33rd Annual International Conference of the IEEE EMBS
Boston, Massachusetts USA, August 30 - September 3, 2011

Comparisons between Linear and Nonlinear Methods for Decoding Motor Cortical Activities of Monkey

Kai Xu, Yueming Wang, Shaomin Zhang, Ting Zhao, Yiwen Wang, Weidong Chen, Xiaoxiang Zheng*

Abstract— **Brain Machine Interfaces (BMI) aim at building a direct communication link between the neural system and external devices. The decoding of neuronal signals is one of the important steps in BMI systems. Existing decoding methods commonly fall into two categories, i.e., linear methods and nonlinear methods. This paper compares the performance between the two kinds of methods in the decoding of motor cortical activities of a monkey. Kalman filter (KF) is chosen as an example of linear methods, and General Regression Neural Network (GRNN) and Support Vector Regression (SVR) are two nonlinear approaches evaluated in our work. The experiments are conducted to reconstruct 2D trajectories in a center-out task. The correlation coefficient (CC) and the root mean square error (RMSE) are used to assess the performance. The experimental results show that GRNN and SVR achieve better performance than Kalman filter with average improvements of about 30% in CC and 40% in RMSE. This demonstrates that nonlinear models can better encode the relationship between the neuronal signals and response. In addition, GRNN and SVR are more effective than Kalman filter on noisy data.**

I. INTRODUCTION

A BMI system is to set up a communication link between the neural system and machines. It records electrophysiological activities and translates raw neural signals into appropriate commands to drive some devices such as computers and neuroprosthetic devices [1], [2]. Thus, BMIs have potentials in many applications. For example, in rehabilitation, BMIs can help those people with severe physical disabilities to restore some functions.

With the rapid progress of microelectrodes and integrated circuits, spike trains can be easily recorded by the intracerebral microelectrode array. This motivates a large number of BMI systems to use spike trains to figure out a subject's intention. Some existing work use linear models to decode spike signals, including the population vector method [3], [4], the Wiener filter [5], [6], and Kalman filter [7], [8], [9]. Also, a few nonlinear methods have been proposed to attack the decoding problem, e.g., particle filter [10], point process methods [11], [12], and artificial neural networks [13], [14]. Facing the problem of decoding motor cortical activities of the monkey, this paper compares the performance of a linear model, Kalman filter, with two nonlinear models, General Regression Neural Network (GRNN) and Support Vector Regression (SVR). Kalman filter has been proven to be a relatively effective linear method in decoding moving trajectory. General Regression Neural Network is a kind of kernel regression technology which can approximate smooth functions given enough data. Support Vector Regression shows high performance for brain machine interfaces in simulation [15], [16]. Based on spike trains collected from the primary motor cortex of the monkey, we apply the above three methods to reconstruct 2D trajectories. Root mean square error (RMSE) and correlation coefficient (CC) are used to evaluate the performance of these algorithms. The experimental results indicate that the performance of both two nonlinear methods (GRNN: CC = 0.71 $\sim$ 0.85, RMSE = 8.3mm $\sim$ 11.3mm SVR: CC = 0.71 $\sim$ 0.86, RMSE= 7.8mm $\sim$ 10.8mm) are superior to the linear one (KF: CC = 0.34 $\sim$ 0.73, RMSE = 10.9mm $\sim$ 17.3mm).

II. DATA COLLECTION

A macaque monkey was trained to perform a center-out task, i.e., moving a circle controlled by a joystick to a target on a computer screen. The target appears randomly in one of the four locations of a circle. If the monkey can hit the target in a small amount of time, the task is considered to be successfully performed.

If the successful rate of the task was larger than 95%, the monkey was well trained. Then, a 96-electrode Utah array (Blackrock Microsystems Inc., USA) was implanted in the monkey's primary motor cortex (M1). Neural signals were recorded by Cerebus 128TM (Cyberkinetics Neurotechnology Systems, Inc.) at a sampling rate of 30kHz. Positions of joystick were recorded by a micro-controller system with a sampling rate of 20Hz. Both the neural signals and the positions of the joystick were recorded synchronously.

