

Decoding the Non-Stationary Neuron Spike Trains by Dual Monte Carlo Point Process Estimation in Motor Brain Machine Interfaces

Yuxi Liao, Hongbao Li, Qiaosheng Zhang, Gong Fan, Yiwen Wang, and Xiaoxiang Zheng

Abstract— Decoding algorithm in motor Brain Machine Interfaces translates the neural signals to movement parameters. They usually assume the connection between the neural firings and movements to be stationary, which is not true according to the recent studies that observe the time-varying neuron tuning property. This property results from the neural plasticity and motor learning *etc.*, which leads to the degeneration of the decoding performance when the model is fixed. To track the non-stationary neuron tuning during decoding, we propose a dual model approach based on Monte Carlo point process filtering method that enables the estimation also on the dynamic tuning parameters. When applied on both simulated neural signal and in vivo BMI data, the proposed adaptive method performs better than the one with static tuning parameters, which raises a promising way to design a long-term-performing model for Brain Machine Interfaces decoder.

I. INTRODUCTION

Brain machine interfaces (BMIs) build an alternative pathway for the communication between the brain and the outside instruments. Neural activity collected from motor cortex can be translated to movement position or velocity, helping animals or paralyzed patients to finish tasks like lever pressing, 2-D or 3-D target tracking or self-feeding [1-4].

Several signal processing algorithms have been carried out to seek the neural representation of animals' kinematic parameters in motor cortex. One study performs the standard linear or nonlinear regression using the binned spike trains [5, 6]. They avoid the specific tuning model, which explains how the information been encoded in the neural activities. Yet another study takes the probabilistically Bayesian recursive formulation, which involves the neural tuning property [7-10]. They usually assume that the tuning function is stationary. Once the decoding model is obtained from the training set, the parameters will be fixed and applied to the whole decoding process.

However, more and more evidences have shown that the connection between the neural firing and movements keeps

Resrach supported by grants from the National Natural Science Foundation of China (No.61031002), 973 Program (2013CB329506), the National High Technology Research and Development Program of China (No.2012AA011602), the National Natural Science Foundation of China (No. 61305146), and the Zhejiang Provincial Natural Science Foundational science and technology cooperation program (LY14F030015). All the authors are with Qiushi Academy for Advanced Studies, Zhejiang University.

Yuxi Liao, Hongbao Li, Qiaosheng Zhang, Gong Fan and Xiaoxiang Zheng are with the Qiushi Academy for Advanced Studies and Department of Biomedical Engineering, Zhejiang University, Hangzhou 310027, China.

Yiwen Wang are with the Qiushi Academy for Advanced Studies, Zhejiang University, Hangzhou 310027, China (e-mail: eewangyw@zju.edu.cn)

changing over time [5, 11]. For example, a micro-electrode may pick up a different neuron in neighbor, due to the minor shift happens between the electrode and the cortex during breathe or individual neuron dies. Also, neurons are proved to be plastic even in practicing a simple task [12]. Bizzi found many cells changed their tuning properties after an extra force field added to the experiment, and this change can be memorable or memory-less [13]. As a result, decoding with a static model will receive a worse performance as time progresses into testing [10].

Many works have addressed on this problem. One option is to update the parameters of an adaptive model when new observations are available [14-18]. Alternative option is to monitor the change of tuning properties and adjust the corresponding parameters [19-20]. It helps us to understand more about dynamic neural activity. One essential problem here is that how to estimate the tuning curve and how to describe their time-variant characters. Since Georgopoulos first rises up the cosine curve with preferred direction [21], many different forms of tuning model have been proposed, from linear [2], Gaussian [8], Exponential [9] to an assumption-less linear-nonlinear-Poisson (LNP) model [22]. Unfortunately, an assumption-less model is too complicate to trace, while a simple one like linear or exponential may lose important information. To take a balance, we propose to approximate the tuning curve with a complex model, like two Gaussian kernels [23].

