. C. Paiva was supported by Fundação para a Ciência e a Tecnologia under grant SFRH/BD/18217/2004.

Yiwen Wang, António R. C. Paiva and José C. Príncipe are with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611, USA (e-mail: {wangyw, arpaiva, principe}@cnel.ufl.edu).

Justin C. Sanchez is with the Department of Pediatrics, Division of Neurology, University of Florida, Gainesville, FL 32611, USA (e-mail: jcs77@ufl.edu).

Yet another methodology can be derived probabilistically using a Bayesian formulation. From a sequence of noisy observations of the neural activity, the probabilistic approach analyzes and infers the response as a state variable of the neural dynamical system. The neural tuning property relates the measurement of the noisy neural activity to the stimuli, and builds up the observation measurement model. Consequently, a recursive algorithm based on all available statistical information can be used to construct the posterior probability density function of the biological response for each time, and in principle yields the solution to the decoding problem. Movements can be recovered probabilistically from the multi-channel neural recordings by estimating the expectation of the posterior density or by maximum likelihood estimation.

The Kalman filter is a special case of this framework and was previously applied to BMI [6]. Two strong assumptions of the Kalman filter are that time-series neural activities are generated from the stimulus through a linear system and that, given the neural spiking activities at every time step, the posterior density of the kinematic variable is Gaussian. These two assumptions may be too restrictive for BMI applications. The particle filter algorithm, generalizes the Kalman filter, and was also investigated to recover movement velocities from binned neural activity [7] [8].

The above algorithms are coarse approaches that lose spike timing resolution due to binning and may exclude rich neural dynamics due to single spikes. The primary reason for this limitation is that the methods are designed for continuous random valued observations, and cannot be applied directly to point processes. Indeed, a spike train point process is completely specified by the spike times.

A general point process adaptive filtering paradigm was recently proposed [9] to probabilistically reconstruct a freely running rat's position from the discrete observation of the neural firing. This algorithm modeled the neural spike train as an inhomogeneous Poisson process feeding a kinematic model through a nonlinear tuning function. This approach also embodies the conceptual Bayesian filtering algorithm: predicting the posterior density by a linear state update equation and revising it with the next observation measurement. Nevertheless, the method still assumes the posterior density of the state vector, given the discrete observation, is always Gaussian distributed.

Recently, we proposed a Monte Carlo sequential estimation algorithm on point process as a probabilistic approach to infer the kinematic information directly from the neural spike train [10]. The posterior density of the kinematic stimulus, given the neural spike train was estimated at each time step *non-parametrically*. The preliminary simulations showed a better velocity reconstruction from the exponentially tuned neural spike train without imposing a Gaussian assumption.

To derive the kinematic information from the neural activity for the BMI all the probabilistic approaches discussed require pre-knowledge of the neuron receptive properties (tunning functions). This is because the probabilistic approaches use the Bayesian formulation to construct the posterior density at each time step from the prior density of the kinematic state, from which we infer the kinematic value. For this reason such, these tunning functions need to be estimated from the data.

In this paper, we emphasize spike-based modeling of the functional relationship between neuron spike trains and kinematics. Our goal is to build an adaptive signal processing framework for Brain Machine Interfaces working directly in the spike domain, where the binning window size is not a concern. Multi-channel spike trains reserve the time resolution of neuron activities, where the algorithm could deal with the randomness of the neuron behaviors. Such an algorithm could be performed online to reconstruct the kinematics from observation of the neuron spike trains and also construct the base to adaptively model the nonstationary aspects of the neuron receptive fields. In this sense, the analysis of the neuron encoding properties and decoded movement results could contribute to the understanding of the physiologic reasoning of the neuron behaviors.

The remainder of this paper is organized as follows. In Part II, we review the Monte-Carlo sequential estimation for the point process optimum filtering algorithm and apply this algorithm toward solving the decoding of the Brain Machine Interfaces. Starting by exposing the differences between the simulation data and real BMI data, neural tuning properties are modeled to encode the movement information of the experimental primate as the pre-knowledge for Monte-Carlo sequential estimation for BMI. The final decoding framework for Brain Machine Interfaces is presented directly in the spike domain and is followed with preliminary kinematics reconstruction results in Part III.