After that, the band range of neural signals were filtered between 250Hz and 7000Hz. The threshold crossing method and template matching method were used to detect and classify spikes. Fig. 1 depicts some examples of the spikes and trajectories of the joystick. The purpose of decoding is to construct the mapping between the neuronal signals and trajectories.

Manuscript received March 26, 2011. This work was supported by grants from the National Natural Science Foundation of China (No. 61031002, No. 61001172, No. 30800287 and No. 60873125), the Zhejiang Provincial Natural Science Foundation of China (No. Y2090707), the Zhejiang provincial key science and technology program for international cooperation(No. 2011C14005), and the Fundamental Research Funds for the Central Universities.

Xiaoxiang Zheng is with the Department of Biomedical Engineering, and also with Qiushi Academy for Advanced Studies, Zhejiang University, Hangzhou 310027, P.R.China. Email: qaas@zju.edu.cn.

Kai Xu, Yueming Wang, Shaomin Zhang, Ting Zhao, and Yiwen Wang are with Qiushi Academy for Advanced Studies, Zhejiang University, Hangzhou 310027, P.R.China.

Weidong Chen is with the College of Computer Science and Technology, and with Qiushi Academy for Advanced Studies, Zhejiang University, Hangzhou 310027, P.R.China.

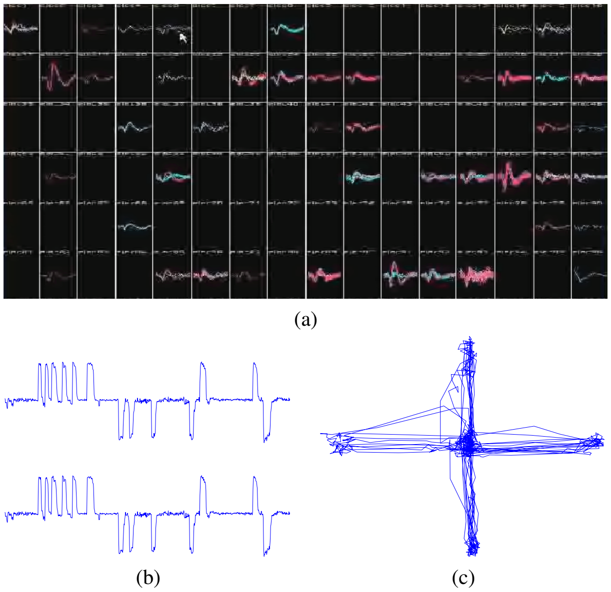


Fig. 1: (a) Spikes detected from primary motor cortex. (b) Trajectory examples of joystick along x and y axes. (c) The typical trajectories of the joystick in 2-D plane in the center-out task.

III. Methods

A. Kalman filter

Suppose the target state at time k is $\mathbf{x}_k$ and the neural activity is $\mathbf{z}_k$, Kalman filter defines two equations:
time update equation:

$$\mathbf{x}_k = A\mathbf{x}_{k-1} + \mathbf{w}_{k-1}, \tag{1}$$

measurement update equation:

$$\mathbf{z}_k = H\mathbf{x}_k + \mathbf{v}_k, \tag{2}$$

where $\mathbf{w}_k$ and $\mathbf{v}_k$ are random variables representing the process and measurement noise respectively. A is a state transition matrix describing the relation between the state at time k-1 and k; H is the observation model which maps the state space to the observed space.

In order to obtain an estimate at each time step, the two equations are alternatively updated. First, we use (1) to produce a priori estimate for the current time step by projecting previous state. Then, (2) is used to obtain a posteriori estimate by incorporating a new measurement into the priori estimate. The details of Kalman filter can be found in [18].

B. General Regression Neural Network (GRNN)

GRNN is a method using Parzen window to estimate the probability density function (pdf) from observed samples. It can be used to deal with regression problems without the assumption of linearity. The key formula of GRNN is as follows:

$$\hat{y}(\mathbf{x}) = \frac{\Sigma_{i=1}^{N} y^i e^{-\frac{D_i^2}{2\sigma^2}}}{\Sigma_{i=1}^{N} e^{-\frac{D_i^2}{2\sigma^2}}}, \tag{3}$$

where D_i is the Euclidean distance between the input $\mathbf{x}$ and the ith observed sample. More details of GRNN can be found in [17]. The only parameter in GRNN model is the bandwidth σ, which decides the smoothness of the pdf curve. If σ is too small, the pdf will be sensitive to every point; if it is too large, it will smooth the pdf curve and lose amount of information about the joint distribution of $f(\mathbf{x}, y)$. In the experiment, we adopted cross-validation method to determine σ by minimizing the root mean square error in training set.