In this paper, we extend a previous work on tracking time variant neuron tuning properties with an exponential-Gaussian complex tuning model, and update the parameters of Monte Carlo point process (MCP) estimator during the decoding. Detail of the algorithm will be explained in section II, after a short review of LNP model. Decoding results both on simulated neural ensemble and in vivo BMI data, using the Dual MCP will be compared with the one estimated by static MCP in section III, followed by the discussion.

II. DATA COLLECTION AND METHODS

A. Data collection

The experiment paradigm was implemented in Qiushi Academy for Advanced Studies in Zhejiang University. A monkey was trained to move a joystick to follow a continuously moving visual target using left hand. A Utah array was chronically implanted in the hand area of the right primary motor cortex (M1). Multi-channel neural activities were recorded by a 128 channel CerebusTM data acquisition system (Blackrock Microsystems, USA) at the sample rate of 30 kHz. Analogy waveforms of the action potential were amplified and band pass filtered from 250 Hz to 7.5 kHz.

About 91 neurons were detected and sorted in real time with the CerebusTM system. An optimum time interval of 10ms was selected to translate the spike trains into a sequence of 1 (spike) and 0 (no spike) as multi-channel point process observations. The corresponding position of the joystick was synchronously recorded at the sample rate of 20 Hz. We collected both data for 1100s.

B. Tuning curve estimation

We selected the instantaneous LNP model [24] to estimate the nonlinear tuning curve of neurons without any further assumption. Suppose the multi-dimension kinematics are defined as $x = [p_x v_x p_y v_y]^T$ with 2-dimensional of position and velocity. As shown in Fig. 1, a linear filter first projects it into the one dimension, where the preferred direction k is estimated from the training set. A neuron can be seen as an inhomogeneous Poisson model. From the spike trains, we can estimate their instantaneous firing probability λ . Then the nonlinear tuning function f can be observed directly in the 2D plane. Unlike other tuning models like linear or exponential one, here places no assumption on the distribution of f .

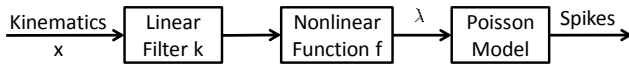


Figure 1. Block diagram of linear-nonlinear-Poisson model [26]

To reduce the number of parameter we need to update, here we try to approximate the tuning curve estimated by the LNP model with one exponential kernel plus one Gaussian function. Fitting result can be expressed as (1)

$$f = \exp(a \cdot k \cdot x) + \exp(-(\frac{k \cdot x - b}{c})^2) + d \quad (1)$$

where $[a, b, c, d]$ are the parameters of tuning function approximated by Levenberg-Marquardt algorithm. We would like to remark at this point that it is not the only option. We also tried two Gaussian or two exponential functions. In our data, one exponential kernel plus one Gaussian is more stable.

C. Dual Monte Carlo Point Process

Monte Carlo point process (MCP) filter was proposed [10] to recursively estimate the probability density function from the discrete observation of the neural firing based on a Bayesian framework, with no constrains on the form of the posterior density. Suppose the kinematic variables at time index k is x_k , and the state evolves according to a linear relation

$$x_{k|k-1} = F x_{k-1|k-1} + \eta_k \quad (2)$$

where F is estimated from training data by a least square solution $(E[x_k^T x_k])^{-1} E[x_k x_{k-1}]$, and η_k is a zero mean white noise with covariance Q . In the inhomogeneous Poisson process, the probability λ of observing a spike ΔN in the interval $(t_{k-1}, t_k]$ is defined as

$$p(\Delta N_k | x_k) = (\lambda(t_k | x_k) \Delta t)^{\Delta N_k} \exp(-\lambda(t_k | x_k) \Delta t) \quad (3)$$

where the instantaneous firing rate $\lambda(t_k | x_k)$ is predicted from the tuning curve $f(x_k)$, which is estimated in section II.B. During the decoding, we estimate the prior probability $x_{k|k-1}$

from x_{k-1} according to (2), and then gain the instantaneous firing rate $\lambda(t_k | x_k)$ from $x_{k|k-1}$ by (1). Finally, the posterior density of x_k can be predicted towards (3). Unlike other point process estimator using Gaussian assumption, MCP estimates the posterior density function in a nonparametric way by using the Monte Carlo sampling. More details please refer to [10].