II. Decoding in the Spike Domain for Brain Machine Interfaces

In this section, we review the Monte-Carlo sequential estimation algorithm for the point process optimum filtering, followed by the framework of BMI decoding in spike domain.

A. Monte Carlo Sequential Estimation for Point Processes

Given an observation interval $(0,T]$, the number $N(t)$ of events (e.g. spikes) can be modeled as a stochastic inhomogeneous Poisson process characterized by its conditional intensity function $\lambda(t \mid \mathbf{x}(t), \boldsymbol{\theta}(t), \mathbf{H}(t))$, i.e. the instantaneous rate of events, defined as

$$\lambda(t \mid \mathbf{x}(t), \boldsymbol{\theta}(t), \mathbf{H}(t)) = \lim_{\Delta t \to 0} \frac{\Pr(N(t+\Delta t) - N(t) = 1 \mid \mathbf{x}(t), \boldsymbol{\theta}(t), \mathbf{H}(t))}{\Delta t} \quad (1)$$

where $\mathbf{x}(t)$ is the system state, $\boldsymbol{\theta}(t)$ is the parameter of the adaptive filter, and $\mathbf{H}(t)$ is the history of all the states, parameters and the discrete observations up to time *t*. The relationship between the single parameter Poisson process λ, the state $\mathbf{x}(t)$, and the parameter $\boldsymbol{\theta}(t)$ is a nonlinear model represented by

$$\lambda(t \mid \mathbf{x}(t), \boldsymbol{\theta}(t)) = f(\mathbf{x}(t), \boldsymbol{\theta}(t)) \quad (2)$$

Using the nonlinear function $f(\cdot)$, assumed to be known or specified according to the application, such as tuning function in BMI decoding. Let us consider hereafter the parameter $\boldsymbol{\theta}(t)$ as part of the state vector $\mathbf{x}(t)$. Suppose at time instant *k* the previous system state is $\mathbf{x}_{k-1}$. Recall that because the parameter $\boldsymbol{\theta}$ was embedded in the state, all we need is the estimation of the state from the conditional intensity function (1), since the nonlinear relation $f(\cdot)$ is assumed known. Random state samples are generated using Monte Carlo simulations [11] in the neighborhood of the previous state according to

$$\mathbf{x}_k = F_k \mathbf{x}_{k-1} + \eta_k \quad (3)$$

Then, weighted Parzen windowing [12] was used with a Gaussian kernel to estimate the posterior density. The process is recursively repeated for each time instant propagating the estimate of the posterior density, and the state itself, based on the discrete events over time. Notice that due to the recursive approach the algorithm not only depend on the previous observation, but also depend on the whole path of the spike observation events.

Let $\{\mathbf{x}_{0:k}^i, w_k^i\}_{i=1}^{N_S}$ denote a Random Measure [13] in the posterior density $p(\mathbf{x}_{0:k} \mid N_{1:k})$, where $\{\mathbf{x}_{0:k}^i, i = 1, \cdots, N_S\}$ is the set of all state samples up to time k with associated normalized weights $\{w_k^i, i = 1, \cdots, N_S\}$, and N_S is the number of samples generated at each time index. Then, the posterior density at time *k* can be approximated by a weighted convolution of the samples with a Gaussian kernel as

$$p(\mathbf{x}_{0:k} \mid N_{1:k}) \approx \sum_{i=1}^{N_S} w_k^i \cdot k(\mathbf{x}_{0:k} - \mathbf{x}_{0:k}^i, \sigma) \quad (4)$$

where $N_{1:k}$ is the spike observation events up to time k modeled by an inhomogeneous Poisson Process, and $k(x-\bar{x},\sigma)$ is the Gaussian kernel in term of x with mean $\bar{x}$ and covariance σ. w_k^i is the weight derived by Bayes' rule and Markov Chain property

$$w_k^i \propto p(\Delta N_k \mid \mathbf{x}_k^i) \quad (5)$$

where $p(\Delta N_k \mid \mathbf{x}_k^i)$ is the probability of observing spikes in the interval $(t_{k-1}, t_k]$. The posterior density of the current state given the observation is modified by the latest probabilistic measurement of the observing spike event $p(\Delta N_k \mid \mathbf{x}_k^i)$, which is the updating stage in adaptive filtering.