C. Support Vector Regression (SVR)

Support Vector Machine (SVM) is a popular supervised learning algorithm used for classification and regression. It is grounded in the framework of statistical learning theory. Support Vector Regression (SVR) is the method when SVM is applied to regression problems. Some regression technologies find a function $f(\mathbf{x})$ that has the smallest deviation between observed and predicted responses on training data. In order to achieve better generalization performance, instead of minimizing only the error on training set, SVR attempts to minimize both the training error and a regularization term which controls the complexity of the model.

LIBSVM is used in this experiment [19]. The version of SVR we adopt is ε-SVR. Given a training set, $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots (\mathbf{x}_n, y_n)\}$ the ε-SVR seeks a function $f(\mathbf{x})$ whose deviation at each training point does not exceed ε, meanwhile the function itself is kept as flat as possible. The function $f(\mathbf{x}) = <\mathbf{w}, \mathbf{x}> + b$ is obtained by solving the following optimization problem:

$$\begin{aligned} \text{minimize} : & \frac{1}{2}||w||^2 + C\Sigma_{i=1}^{N}(\zeta_i + \zeta_i^*), \\ \text{s.t.} : & -(<\omega, x_i> + b - y_i) \leq \varepsilon + \zeta_i, \\ & <\omega, x_i> + b - y_i \leq \varepsilon + \zeta_i^*, \\ & \zeta_i, \zeta_i^* \geq 0, \end{aligned} \tag{4}$$

where ζ_i, ζ_i^* are non-negative slack variables. The sum of ζ_i, ζ_i^* constitutes the training error. The parameter C is used to make a trade-off between the flatness of function f and the penalty to the training error. More details about support vector regression can be found in [20].

IV. Experiments

The quality of reconstructed trajectory was assessed by two criteria: the correlation coefficient (CC) and the root of mean square error (RMSE). CC denotes the correlation between the prediction and real trajectory, and RMSE is the Euclidean distance between them,

$$CC = \frac{\Sigma_t(x_t - \bar{x})(\hat{x}_t - \bar{\hat{x}})}{\sqrt{\Sigma_t(x_t - \bar{x})^2 \Sigma_t(\hat{x}_t - \bar{\hat{x}})^2}}, \tag{5}$$

$$RMSE = \sqrt{\frac{1}{T}\Sigma_{t=1}^{T}(x_t - \hat{x}_t)^2}. \tag{6}$$

In order to fully assess the ability of the algorithms, we use the cross-validation method to search for the optimal values of the parameters in these models.The smooth parameter σ

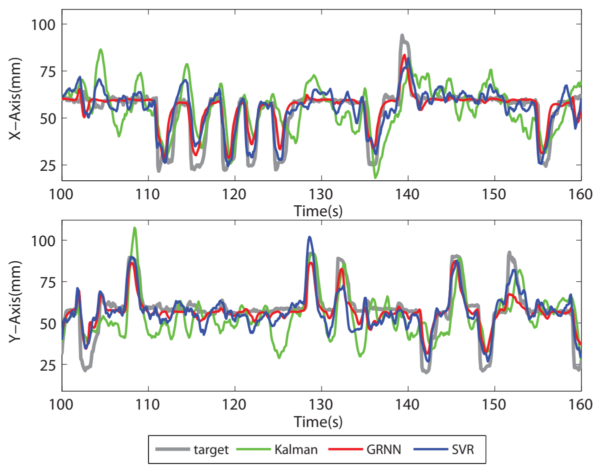


Fig. 2: Comparisons between the real trajectories and the prediction results of the different algorithms. (a) Trajectories of joystick along x-direction. (b) Trajectories of joystick along y-direction.