In traditional MCP, the parameters in tuning function (1) are fixed after training. In contrary, we will update these parameters along the time in Dual MCP. Take the a of the neuron j for example

$$a_k^j = F a_{k-1}^j + u_k^j \quad (4)$$

where F is a transfer matrix, need to be estimated from the training data by least square $(E[a_k^j a_{k-1}^j])^{-1} E[a_k^j a_{k-1}^j]$. u_k is a zero mean white noise with covariance Q_a . Combining the (1), (3), (4) we can form a second MCP and predict the tuning parameters recursively in the same way during the decoding process.

III. RESULTS

In this section, we present the performance tests of the proposed Dual MCP and compare it with original MCP in simulated and real scenarios for neuron decoding.

A. Simulation of Neuron Decoding

To check the performance, we first compare the decoding result by these two methods in simulated neural data. The desire trajectory is collected from the real position recorded from the experiment. The $[b c d]$ are all set to 0 and stay still in the whole simulation time, a starts from 5 and drop gradually with F is 0.99999 and Q_a is 10^{-7} . Two channels of neuron spike are drawn as a Bernoulli random variable with probability $\lambda(t_k) \Delta t$ within each 1ms time window. To control stochasticity, we generate 20 segments of data. The length of each segment is 700s. The first 100s are used for training, the others are used to decode. For each data, we set 5 Monte Carlo trials. The number of particles generated in a Monte Carlo sampling is 1000.

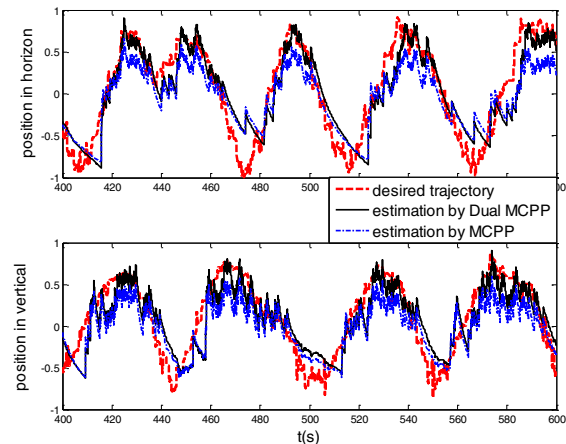


Figure 2. Trajectory reconstruction example by MCP and Dual MCP in simulation for horizon position (above) and vertical position (below)

Fig.2 gives an example of trajectory construction by the MCPP and the Dual MCPP. The dash red line indicates the desire position in horizon (above) and vertical (below), the dash-dotted blue line indicates the estimation by MCPP and the solid black line indicates the estimation by Dual MCPP. We can see the blue line usually fails to reach these peaks, and the error seems to be enlarged as time passed. The normalized mean square error (NMSE) between the model prediction and the true signal are 0.2799 (horizon) and 0.3704 (vertical). In contrary, the black line performs better along the time, and the NMSE are 0.1926 (horizon) and 0.2630 (vertical), which are 31.2% and 29% smaller respectively.

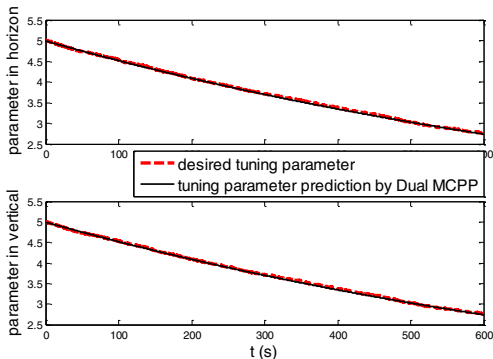


Figure 3. Estimation of tuning parameter by Dual MCPP in simulation

Figure 3 clearly shows the change of tuning parameter in simulated data. The red dash line exhibits the desire tuning property of the generated a for two channels, the black solid line exhibits the estimation result by Dual MCPP. We can see the Dual MCPP precisely follow the change of these tuning parameters. With the tuning parameters updated in time, the Dual MCPP can produce a much closer prediction.