Without a close form of the state estimation, we measure the posterior density of the state given the observed spike event $p(\mathbf{x}_k \mid N_{1:k})$ every time and apply two methods to get the state estimation $\widetilde{\mathbf{x}}_k$ by Maximum Likelihood Estimation (MLE), or the expectation of the posterior density as (6) and (7):

$$\widetilde{\mathbf{x}}_k = \sum_{i=1}^{N_S} p(\Delta N_k \mid \mathbf{x}_k^i) \cdot \mathbf{x}_k^i \quad (6)$$

$$V_k = \sum_{i=1}^{N_S} p(\Delta N_k \mid \mathbf{x}_k^i) \cdot (\sigma + (\mathbf{x}_k^i - \widetilde{\mathbf{x}}_k)(\mathbf{x}_k^i - \widetilde{\mathbf{x}}_k)^T) \quad (7)$$

B. Data Collection and Analysis of Difference between Simulation and Real Data

In the one-neuron spike train decoding simulation, the Monte Carlo sequential estimation algorithm provided a better estimate of the state recursively without Gaussian distribution [10]. The Monte Carlo sequential estimation in spike domain is a promising signal processing tool to decode the continuous kinematics variable directly from neural spike trains in Brain Machine Interfaces. With this method, spike binning window size is no longer a concern, as one can directly utilize the spike timing event. The online state estimation is suitable for real-time BMIs decoding without the desired signal; however, both the neural activity recoding and desired trajectories are required to estimate the neuron tuning function. The decoding results by Monte Carlo estimation could be different between realizations because of the random manner in which samples are generated to construct the posterior density.

The Brain Machine Interfaces paradigm was designed and implemented by the Nicolelis lab at Duke University. The electrical neural activities were recorded invasively in the brain of an adult female Rhesus monkey named Aurora, and synchronized with her task behaviors. Several micro-electrode arrays were chronically implanted in five of the monkey's cortical neural structures: right dorsolateral premotor area (PMd), right primary motor cortex (MI), right primary somatosensory cortex (S1), right supplementary motor area (SMA), and the left primary motor cortex (MI). Each electrode array consisted of up to 128 microwires (30 to 50 μm in diameter, spaced 300 μm apart), distributed in a 16×8 matrix. Each recording site occupied a total area of 15.7 mm^2 (5.6×2.8 mm) and was capable of recording up to four single cells from each microwire for a total of 512 neurons (4×128) [14].

After the surgical procedure, a multi-channel acquisition processor (MAP, Plexon, Dallas, TX) cluster was used in the experiments to record the neuronal action potentials simultaneously. Analog waveforms of the action potential were amplified and band pass filtered from 500 Hz to 5KHz.The spikes of a single neuron from each microwire were discriminated based on time-amplitude discriminators and a principal component (PC) algorithm [1][15] . The firing times of each spike were stored.

The monkey performed a two-dimensional target-reaching task to move the cursor on a computer screen by controlling a hand-held manipulandum in order to reach the target. The monkey was rewarded when the cursor intersected the target. The corresponding position of the manipulandum was recorded continuously for an initial 30-min period at a 50Hz sampling rate, referred to as the "pole control" period [16] .

BMI data provides us with 185 neural spike train channels and 2-dimensional movement trajectories for about 30 minutes. Compared to the one-neuron decoding simulation in [10], there are big differences.

At first glance, it is remarkable that the time resolution for the neural spike train is about a millisecond, while the movement trajectories have a sampling frequency 50Hz.The neural spike trains allow us to more closely observe the true random neural behavior. Consequently, however the millisecond scale requires more computational complexity. We must bridge the disparity between the microscopic neural spikes and the macroscopic kinematics.