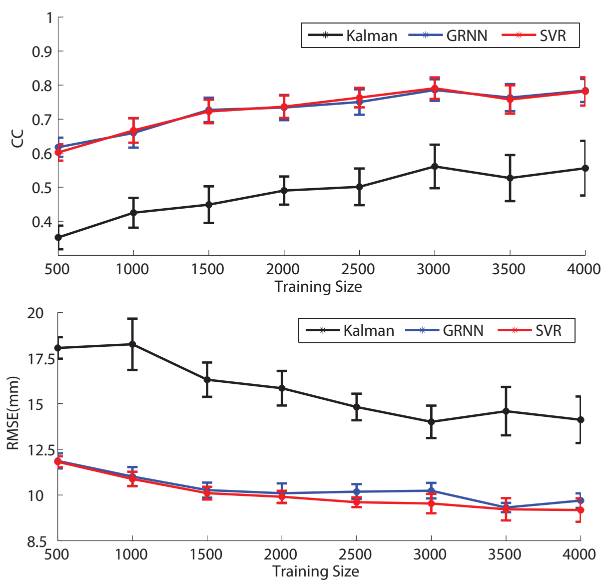


Fig. 3: Changes of CC and RMSE of different decoding methods against the size of training set. The performance of GRNN and SVR are similar and superior to Kalman filter with the same training size.

of GRNN is set to 2.5. In ε-SVR, the radial basis kernel is used. After the grid-searching procedure, the gamma in kernel function is set to be 0.005, parameter C is set to be 2048.

First, we give a visual view of the decoding results of Kalman filter, GRNN, and SVR on a 60-second trajectory. As shown in Fig. 2, it can be found that all results predicted by the three methods approach the real trajectory, while the GRNN and SVR achieve slightly better predictions.

Fig. 3 shows the performance of each method on the different sizes of training set. For a same size of training data, GRNN and SVR perform superior CC and RMSE,

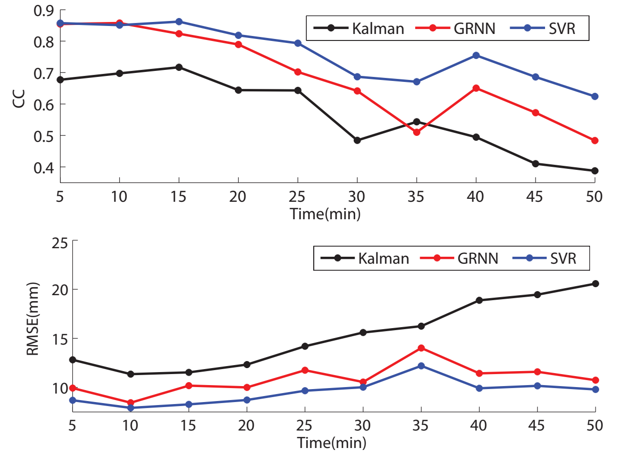


Fig. 4: CCs and RMSEs of different decoding models when tested on data of different time spans.

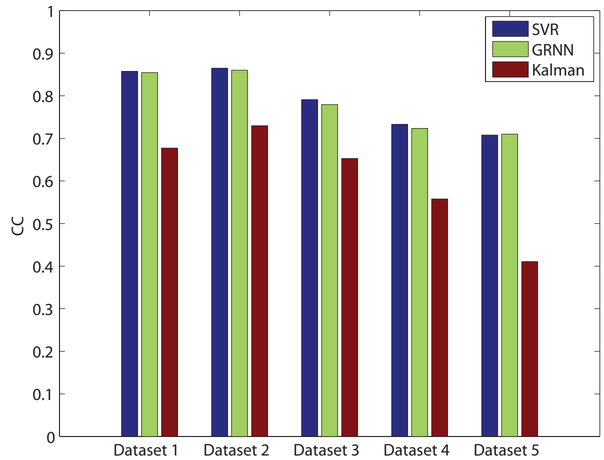
compared with those of Kalman filter. When the training samples increase, the performance of these methods also improves. To reach a similar performance, the size of training data required by GRNN and SVR are much smaller than Kalman filter. It is worth noting that although more training samples makes the models more precise, they also lead to high computational and memory cost. Thus, it is valuable to choose an appropriate size of training data to obtain both high performance and computational efficiency, e.g., 3000 in our experiment. In the subsequent experiments, we use the decoding models trained on the data with the size of 3000.

