NONLINEAR MODELING AND SYSTEM IDENTIFICATION FOR CORTICAL CONTROL OF ARM PROSTHETICS

Yongji Wang[1], Jian Huang[1], Qi Xu[1], Jiping He[1,2]

[1]*Department of Control Science & Engineering,*
Huazhong University of Science & Technology,
Wuhan, 430074,CHINA;

[2]*Neural Interface Design, the Biodesign Institute,*
and Harrington Department of Bioengieering,
Arizona State University, Tempe, AZ 85287,USA

Abstract: A nonlinear model was established in this paper for the closed-loop system simulating the non-human primates' cortical control of a computer cursor's movement in a 3D virtual reality environment. The model was developed based on artificial neural networks. Elaborately designed experiments were performed on monkeys who have chronically implanted microelectrode arrays in motor cortex areas corresponding to the arm. Monkeys were trained to control the movement of a cursor using cortical signals sampled from the electrodes. Using advanced present system identification tools, the mappings between recorded monkeys' arm movement parameters and neuronal activities in cortex areas were extracted from experimental data sets. This improved substantially the accuracy in predicting arm trajectories from recorded cortical signals than reported in the literature and other methodologies. This new algorithm contributes to the promising application of assisting paralysed people to control neuroprosthetic devices using their thoughts through recorded cortical signals.

Keywords: Cortical control, spike train, extraction algorithm, monkey, nonlinear system identification, artificial neural network

1. INTRODUCTION

Advances in experimental techniques allow neuroscientists to observe and analyze activities from populations of neurons in various cortical areas simultaneously while non-human primates learn to perform various movement tasks. This experimental approach creates data for direct analysis of correlation between patterns of cortical neuron activities and motor behaviours such as arm reaching and grasping tasks. As we gain better understanding of the inter-work relationship between cortical neuronal activities and primates' arm movement trajectories, we can one day apply the new knowledge to develop revolutionary neuroprosthetics that take command from a paralysed individual's thought who has the desire to move but cannot.

Researchers have performed a great deal of experiments to investigate the biological and technical challenges and demonstrated feasibility of training cortical neurons to produce signals to predict the actual movement of the arm or control external devices. Rats were trained to generate signals from their cortex to control a one-dimensional (1D) movement of a robot arm (Chapin et al., 1999). Wessberg et al. applied linear and nonlinear algorithms to monkeys' neuronal ensembles to predict and one or three-dimensional (1D and 3D) real-time arm movement trajectories (Wessberg *et al.*, 2000). Based on data from open-loop experiments the trajectories were recreated from cortical data offline. The "open-loop" experiments here mean that subjects are unaware of the performance of trajectory estimates or device motion predicted from neuronal signals recorded through chronically implanted electrodes in their brain. In these "open-loop" experiments a very high number of cortical neurons (estimates range from 150 to 600 cells) would be required to predict intended movement accurately enough to make this technology practical. Taylor et al. compared the above-mentioned approach to a "closed-loop" paradigm in which subjects had real-time visual feedback of their brain-controlled movement (Taylor et al., 2002). By real-time interaction with the control algorithms subjects could adapt the cortical signals and some neuronal activity patterns to improve the performance of direct cortical control of neuroprostheses. In this experiment, where non-human primates were trained to control the movement of a computer cursor in a 3D virtual environment with visual feedback, the performance of the closed-loop system far exceeded the open-loop with less than 50 cortical neurons.

Results from all these experiments revealed that a robust extraction algorithm to identify a consistent pattern or predict the intention of the brain from a population of cortical neurons is one of the key challenges, in addition to developing more advanced chronically implantable neural probes for access cortical neurons and more appropriate training paradigms. Currently most extraction algorithms are based on analyzing spike trains which consist of time stamps when individual neurons become activated. The frequency of the spikes is assumed to be modulated with parameters of movement. Reasonably accurate trajectories have been produced using current extraction algorithms in intact subjects, with the idea that, if computationally efficient, they will successfully generate robust signals for prosthetic control in subjects who cannot move. Usually we deem that the desired movement parameters are related to firing rates from spike trains of the cortical neuronal ensemble.

