2021 43rd Annual International Conference of the
IEEE Engineering in Medicine & Biology Society (EMBC)
Oct 31 - Nov 4, 2021. Virtual Conference

Considering Neural Connectivity in Point Process Decoder for Brain-Machine Interface*

Shuhang Chen, Xi Liu, Yiwen Wang, *Senior Member, IEEE*

***Abstract*—Brain machine interface (BMI) can translate neural activity into digital commands to control prostheses. The decoder in BMI models the mechanism relating to neural activity and intents in brain. In our brain, single neuronal tuning property and neural connectivity contribute to encoding the intents together. These properties may change, a phenomenon which is named neural adaptation during using BMIs. Neural adaptation requires the decoder to consider the two factors at the same time and has the potential to follow their changes. However, in the previous work, the class of neural network and clustering decoder can consider the neural connectivity regardless of the single neuronal tuning property. On the other hand, point process methods can model the single neuronal tuning property but fail to address the neural connectivity. In this paper, we propose a new point process decoder with the information of neural connectivity named NCPP. We derive the neural connectivity component from the point process method by Bayes' rule and use a clustering decoder to represent the neural connectivity. This method can consider the neural connectivity and the single neuronal tuning property at the same time. We validate the method on simulation data where the point process method cannot achieve a good decoding performance and compare it with sequential Monte Carlo point process method (SMCPP). The results show our method outperforms the pure point process method which indicates our method can model the neural connectivity and single neuronal tuning property at the same time.**

***Clinical Relevance*— This paper proposes a decoder that can model the neural connectivity and the single neuronal tuning property at the same time, which is potential to explain the neural adaptation computationally**

I. Introduction

Brain-machine interface (BMI) can help disabled patients restore their motor functions by controlling prostheses directly from brain [1]. The decoder in BMI system plays an important role which translates neural activity into movement intents. Apart from the translation of neural activity, the decoder mathematically models the mechanism of brain and help us understand how neurons contribute to generate movements. When patients use BMIs, the neural activity responding to the same movement may change adaptively [2, 3]. This phenomenon named neural adaptation mainly including the changes of single neuronal tuning property (such as preferred direction and modulation depth) and neural connectivity. If the decoder is fixed, we cannot keep a stable decoding performance and model the brain correctly. Therefore, to model the brain precisely, the decoder is required to have the ability to adaptively change considering both the changes.

In the previous work, many decoding algorithms were designed and implemented in BMI system, and some of them have the potential to follow a part of neural adaptation. The class of neural network and clustering methods provide the tool to explore neural connectivity, which decode movements from the neural population patterns [4, 5]. The neural population patterns connect the behaviors of the neurons as an ensemble. The methods consider the difference of neural activity patterns in the observation space and project the patterns into the movement space. By updating the parameters of neural network or cluster distribution, the changes of neural connectivity can be followed. However, these methods ignore the single neuronal tuning property because the features of neural activity are delivered in the scale of neural population.

On the other hand, point process decoders can help us to model single neuronal behavior [6-8]. The point process observation can extract the information of single neuronal tuning property from discrete spike timing. A tuning function for each neuron is built after parameterizing the tuning property. The tuning function indicates how a neuron encodes the movement independently. The single neuronal tuning function participate into the decoding as the observation function of state-observation model. By updating the parameters in tuning function, the neural adaptation of single neuronal behavior can be addressed. However, point process methods do not consider the neural connectivity because they assume the neurons independently encode the movement.

The above two classes of decoders only consider one perspective of neural adaptation. But we cannot guarantee what neural adaption happens when subjects use BMIs. If we can consider both perspectives of neural adaptation, the decoding performance may be improved, and the model can be closer to the real brain mechanism.

In this paper, we propose a neural-connectivity point process decoder (NCPP) to consider both the neural connectivity and single neuronal tuning property at the same time which could model the brain more precisely. The proposed decoder introduces a clustering decoder which represents the neural connectivity into the sequential Monte

* Research supported by grants from Shenzhen-Hong Kong Innovation Circle (Category D) (No. SGDX2019081623021543), the National Natural Science Foundation of China (No.61836003), Sponsorship Scheme for Targeted Strategic Partnership (FP902), special research support from Chao Hoi Shuen Foundation and in part by the Innovation and Technology Commission (ITCPD/17-9).