As the pattern of neural activity can be dynamic, we conduct an experiment to evaluate the decoding models on neural data recorded in different time spans. The models are trained on the data of the first 5 minutes. The rest data are partitioned into several subsets, each having a length of 5 minutes. The CCs and RMSEs of these testing subsets are shown in Fig. 4. As time goes, CC of Kalman filter drops from 0.68 to 0.39, and RMSE increases by about 60%. Comparatively, GRNN and SVR perform much better than Kalman filter. It means that GRNN and SVR are more stable than Kalman filter in a long time period. In addition, SVR performs a little better than GRNN.

Finally, to obtain a general performance for the three models, we collect 5 segments of neural data and the corresponding positions of the joystick. Every segment contains 10 minutes neural activity together with the positions of the joystick. For each data segment, models are trained using the first 5-minute data and the rest 5-minute data are used for testing. Fig. 5 shows the results. In general, GRNN and SVR still perform better than Kalman filter, which demonstrates that the two nonlinear models are more robust than the linear one.

V. Discussion and Conclusion

In this paper, Kalman filter, General Regression Neural Network, and Support Vector Regression are taken as decoding methods. Correlation coefficient (CC) and the root



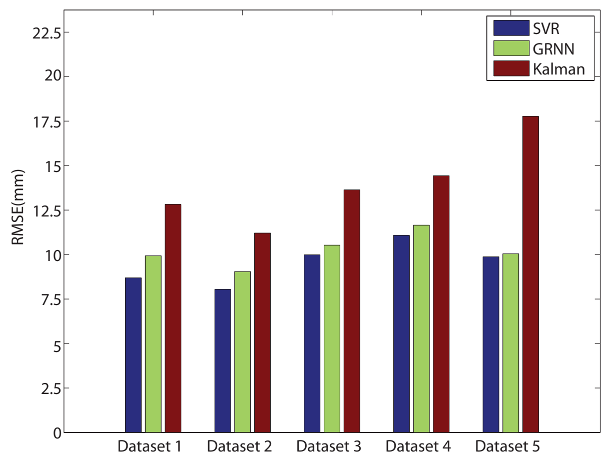


Fig. 5: CCs and RMSEs for on 5 data segments. Kalman filter has the lowest value of CC and highest value of RMSE.

mean square error (RMSE) are adopted to assess the performance of these methods. Decoding results demonstrate that nonlinear methods such as GRNN and SVM can reconstruct the 2-dimensional trajectories in the center-out task more accurately and effectively than Kalman filter.

Although Kalman filter has been successfully used to reconstruct 2-dimensional trajectory in existing work. Our experiments show that it can not produce a satisfactory performance in the center-out paradigm. Besides the limitation of linear assumption between neural activity and movement it makes, Kalman filter also assumes that current state is linear with previous state, it means the trajectory is likely to be a random walk, while as Fig. 1b shows, the trajectories in the center-out task change sharply at some time steps. Thus it is not the best choice for the center-out task or similar paradigms.

The experimental results show that nonlinear methods (GRNN and SVR) have great advantages in building BMI systems. To achieve high performance, less training samples are required, which means the lower computational and memory cost. They work well in the presence of noise. In addition, the stability of these models enable systems to be effective during a long time without the need of retraining. All these characters indicate the superiority of GRNN and SVR in decoding motor cortical activities in BMI systems.

References

[1] P. R. Kennedy, R. A. E. Bakay, M. M. Moore and K. Adams, "Direct control of a computer from the human central nervous system," IEEE Trans. Rehabilitation Engineering, vol. 8, pp. 198-202, Jun. 2000.

[2] M. Velliste, S. Perel, M. C. Spalding, A. S. Whitford and A. B. Schwartz, "Cortical control of a prosthetic arm for self-feeding," Nature, vol. 453, pp. 1098-1101, Jun. 2008.