The tuning function provides a basis on which to build a simultaneously functional relationship. In the simulation [10], tuning function is simply assumed as exponentially increasing firing rate conditioned on the velocity. For the real BMI data, is this tuning function still valid and cogent? The Monte Carlo sequential estimation algorithm works as probabilistic approach directly in the spike domain with major assumption that we have enough knowledge of both the system model and the observation model. This assumption establishes a reliable base to propagate the posterior density leading to the state estimation at each time iteration. The knowledge should be gained as insight into neural tuning properties by analyzing the existing neuron and kinematics data, which leads to better kinematics decoding from neural activities in the future.

Anther issue to resolve is dealing with multi-channel neural spike trains when there is only one neural channel in the simulation. In the real BMI data, how can we account for the association between channels? Most of the work focused on the exclusive relationship between neural activities, such as the correlation between neurons characterized by the neural firing, or between neuron microscopic spiking and field potentials. With regard to both external kinematics and neural activities, neural spike trains between channels are usually assumed to be conditionally independent of kinematics. In other words, spike generation is determined once the kinematics and parameters of the neuron tuning are known. We should emphasize that the assumption of conditional independence does not conflict with the association analysis between neurons. If the firing rates of two neurons are generated independently through two similar tuning functions in a certain time period, similar firing patterns are expected during this time period, and the analysis on the correlation between them is still valid.

C. Neuron Tuning Analysis

The literature contains many different examples of tuning functions relating movement to neural activities. Most were linear weight combinations of the projection on 2 or 3 dimensions of kinematic vectors and bias, including the direction angle information. Moran and Schwartz [17] introduced an exponential velocity and direction tuned motor cortical model. Emery Brown used a Gaussian tuning function for the hippocampal pyramidal neurons [9]. These nonlinear mathematical models are not optimal for dealing with the real data because the tuned cells can have very different tuning properties and probably change over time. The accuracy of the tuning function estimation can directly affect the pre-knowledge of the Bayesian approach and, therefore, the results of the kinematic estimation. Marmarelis and Naka [18] developed a statistical method, called white noise analysis, to model the neural responses with stochastic stimuli. This method was improved by Simoncelli, Paninski and colleagues [19] . By parametric model identification, the nonlinear property between the neural spikes and the stimuli was directly estimated from data, which is more reliable than just assuming a linear or Gaussian format. In our BMI, we want to use sequential state estimation on a point process algorithm to infer the kinematic vectors from the neural spike train, which is the opposite of sensory neurons. However, we can regard the proper kinematic vector as the outcome of the motor cortex neurons. The tuning function between the kinematic vector and the neural spike train is exactly the observation model between the state and observation in our algorithm.

The tuning function is modeled as a linear filter followed by a static nonlinearity followed by a Poisson model, as shown in Figure 1 [19].

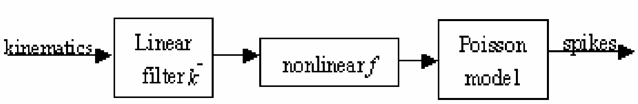

Fig. 1 Block diagram of linear-nonlinear-Poisson model

At each time step, the multi-dimensional kinematic vector is converted by linearly combining it with a weight vector k to a scalar output. The linear filter also represents the preferred kinematic direction. This linear filter response is then passed through a nonlinear function f, the output of which is determined by the instantaneous firing rate of a Poisson spike generator. The tuning function can be shown as (8) and (9).

$$\lambda_t = f(\vec{k} \cdot \vec{\mathbf{x}}_{t+lag}) \qquad (8)$$

$$spike_t = Poisson(\lambda_t) \qquad (9)$$

where $\vec{\mathbf{x}}_t$ is the instantaneous kinematics vector, which contains all the relevant information of position, velocity and acceleration $[\vec{\mathbf{p}}_x\ \vec{\mathbf{v}}_x\ \vec{\mathbf{a}}_x\ \vec{\mathbf{p}}_y\ \vec{\mathbf{v}}_y\ \vec{\mathbf{a}}_y]_t^T$, with the causal time delay estimated for the motor cortical neurons [20] .

D. Monte Carlo Sequential Estimation Framework for BMI Decoding

We have thus far presented background on the difference between simulation and BMI real data, and have elaborated on the Monte Carlo sequential estimation algorithm. Based on this information, we now present a systematic framework for BMI decoding using a probabilistic approach.