Shuhang Chen is with the Program of Bioengineering, the Hong Kong University of Science and Technology (e-mail: schenbx@connect.ust.hk).

Xi Liu is with the department of Electronic and Computer Engineering, the Hong Kong University of Science and Technology (e-mail: lxi1102@163.com).

Yiwen Wang is with the department of Electronic and Computer Engineering, and with the department of Chemical and Biological Engineering, the Hong Kong University of Science and Technology (e-mail: eewangyw@ust.hk). Yiwen Wang serves as the corresponding author.

Carlo point process decoder (SMCPP). A set of movement particles is generated and propagated in the proposed method to reconstruct the probability density for each time index. Compared to the SMCPP where the associated weight of each particle is only determined by the single neuronal tuning property, the associated weights in NCPP are also proportional to the probability of the movement decoded by the clustering decoder. Therefore, the posterior density of movement at each time index is modulated by the neural connectivity and single neuronal tuning property at the same time. We validate the method on simulation data of a rat two-lever discrimination task and the results shows that the proposed method can improve the decoding performance by introducing accurate neural connectivity information into the point process method. The rest of this paper is organized as followed. Section II introduce the deduction and details of the algorithm. The simulation data and the results are shown in Section III followed by the discussion in Section IV.

II. METHODOLOGY

A. Sequential Monte Carlo Point Process Decoder

Given a discrete observation time sequence $\{t_k\}_{k=0}^{K}$ with a constant time interval $\Delta t = t_k - t_{k-1}$, N_k is defined as the total number of neuronal spikes up to t_k, and $\Delta N_k = N_k - N_{k-1}$ represents the number of spikes within the interval $(t_{k-1}, t_k]$. The observation can be modeled as an inhomogeneous Poisson process with its conditional intensity function $\lambda_k(x_k, \theta_k, H_{k-1})$ which is defined as

$$\lambda_k(x_k, \theta_k, H_{k-1}) = \lim_{\Delta t \to 0} \frac{\Pr(\Delta N_k = 1 | x_k, \theta_k, H_{k-1})}{\Delta t}, \quad (1)$$

where x_k is the movement, θ_k is the tuning parameter at time t_k, and $H_k = [x_{1:k}, N_{1:k}]$ is the history of all the movements and observations up to time t_k. The conditional intensity function is assumed to be a nonlinear observation model represented by

$$\lambda_k(x_k, \theta_k) = f(x_k, \theta_k), \quad (2)$$

The nonlinear function $f(\cdot)$ is assumed to be known and the parameter θ_k is considered fixed in this model. At the time t_k, the posterior density of the movement x_k conditioned on the observation ΔN_k and the history H_{k-1} can be represented by Bayes' rule as

$$p(x_k | \Delta N_k, H_{k-1}) = \frac{p(\Delta N_k | x_k, H_{k-1}) p(x_k | H_{k-1})}{p(\Delta N_k | H_{k-1})}, \quad (3)$$

where $p(\Delta N_k | x_k, H_{k-1})$ is the likelihood of observed spikes within the interval $(t_{k-1}, t_k]$. Due to the assumptions of the conditional intensity function, the likelihood can be simplified as $p(\Delta N_k | x_k)$, which is defined as

$$p(\Delta N_k | x_k) = Poisson\ (\lambda_k, \Delta N_k) = \frac{\lambda_k^{\Delta N_k}}{\Delta N_k!} \exp(\lambda_k). \quad (4)$$

The prior density of the movement $p(x_k | H_{k-1})$ is given by the Chapman-Kolmogorov equation

$$p(x_k | H_{k-1}) = \int p(x_k | x_{k-1}, H_{k-1}) p(x_{k-1} | \Delta N_{k-1}, H_{k-2}), \quad (5)$$

where the movement x_k evolves based on the linear function

$$x_k = F x_{k-1} + r_k. \quad (6)$$

F is the state transition matrix which is trained by the least squares method. r_k is a zero-mean Gaussian noise with covariance R which is estimated by the residue of the linear approximation.