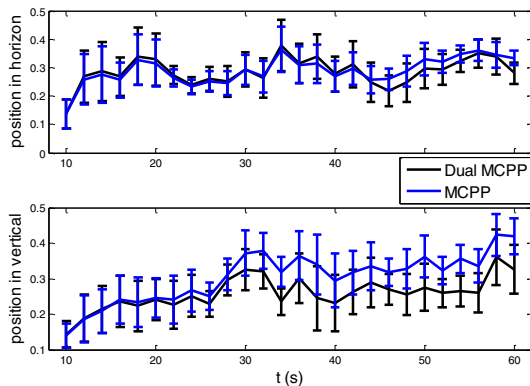


Figure 4. Average NMSE between the trajectory reconstruction and the real trajectory when tuning parameters changes in simulation

To statistically compare the performances by MCPP and Dual MCPP, the average NMSE between the trajectory reconstruction and the desire signal of the 20 segments are evaluated. As shown in Fig. 4, the blue line is the average NMSE by MCPP, while the black line is the average NMSE by Dual MCPP. We can see that though the error of these methods both grow, the Dual MCPP consistently performs better than the MCPP. And the advantage seems to be enlarged as time progresses into testing. The NMSE starts at

0.28% (horizon) and 3.19% (vertical) smaller at 10s, and the drops up to 15.49% and 20.69% smaller respectively after 600s. Moreover, pairwise t-test is performed to see if the advantages are statistically significant. Right tail test $NMSE_{MCPP} > NMSE_{Dual\ MCPP}$ is carried out at $\alpha = 0.05$ significance level. The p-value, which means the probability of observing a sample against our hypothesis, is 0.3419 and 2×10^{-7} respectively.

B. In Vivo Neural Decoding for BMIs

The neural signals recorded in vivo are more complicated to analysis, as the number of neurons is more, besides the neural tuning models and their time-varying properties are unknown. We applied the Dual MCPP on our in vivo recording, hoping to exact more information and compare with the original MCPP.

We select 5 important neurons to analysis their tuning characterizes, which are significantly correlated to the movement in our mutual information analysis. Details of the analysis are described in [25]. The first 200s data are used to train the parameters of tuning model $[a, b, c, d]$ and time-varying parameters F and Q_{α} , which describe how the tuning model changed. To simplify the analysis, $[b, c, d]$ are all set to be still, only a will be updated along the time. The rest data are used to test the decoding performance. We set 20 Monte Carlo trials to control the stochasticity. The number of particles generated in a Monte Carlo sampling is 1000.

Figure 5 gives an example of trajectory reconstruction. The dash red line shows the desired trajectory in horizon (above) and vertical (below), the dash-dotted blue line indicates the estimation by MCPP and the solid black line indicates the estimation by Dual MCPP. We can see though both of these methods can follow the waves, the Dual MCPP produces a better estimation, especially around the peaks and valleys. The NMSE between the desired signal and the prediction by MCPP is 0.2379 in horizon and 0.2473 in vertical. This result drops to 0.2260 and 0.1413 for Dual MCPP, which is 5% and 42.96% smaller respectively.

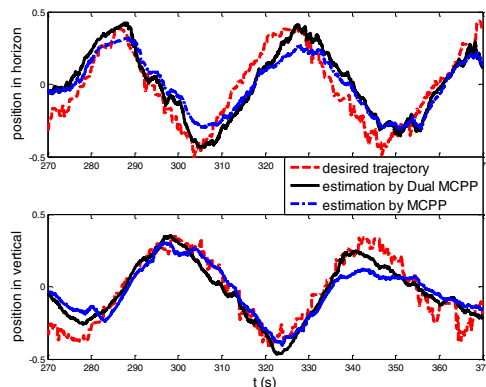


Figure 5. Trajectory reconstruction example by MCPP and Dual MCPP in vivo BMI data for horizon position (above) and vertical position (below)

Figure 6 clearly gives an example for the prediction of tuning curves (neuron 2). The dash-dotted blue line presents the original tuning curve estimated after training, which is used in the MCPP by the whole time. The dashed red line

presents the desired tuning curves through data analysis and the solid black line presents the tuning curves predicted by the Dual MCPP. We can see the black line is much closer to the desired signal than the blue one. As the time-varying neural behavior accumulates, the distance between the blue line and desired red one has been significantly enlarged from 48s to 72s. This indicates that an adaptive decoding algorithm which can track the non-stationary neural tuning property is more promising to gain a steady good estimation.