The decoding of Brain Machine Interfaces is intended to infer the primate's movement from the multi-channel neuron spike trains. The spike time is our observation signal. The kinematics is the state that needs to be derived from the point process observation through the tuning function by our Monte Carlo sequential estimation algorithm. The following steps represent the entire process:

***Step 1*:** Preprocess and analysis.

1) Generate spike trains from stored spike times.
2) Synchronize all the kinetics with the spike trains.
3) Evaluate the information theoretic tuning depth for all neurons. Label the important tuned neurons as the candidates for the subset.
4) Assign the kinematic vector $\vec{\mathbf{x}}$ to reconstruct.

***Step 2*:** Model estimation.

1) Estimate the kinematics dynamic system model
$$F_k = (E[\vec{\mathbf{x}}_{k-1}\vec{\mathbf{x}}_{k-1}^T])^{-1} E[\vec{\mathbf{x}}_{k-1}\vec{\mathbf{x}}_k^T]$$
2) For each neuron j, estimate the tuning function
   - Linear model $\vec{k}^{\,j} = (E[\vec{\mathbf{x}}^T\vec{\mathbf{x}}])^{-1} E_{\vec{\mathbf{x}}|spike^j}[\vec{\mathbf{x}}]$
   - Nonlinear function $f^j(\vec{k}^{\,j} \cdot \vec{\mathbf{x}}_t) = p(spike^j, \vec{k}^{\,j} \cdot \vec{\mathbf{x}}) \big/ p(\vec{k}^{\,j} \cdot \vec{\mathbf{x}})$
   - Build the inhomogeneous Poisson generator

***Step 3*:** Monte Carlo sequential kinematics estimation

For each time k, a set of samples for state $\vec{\mathbf{x}}_k^i$ are generated, i=1:N

1) Predict new state samples $\vec{\mathbf{x}}_k^i = F_k \vec{\mathbf{x}}_k^i + \eta_k$, i=1:N
2) For each neuron j,
   - Estimate the conditional firing rate $\lambda_k^{i,j} = f^j(\vec{k}^{\,j} \cdot \vec{\mathbf{x}}_k^i)$, i=1:N
   - Update the weights $w_k^{i,j} \propto p(\Delta N_k^j \mid \lambda_t^{i,j})$, i=1:N
3) Draw the weight for the joint posterior density
$$W_k^i = \prod_j w_k^{i,j}, \quad i=1:N$$
4) Normalize the weights $W_k^i = W_k^i \big/ \sum_i W_k^i$, i=1:N
5) Draw the joint posterior density
$$p(\vec{\mathbf{x}}_k \mid N_{1:k}) \approx \sum_{i=1}^{N} W_k^i \cdot k(\vec{\mathbf{x}}_k - \vec{\mathbf{x}}_k^i)$$