In order to construct the non-Gaussian density of the movement, a set of particles $\{x_k^i\}_{i=1}^{N_s}$ is resampled from the prior density $p(x_k | H_{k-1})$to reconstruct the posterior density with their associated weights $\{w_k^i\}_{i=1}^{N_s}$, where N_s is the number of particles and i is the index of particle. The associated weight is proportional to the likelihood of the observed spikes. And due to the assumption of neural independence, the weight of each particle can be expressed as the product of each neuronal likelihood:

$$w_k^i \propto p(\Delta N_k | x_k^i) = \Pi_{j=1}^{J} Poisson(\lambda_k^j), \quad (7)$$

where J is the number of neurons and j is the neuron index. The posterior density can be formed in the Parzen window estimation [9] as

$$p(x_k | \Delta N_k, H_{k-1}) = \frac{1}{N_s} \Sigma_{i=1}^{N_s} w_k^i \kappa(x_k - x_k^i), \quad (8)$$

where $\kappa(\cdot)$ is a Gaussian kernel and its size is determined by the Silverman's rule [9]. The expectation of the movement on the posterior density $E[x_k | p(x_k | \Delta N_k, H_{k-1})]$ is output as the estimation of movement.

B. Deriving Neural Connectivity from the Point Process Decoder

In order to derive the neural connectivity part from the point process, we introduce the firing probability λ_k as an intermediate variable into the likelihood of observation $p(\Delta N_k | x_k)$ in Eq. 3 to transform the likelihood by Bayes' rule as

$$p(\Delta N_k | x_k) = \frac{p(\lambda_k^- | x_k) p(\Delta N | \lambda_k^+, x_k)}{p(\lambda_k^- | \Delta N_k, x_k)}. \quad (9)$$

where the firing probability λ_k in Eq. 9 can be directly conducted from the smoothed spike trains (labelled as λ_k^-) or be determined by the conditional intensity function Eq. 2 which is related to x_k (labelled as λ_k^+). Here, we assume λ_k in Eq. 9 is from the most directly condition to simplify the calculation. $p(\lambda_k | x_k)$ can be transformed by the conditional probability equation as

$$p(\lambda_k^- | x_k) = \frac{p(x_k | \lambda_k^-) p(\lambda_k^-)}{p(x_k)}, \quad (10)$$

where $p(x_k | \lambda_k^-)$ is the probability density of the movements decoded from the firing probability pattern which is conducted by smoothing the spike trains. This probability contains the information of neural connectivity and we use a clustering decoder to conduct it. $p(\lambda_k^-)$ is the marginal probability of the firing probability which is regardless of x_k. And $p(x_k)$ is the marginal probability of the movement which can be estimated from the training data

$$p(x_k) \propto \frac{1}{Q} \Sigma_q^Q \kappa(x_k - x_q), \quad (11)$$

where $\{x_q\}_{q=1}^{Q}$ is the set of movement samples in training data, Q is the number of training samples and q is the index of training samples. For the probability of observation $p(\Delta N_k|\lambda_k^+, x_k)$, we assume the firing probability is conducted from Eq. 2 so that the probability of observation can be calculated by Eq. 7.

Therefore, for the particles $\{x_k^i\}_{i=1}^{N_s}$, their weights $\{v_k^i\}_{i=1}^{N_s}$ can be represented as

$$v_k^i \propto p\left(\Delta N_k|x_k^i\right) = \frac{p(x_k|\lambda_k^-)}{p(x_k)} \Pi_{j=1}^{J} Poisson(\lambda_k^j), \qquad (12)$$

The posterior density is reconstructed as the Eq. 8, and the expectation of the movement on this density is output as the estimation.