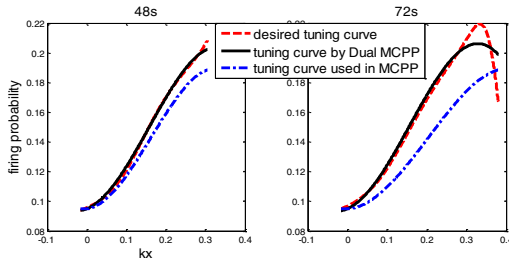


Figure 6. Tuning curve prediction example of neuron 2

IV. CONCLUSION AND DISCUSSION

In this paper, we propose a Dual MCPP estimator to track the non-stationary neural tuning properties for Brain Machine Interfaces. The tuning functions are approximated by an exponential kernel plus a Gaussian one. With a less number of parameters need to be monitored, the time-varying feature of neurons can be captured and the model can be updated in time. When applied to the simulated data and in vivo recording, the Dual MCPP always receives a better reconstruction. Although a systematic comparison on longer data should be tested in the future.

For methods using single neuron activities, the change of the number of neurons recorded usually block these method to be applied on data from days to days. For example, a different neuron may be detected by the same electrode on different days. With a totally different tuning function, it is hardly possible to make a close estimation. Take the multiple units activities may be a compromise choice. However, a flexible model which can select neurons automatically with a close tuning property should be discussed as an extra option.

Besides, the tuning function is a statistical norm to describe the encoding property of a neuron, which is summarized from a long length of data. It not only produces a delay to update the model, but also loses some information during the averaging. Is there a quicker way to follow the instantaneous change of encoding habit shall be a future issue also.

REFERENCES

- [1] J.K. Chapin, K.A. Moxon, R.S. Markowitz and M.A.L. Nicolelis, "Real-time control of a robot arm using simultaneously recorded neurons in the motor cortex". *Nature*. 2,664-670, 1999.
- [2] J. Wessberg, C.R. Stambaugh, J.D. Kralik, P.D. Beck, M. Laubach, J.K. Chapin, J. Kim, S.J. Biggs, M.A. Srinivasan, M.A. Nicolelis, "Real-time prediction of hand trajectory by ensembles of cortical neurons in primates". *Nature*. 408, 361-365, 2000
- [3] M.D. Serruya, N.G. Hatsopoulos, L. Paninski, M.R. Fellows, J.P. Donoghue. "Brain-machine interface: Instant neural control of a movement signal". *Nature*. 416, 141-142, 2002