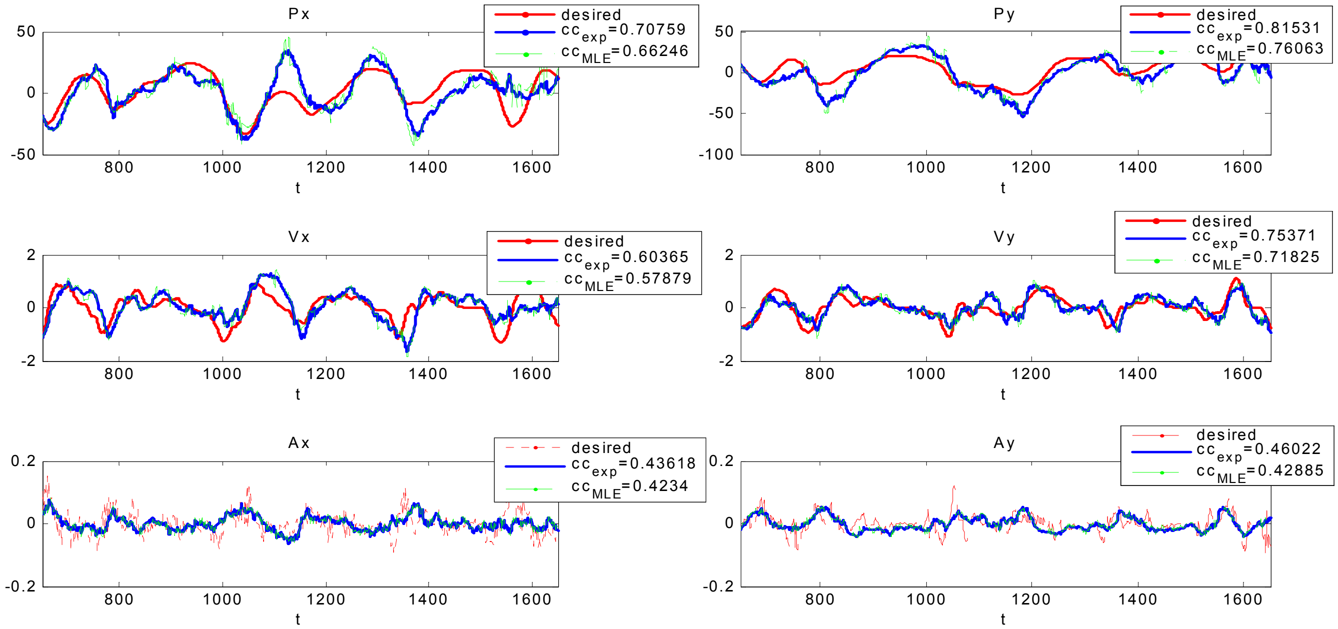

Fig 2. The reconstructed kinematics for 2-D reaching task

TABLE 1 CORRELATION COEFFICIENT EVALUATED BY THE SLIDING WINDOW

CC		Position		Velocity		Acceleration	
		x	y	x	y	x	y
Training	EXP.	0.8401 ± 0.0738	0.8945 ± 0.0477	0.7944 ± 0.0578	0.8142 ± 0.0658	0.5256 ± 0.0658	0.4460 ± 0.1495
	MLE	0.7984 ± 0.0963	0.8721 ± 0.0675	0.7805 ± 0.0491	0.7918 ± 0.0710	0.4950 ± 0.0430	0.4471 ± 0.1399
Testing	EXP.	0.7555 ± 0.1085	0.8596 ± 0.1267	0.6980 ± 0.0764	0.7395 ± 0.0341	0.4312 ± 0.1619	0.4100 ± 0.0850
	MLE	0.7257 ± 0.1051	0.8345 ± 0.1422	0.6670 ± 0.0762	0.7053 ± 0.0343	0.4186 ± 0.1568	0.3785 ± 0.0903

6) Estimate the state $\mathbf{x}_k^*$ from the joint posterior density by MLE or expectation.

7) Resample $\vec{\mathbf{x}}_k^i$ according to the weights W_k^i.

III. RESULTS OF MONTE CARLO SEQUENTIAL ESTIMATION ON BMI DECODING IN SPIKE DOMAIN

We first preprocessed the 185 channels of the neuron spiking time as a 0, 1 point process. A small time interval was chosen as 10 ms. The interval is small enough so that 99.62% of intervals had a spikes count less than 2. For each neuron, 1 was assigned when 1 or more than 1 spikes appeared during this interval, otherwise 0 was assigned. 185 multi-channel spike trains were generated 1750 seconds long. The recorded 2-D position vector $\vec{\mathbf{p}}$ is interpolated to be synchronized with the spike trains. The velocity $\vec{\mathbf{v}}$ is derived as the difference between the current and previous positions, and the acceleration $\vec{\mathbf{a}}$ is derived the same way from the velocity.

10000 samples of kinematics vectors were generated as training set for the parameter estimations. The neural activities were aligned with the kinematics according to their best causal time delay lag, which is estimated by information theoretical analysis on the mutual information between the spike and the delayed linear filter kinematics vector [20]. The kinematics dynamic system model F_k was estimated by the least square solution as stated in the framework. The noise distribution $p(\eta)$ is approximated by the histogram of $\eta_k = \mathbf{x}_k - F_k \mathbf{x}_{k-1}$. For each neuron j, the linear parameter $\vec{k}^j$ and nonlinear function f^j in the tuning function $\lambda_k^j = f^j(\vec{k}^j \cdot \vec{\mathbf{x}}_k)$ are estimated and stored using the method shown in part II.D. The kernel size in estimation nonlinear function f^j was assigned 0.02.