C. The Clustering Decoder in NCPP

A clustering decoder is implemented in NCPP to consider the information of neural connectivity. This is because the clustering decoder estimates the movement according to the neural activity pattern instead of considering single neuronal behavior independently. A set of training samples $\{(x_q, \lambda_q)\}_{q=1}^{Q}$ is used to train the parameters of the clustering decoder where x_q is the movement and λ_q is the corresponding firing probability. Nearest-neighbor clustering is implemented to assign the training samples into the C clusters according to the Euclidean distance, with the centers of the clusters $\{(x_c, \lambda_c)\}_{c=1}^{C}$. Therefore, the probability of movement according to the clustering decoder can be represented as

$$p(x_k|\lambda_k) \propto \frac{1}{C} \Sigma_{c=1}^{C} \kappa(x_k - x_c)\kappa(\lambda_k - \lambda_c), \qquad (13)$$

The conducted probability in Eq. 13 will be used in Eq. 12 to weight each particle to reconstruct the posterior density of the movement.

III. Result

In this section, we want to simulate a scenario where the traditional point process method based on the single neuronal tuning property cannot achieve a good decoding performance. NCPP and SMCPP are implemented and compared in this simulation. A two-dimensional trajectory $\{X_k = [x, y]\}_{k=1}^{10000}$ is generated according to the rat two-lever discrimination task. In this task, each trial records the trajectory from the rest stage (locating around $[0,0]$) to lever-pressing (locating around $[1,1]$ for high lever and $[1,-1]$ for low lever), and then return to the rest stage as shown in the Fig. 1. Each trial contains 200 data samples, and there are 50 trials generated including 25 high-lever trials and 25 low-lever trials. A zero-mean Gaussian noise with 0.1 variance is added to both dimensions of the trajectory.

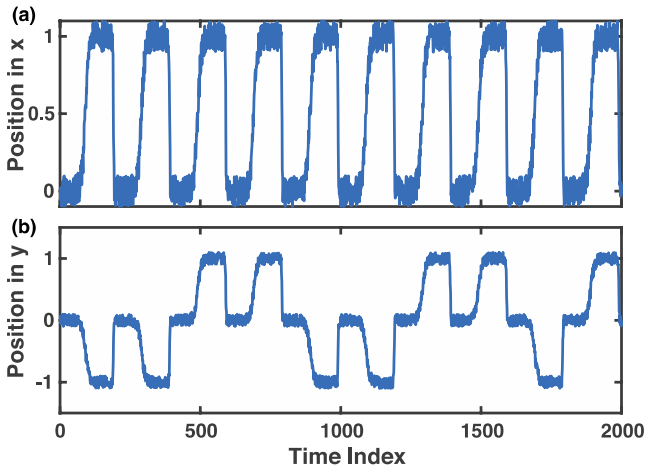

Fig. 1. The simulated movement. The horizontal axis is the time index. The vertical axis of (a) is the position in x and the vertical axis of (b) is the position in y

The firing probability $\{\lambda^j\}_{j=1}^{3}$ are generated followed the group of the conditional intensity functions

$$\begin{cases} \lambda_k^1 = \exp(0.3x_k - 1.4) \\ \lambda_k^2 = \exp(0.21x_k + 0.214y_k - 1.14) \\ \lambda_k^3 = \exp(0.24x_k - 0.18y_k - 1.2) \end{cases} \qquad (14)$$

As shown in the Fig. 2 where the blue curves represent the firing probability of these three neurons, the first neuron can distinguish press or not. The second neuron has higher firing probability for pressing high lever while the third neuron has higher firing probability for pressing low lever. Meanwhile, the high background of the firing probability makes the point process observation is noisy which may result to a bad decoding performance. Finally, the spike trains are generated through a Bernoulli stochastic process with the firing probability $\{\lambda^j\}_{j=1}^{3}$, which are shown in Fig. 2 as the red bars.

70% of the simulation data are used to train the models. The transition function F and R is trained from the training data. The parameters of the conditional intensity function use the ground truth values as Eq. 14. The firing probabilities of training data are clustered into 5 clusters with the distance threshold 0.07. The number of clusters and the distance threshold is explored according to the decoding performance. Fig. 3 shows the distribution of the 5 clusters in the first two important dimensions of principal component analysis (PCA). The red cluster represents the rest stage. The blue cluster and the green cluster represent the movement reaching the low lever and the high lever respectively. And pressing low lever is represented by the black cluster while pressing high lever is