[3] A. P. Georgopoulos, A. B. Schwartz, and R. E. Kettner, "Neuronal population coding of movement direction," Science, Vol. 233, pp. 1416-1419, Sep. 1986.

[4] R. Wahnoun, J. P. He, S. I. H. Tillery, "Selection and parameterization of cortical neurons for neuroprosthetic control,“ J Neural Eng, vol. 3, no. 2, pp. 162-171, Jun. 2006.

[5] J. M. Carmena, M. A. Lebedev, R. E. Crist, et al., "Learning to control a brain-machine interface for reaching and grasping by primates," PLoS Biol, vol. 1, pp. 193-208, Nov. 2003.

[6] L. R. Hochberg, M. D. Serruya, G. M. Friehs, J. A. Mukand, M. Saleh, et al., "Neuronal ensemble control of prosthetic devices by a human with tetraplegia," Nature, vol. 442, pp. 164-171, July.2006.

[7] W. Wu, M. J. Black, E. Bienensock, M. Serruya, et al, " Neural Deocoding of Cursor Motion using a Kalman Filter," Advances in Neural Information Processing Systems 15, MIT Press, 2003

[8] W. Wu, M. J. Black, D. Mumford, Y. Gao, E. Bienenstock, et al., "Modeling and decoding motor cortical activity using a switching Kalman filter," IEEE Trans. Biomed Eng, vol. 51, pp. 933-942, Jun. 2004.

[9] W. Wu, Y. Gao, E. Bienenstock, J. P. Donoghue, and M. J. Black, "Bayesian population decoding of motor cortical activity using a Kalman filter," Neural Computation, vol. 18, pp. 80-118, Jan. 2006.

[10] A. E. Brockwell, A. L. Rojas, and R. E. Kass, "Recursive bayesian decoding of motor cortical signals by particle filtering," J Neurophysiol, vo. 91, pp. 1899-1907, April. 2004.

[11] Y. W. Wang, A. Paiva, J. Principe, J. 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, Oct. 2009.

[12] Y. W. Wang, J. C. Principe, "Point process modeling on decoding and encoding for Brain Machine Interfaces," Asian Control Conference, 2009. ASCC 2009. 7th, pp. 1000-1005, 2009

[13] J. C. Sanchez, S. P. Kim, D. Erdogmus, et al, "Input-output mapping performance of linear and nonlinear models for estimating hand trajectories from cortical neuronal firing patterns,“ Neural Networks for Signal Processing, 2002. Proceedings of the 2002 12th IEEE Workshop on, pp. 139 - 148, 2002.

[14] J. C. Sanchez, J. C. Principe, J. M. Carmena, M. A. Lebedev, and M. A. Nicolelis, "Simultaneus prediction of four kinematic variables for a brain-machine interface using a single recurrent neural network," Conf Proc IEEE Eng Med Biol Soc, vol. 2, pp. 5321-5324, Sep. 2004.

[15] K. H. Kim, S. S. Kim, and S. J. Kim, "Advantage of support vector machine for neural spike train decoding under spike sorting errors,“ Engineering in Medicine and Biology Society, 2005. IEEE-EMBS 2005. 27th Annual International Conference of the, pp. 5280-5283, Jan. 2006

[16] K. H. Kim, S. S. Kim, and S. J. Kim, "Superiority of nonlinear mapping in decoding multiple single-unit neuronal spike trains:A simulation study," Journal of Neuroscience Methods, vol. 150, pp. 202-211, Jan. 2006

[17] D. F. Specht, "A general regression neural network", IEEE Transactions on. Neural Networks, vol. 2, pp. 568-576, Nov. 1991.

[18] G. Welch, G. Bishop, "An Introduction to the Kalman Filter,“ Available: http://www.cs.unc.edu/∼welch/media/pdf/kalman_intro.pdf

[19] C. C. Chang, C. J. Lin, "LIBSVM : a library for support vector machines,“ Available: http://www.csie.ntu.edu.tw/ cjlin/libsvm

[20] A. J. Smola, B. Schölkopf, "A tutorial on support vector regression,“ STATISTICS AND COMPUTING, vol. 14, pp. 199-222, Aug. 2004