Monte Carlo sequential estimation for point process was then implemented on the 185 aligned neural spike trains to reconstruct the kinematics vector for 1000 samples long, referred as test data set. Through the dynamic system model at each time index, the noise was randomly generated according to $p(\eta)$. With the conditionally independent assumption, the joint posterior density of 185 neurons was calculated as the product of the posterior density for each neuron by weighted Parzon window. The posterior density was smoothed by convolving with a Gaussian kernel, where the kernel size was designed according to the Silverman's rule [21]. The kinematics vector was estimated and compared by both Maximum Likelihood Estimation and the expectation by collapse.

Figure 2 shows the reconstructed kinematics from all 185 neuron spike trains for test data. The left and right column plots display the reconstructed kinematics for x-axis and y-axis. The 3 rows of plots illustrate from top to bottom the reconstructed position, the velocity and the acceleration. In each subplot, the red line indicates the desired signal, the blue line indicates the expectation estimation, and the green line indicates the MLE estimation. The correlation coefficients between the desired signal and the estimations were shown at upper right corners of each plot.

Another probabilistic approach, the Kalman filter algorithm, was applied to the same data [14] to predict the animal's positions from the binned neural spiking rate. Compared to our approach here, the Kalman filter simply assumes that both the kinematic dynamic system model and the tuning function are linear, and that the posterior density is Gaussian distributed. The average correlation coefficient for the reconstructed position x was 0.62 ± 0.26 and for y was 0.82 ± 0.11 [14] with a sliding window of 4 seconds for 40 sample predictions. Because we have a different sampling frequency, 100Hz for kinematics, rather than 10Hz, our average correlation coefficients are calculated with an overlapping window, also 4 seconds long, for 400 sample predictions. The average correlation for position x is 0.7555 ± 0.1085 and for y is 0.8596 ± 0.1267, which is better than the Kalman filter results. The correlation evaluations by the sliding window for both training and testing reconstructed kinematics are shown in Table 1.

Our approach resulted in a reasonable reconstruction of the position and the velocity. The reconstructed kinematics estimated from the expectation of the joint posterior density performed better than the one from the noise maximum likelihood estimation. The position shows that the best correlation coefficients. This result may be due to the fact that the velocity and the acceleration were derived as differential variables, where the noise in the estimation might be magnified. Another interesting phenomenon is that the y-kinematics is consistently reconstructed better than x, which agrees with previous approaches.

IV. Conclusion

The Monte Carlo sequential estimation framework for BMI is a probabilistic approach to reconstruct the kinematics directly from the neural spike trains. It requires knowledge of physiology and of novel signal processing techniques. The neural activities are observed directly in the spike domain, which reserves the time resolution of the random neuron behaviors. The tuning functional relationship between the kinematics and neuron spikes is properly designed by a parametric linear-nonlinear-Poisson model. Based on the knowledge gained from the neuron physiology function analysis, the Monte-Carlo sequential estimation for the point process adaptive filtering was implemented on real-time BMI to infer the kinematics from a multi-channel spike train. The algorithm estimates the posterior density more accurately without any assumptions. Compared to the same task data with another probabilistic approach, Kalman filter, Monte Carlo estimation proved to be better capable of probabilistically inferring the kinematic states.

Acknowledgments

The authors would like to thank Dr. Miguel Nicolelis for the use of the monkey data collected in his laboratory at Duke University.