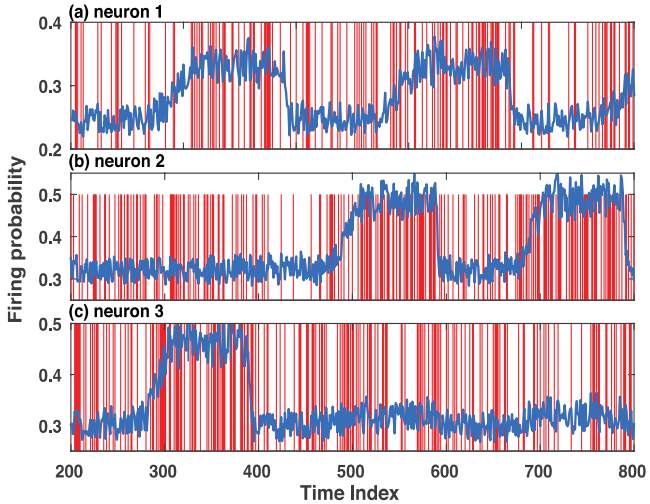

Fig. 2. The firing probability and the spike trains of three neurons. The horizontal axis is the time index, and the vertical axis is the firing probability. The blue lines represent the firing probability. Each red bar represents a spike.

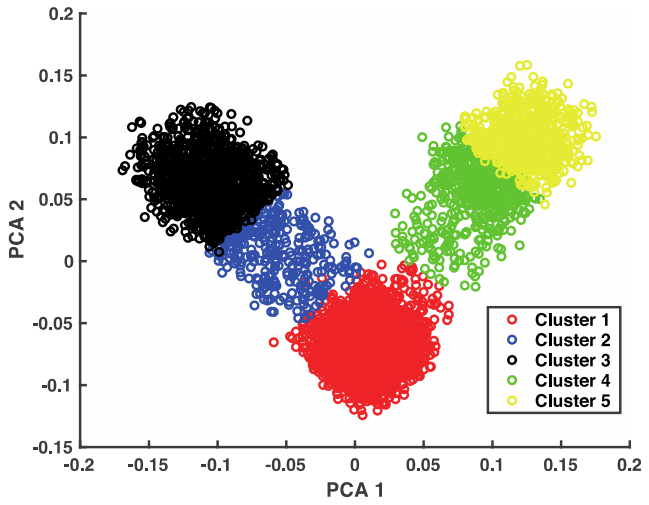

Fig. 3. The clustering result and the distribution of the firing probability pattern of the training data. The horizontal axis is the eigenvalue of the first important component of PCA. The vertical axis is the eigenvalue of the second important component of PCA.

the yellow cluster. The rest 30% of data are used in testing. 500 particles are generated and propagated in the methods, and a gaussian kernel with 20-interval kernel size is used to smooth the spike trains for the clustering decoder.

Fig. 4 shows a segment of the decoding results where the horizontal axis is the time index, and the vertical axis is the value of position. The red lines represent the ground truth of the trajectory. The result obtained by NCPP is represented by the black lines, and the result obtained by SMCPP is the blue lines. For the position in x, mean square error (MSE) between the ground truth and the result of NCPP is 0.3633 while that of SMCPP is 0.4349. And for the position in y, the MSE of NCPP is 0.4109 and that of SMCPP is 0.5191. The validation of MSE demonstrates that the decoding performance of NCPP is better than SMCPP. More specifically at the 3rd, 5th, 6th and 7th trials in the Fig. 4, the result of SMCPP cannot reach the pressing stage while the result obtained by NCPP can catch the stage. This is because the clustering result drive the state of point process to the right location. Therefore, considering the