- [4] M. Velliste, S. Perel, M.C. Spalding, A.S. Whitford, A.B. Schwartz. "Cortical control of a prosthetic arm for self-feeding". *Nature*. 453, 1098-1101, 2008
- [5] J.M. Carmena, M.A. Lebedev, R.E. Crist, J.E. O'Doherty, D.M. Santucci, D.F. Dimitrov, P.G. Patil, C.S. Henriquez and M.A.L. Nicolelis, "Learning to control a brain-machine interface for reaching and grasping by primates". *PLoS Biol.* 1, 193-208, 2003
- [6] S. Zhang, Y. Liao, X. Zheng, W. Chen, Y. Wang, "Decoding the nonstationary neural activity in motor cortex for brain machine interfaces". *International Journal of Imaging Systems and Technology*. 21, 158-164, 2011
- [7] W.Wu, M.J. Black, Y. Gao, E. Bienenstock, M. Serruya, A. Shaikhoumi and J.P. Donoghue, "Neural decoding of cursor motion using a kalman filter". *Adv Neu Inform Process Syst.* 15, 133-140, 2003
- [8] E.N. Brown, D.P. Nguyen, L.M. Frank, M.A. Wilson and V. Solo, "An analysis of neural receptive field plasticity by point process adaptive filtering". *PNAS*. 98, 12261-12266, 2001
- [9] A.E. Brockwell, A.L. Rojas and R.E. Kass, "Recursive bayesian decoding of motor cortical signals by particle filtering". *J Neurophysiol*. 91, 1899-1907, 2004
- [10] Y. Wang, J. Principe and J. Sanchez, "Sequential monte carlo point process estimation of kinematics from neural spiking activity for brain-machine interfaces". *Neural Computation*. 21, 2894-2930, 2009
- [11] C.R. Li, C. P. Schioppa. "Neuronal correlates of motor performance and motor learning in the primary motor cortex of monkeys adapting to an external force field". *Neuron*. 30, 593-607, 2001
- [12] J. Classen, J. Liepert, S.P. Wise, M. Hallett, L.G. Cohen. "Rapid Plasticity of human cortical movement representation induced by practice". *J Neurophysiol*. 79, 1117-1123, 1998
- [13] F. Gandolfo, C.R. Li, C.P. Schioppa, E. Bizzi. "Cortical correlates of learning in monkeys adapting to a new dynamical environment". *PNAS*. 97, 2259-2263, 2000
- [14] S.H. Tillery, D.M. Taylor. "Training in cortical control of neuroprosthetic devices improves signal extraction from small neuronal ensembles". *Reviews in the neurosciences*. 107-119, 2003
- [15] S.P. Kim, J.C. Sanchez, J.C. Principe. "Real-time input subset selection for linear time-variant MIMO systems". *Optimization Methods Software*. 22, 83-98, 2007
- [16] G.J. Gage, K.A. Ludwig, K.J. Otto, E.L. Ionides, D.R. Kipke. "Naïve coadaptive cortical control". *Journal of Neural Engineering*. 2, 52-63, 2005
- [17] W. Wu, N.G. Hatsopoulos. "Real-time decoding of nonstationary neural activity in motor cortex". *IEEE Transactions on Neural systems and rehabilitation engineering*. 16, 213-222, 2008
- [18] U.T. Eden, W. Truccolo. "Reconstruction of hand movement trajectories from a dynamic ensemble of spiking motor cortical neurons". 26th Annual International IEEE EMBS Conference. 4017-4020, 2004
- [19] Y. Wang, J.C. Principe. "Tracking the non-stationary neuron tuning by dual Kalman filter for brain machine interfaces decoding". 30th Annual international IEEE EMBS Conference. 1720-1723, 2008
- [20] Z. Li, J.E. O'Doherty, M.A. Lebedev, M. Nicolelis, "Adaptive decoding for brain-machine interfaces through Bayesian parameter updates". *Neural Computation*. 23, 3162-3204, 2011
- [21] A.P. Georgopoulos, J.F. Kalaska, R. Caminiti, J.T. Massey. "On the relations between the direction of two-dimensional arm movements and cell discharge in primate motor cortex". *J. Neurosci.* 2, 1527-1537, 1982
- [22] L. Paninski, M.R. Fellows, N.G. Hatsopoulos, J.P. Donoghue. "Spatiotemporal tuning of motor cortical neurons for hand position and velocity". *J. Neurophysiol*. 91, 515-532, 2004
- [23] Y. Liao, H. Li, X. Chen, Q. Zhang, Y. Wang and X. Zheng. "Tracking time variant neuron tuning properties of brain machine interfaces". *Applied mechanics and materials*. 461, 654-658, 2014
- [24] Y. Wang, J.C. Principe. "Instantaneous estimation of motor cortical neural encoding for online brain-machine interfaces". *Journal of Neural Engineering*. 7, 1-15, 2010.
- [25] Y. Liao, Y. Wang, X. Zheng, J.C. Principe. "Mutual information analysis on non-stationary neuron importance for brain machine interfaces". 34th Annual international IEEE EMBS Conference. 2748-2751, 2012.