References

[1] Wessberg J, Stambaugh CR, Kralik JD, Beck PD, Laubach M, Chapin JK, Kim J, Biggs SJ, Srinivasan MA, Nicolelis MA (2000). Real-time prediction of hand trajectory by ensembles of cortical neurons in primates, *Nature,* 408:361-365

[2] Serruya M. D., Hatsopoulos N. G, Paninski L., Fellows M. R., and Donoghue J. P. (2002), Brain-machine interface: Instant neural control of a movement signal, *Nature*, vol. 416, pp. 141-142

[3] Georgopoulos A. P., Schwartz A. B., and Kettner R. E. (1986), Neuronal population coding of movement direction, *Science*, vol. 233, pp. 1416-1419

[4] Sanchez, J. C., Kim, S. P., Erdogmus, D., Rao, Y. N., Principe, J. C., Wessberg, J., & Nicolelis, M. A. (2002) Input-output mapping performance of linear and nonlinear models for estimating hand trajectories from cortical neuronal firing patterns. *Proc. Of Neural Net. Sig. Proc.*, pp. 139-148

[5] Kim, S. P., Sanchez, J. C., Erdogmus, D., Rao, Y. N., Wessberg, J., Principe, J. C., & Nicolelis M. A. (2003) Divide-and-conquer approach for brain machine interfaces: nonlinear mixture of competitive linear models. *Neural Networks*, 16, pp. 865-871

[6] Wu W., Gao Y., Bienenstock E., Donoghue J. P., Black M. J. (2006), Bayesian population decoding of motor cortical activity using a Kalman filter, *Neural Comput.*18: 80-118

[7] Brockwell A. E., Rojas A. L. and Kass R. E. (2004), Recursive Bayesian decoding of motor cortical signals by particle filtering, *J Neurophy*. 91: 1899-1907

[8] Schwartz A. B. (1992). Motor cortical activity during drawing movements: Single-unit activity during sinusoid tracing. *J Neurophysiol* 68: 528–541

[9] Brown E. N., Nguyen D. P., Frank L. M., Wilson M. A., and Solo V. (2001), An analysis of neural receptive field plasticity by point process adaptive filtering, *PNAS*, vol. 98, pp. 12261–12266

[10] Wang Y. W. , Paiva A. R. C., Principe J. C. (2006), A Monte Carlo Sequential Estimation for Point Process Optimum Filtering, *IJCNN 2006,* accepted

[11] J Carpenter, P Clifford & P Fearnhead, "Improved Particle Filter for Non-linear Problems", *IEE Proceedings on Radar and Sonar Navigation*, 1999

[12] E. Parzen, "On the estimation of a probability density function and the mode", Annals of Mathematical Statistics, 33(2), 1065-1076, 1962

[13] Arulampalam, M.S. Maskell, S. Gordon N. Clapp, T., "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking", *IEEE Transactions on Signal Processing,* 50, 2:174-188, Feb 2002

[14] Sanchez J. C. (2004), From cortical neural spike trains to behavior: modeling and analysis, *PhD. Dissertation*, University of Florida

[15] Nicolelis, M.A.L., Ghazanfar, A.A., Faggin, B., Votaw, S., & Oliveira, L.M.O. (1997) Reconstructing the engram: simultaneous, multiple site, many single neuron recordings. *Neuron* 18, pp. 529-537

[16] Carmena J. M., Lebedev M. A., Crist R. E., O'Doherty J. E., Santucci D. M., Dimitrov D. F., Patil P. G, Henriquez C. S., Nicolelis M. A. L. (2003), learning to control a brain machine interface for reaching and grasping by primates, *PLoS Biology*, vol. 1, issue 2, 193-208

[17] Moran D. W., Schwartz A. B. (1999), Motor cortical representation of speed and direction during reaching, *J. Neurophysiol.,* 82: 2676-2692

[18] Marmarelis, P. Z. & Naka, K. (1972). White-noise analysis of a neuron chain: An application of the Wiener theory. *Science*, 175, 1276-1278

[19] Simoncelli E.P., Paninski L., Pillow J., Schwartz O. (2004), Characterization of neural responses with stochastic stimuli, *The New Cognitive Neurosci.,* 3rd edition, MIT Press

[20] Wang Y. W., Sanchez J., Principe J., (2007), Information theoretical tuning depth and time delay estimators of motor cortex neuron, *EMBS Conference. on Neural Engineering*, submitted.

[21] Silverman B. W. (1981), Using Kernel Density Estimates to Investigate Multimodality*, J. Roy. Stat. Soc.*, Ser. B., Volume 43, p. 97-99