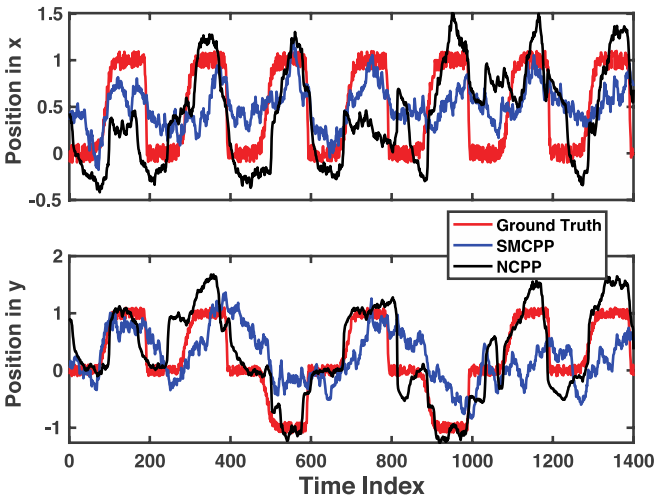

Fig. 4. A segment of the decoding results. The horizontal axis is the time index, the vertical axis is the position in x and the position in y. The red line is the ground truth. The black line is the result obtained by NCPP, and the blue line is the result obtained by SMCPP.

information of neural connectivity can help the point process method improve the decoding performance.

IV. Conclusion

The decoder in BMI can connect the neural activity and external environment or movement. A good decoder can help patients control external devices precisely and help us understand the mechanism of our brain. In the brain, the neural connectivity and single neuronal tuning property both contribute to encode the movement or stimuli. However, most decoders only consider the neural connectivity or the single neuronal tuning property. This may result to a biased model for the brain. In this paper, we propose a new decoder, NCPP, which considers the information of neural connectivity and single neuronal tuning property at the same time. We derive the neural connectivity part from the point process method and use a clustering decoder to represent the information of the neural connectivity. We validate NCPP on the simulation data and compare it to SMCPP. The results show that considering the neural connectivity can help the point process decoder improve the decoding performance. This indicates that our method can model the information of neural connectivity and single neuronal tuning property and has the potential to follow the two perspectives of neural adaptation at the same time. In the future, we will validate the method on real data and use it to track time-varying neural property.

References

[1] M. M. Shanechi, "Brain–machine interfaces from motor to mood," *Nature Neuroscience,* vol. 22, no. 10, pp. 1554-1564, 2019.

[2] K. Ganguly, D. F. Dimitrov, J. D. Wallis, and J. M. Carmena, "Reversible large-scale modification of cortical networks during neuroprosthetic control," *Nature Neuroscience,* vol. 14, no. 5, pp. 662-667, 2011.

[3] Amy L. Orsborn, Helene G. Moorman, Simon A. Overduin, Maryam M. Shanechi, Dragan F. Dimitrov, and Jose M. Carmena, "Closed-loop decoder adaptation shapes neural plasticity for skillful neuroprosthetic control," *Neuron,* vol. 82, no. 6, pp. 1380-1393, 2014,

[4] J. A. Villacorta-Atienza and V. A. Makarov, "Neural network architecture for cognitive navigation in dynamic Environments," *IEEE Transactions on Neural Networks and Learning Systems* vol. 24, no. 12, pp. 2075-2087, 2013.

[5]]J. C. Sanchez, D. Erdogmus, M. A. L. Nicolelis, J. Wessberg, and J. C. Principe, "Interpreting spatial and temporal neural activity through a recurrent neural network brain-machine interface," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 13, no. 2, pp. 213-219, 2005

[6] Y. Wang, A. R. Paiva, J. C. Príncipe, and J. C. Sanchez, "Sequential Monte Carlo point-process estimation of kinematics from neural spiking activity for brain-machine interfaces," *Neural computation,* vol. 21, no. 10, pp. 2894-2930, 2009.

[7] U. T. Eden, L. M. Frank, R. Barbieri, V. Solo, and E. N. Brown, "Dynamic analysis of neural encoding by point process adaptive filtering," *Neural Computation,* vol. 16, no. 5, pp. 971-998, 2004.

[8] M. M. Shanechi, A. L. Orsborn, and J. M. Carmena, "Robust brain-machine interface design using optimal feedback control modeling and adaptive point process filtering," *PLOS Computational Biology,* vol. 12, no. 4, p. e1004730, 2016.

[9] B. W. Silverman, "Using kernel density estimates to investigate multimodality," *Journal of the Royal Statistical Society: Series B (Methodological),* vol. 43, no. 1, pp. 97-99, 1981.