There are different extraction algorithms that might be useful for extracting control signals from recorded cortical neuronal activities. Although individual neurons in the primates' motor cortex are only broadly tuned to a particular direction (preferred direction) in the 3D space, which leads to a linear equation for each single cell, the direction of movement was found to be uniquely predicted by the action of a population of motor cortical neurons (Geogropoulos et al., 1986). All weighted preferred direction vectors of individual neurons summed a population vector which was in a direction congruent with the direction of movement exactly. This linear analysis method is usually called population vector algorithm (PVA). Artificial neural network (ANN) has been used for modeling the mapping of movement parameters to cortical neuronal activities (Wessberg et al., 2000, Schwartz et al., 2001). Contrary to above linear algorithms, the ANN approach did not make any priori assumptions about either the physiological properties (for example, shape of the tuning curve) of the single neurons, or the homogeneity of the neuronal population sample. Wessberg et al. used a three-layer ANN to control a robot arm from fewer recorded cortical neuronal signals than expected. Nevertheless they only applied the feed-forward ANNs with linear output units and their model structure was linear in nature.

In our opinion, a nonlinear relationship between arm movement parameters and cortical neuronal activities should yield much more insights to describe the biological systems (Ljung et al., 1987, Sjöberg et al., 1995). Therefore, we adopted the nonlinear approach and treated the mapping between the cortical neurons and the observed or desired behaviour of either a subject's arm or a neuroprosthetic device as nonlinear black-box models. Thus no prior knowledge was necessary in this study. We investigated the applicability of some appropriate type of ANNs for modeling above black-box mappings. We further adopted the neural network based system identification toolbox (NNSYSID) designed specifically to assist identification of various nonlinear dynamical systems (Norgaard et al., 2001). It contains a number of nonlinear model structures based on neural networks, effective training algorithms and tools for model validation and model structure selection. Here we present some results showing that more accurate prediction on movement trajectories from recorded brain signals than previous work can be obtained by applying this useful toolbox.

2. METHODOLOGY

Our experiments use the same equipments and methods as Taylor did in (Taylor et al., 2002) mostly. Rhesus macaques implanted with microelectrode arrays in cortical areas made virtual arm movements in computer-generated, 3D virtual environment by moving a cursor from a center-start position to one of eight targets located radially at the corners of an imaginary cube. The monkeys could not see their actual arm movements but rather saw two spheres (the stationary "target" and a mobile "cursor") with motion controlled by recorded neuronal activities.

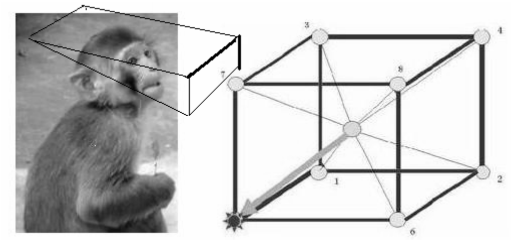
All animal procedures used in acquiring the data described here were performed with University IACUC approval and in accordance with NIH guidelines.

2.1 General setup

The stereo image was controlled by an SGI Octane workstation and viewed through polarized lenses and a 96 Hz light-polarizing shutter screen. Three-dimensional wrist position was sent to the workstation at 100 Hz from an Optotrak 3020 motion tracking system. Cortical activity was collected via a Plexon Data Acquisition system. Spike times were transferred to the workstation, and a new brain-controlled cursor position was calculated every 30msec.

2.2 Experiment

The left arm was restrained while the right arm was free to move during both hand- and brain-controlled movement blocks (Fig. 1). Cursor radius was 1 cm. Target and center radii were 1.2 cm. A liquid reward was given when the cursor boundary crossed the target boundary for 300 ms or more. Radial distance (center start position to center of target) was 4.33 cm under brain-control.



(A) (B)

Fig. 1. Real-time 3D brain-control experimental setup. (**A**) A mirror in front of the monkey's face reflects a 3D stereo image of the moving cursor and static target projected from above. On the other side of the mirror, the monkey moves one arm with a position sensor taped to the wrist. The 3D position of the cursor is determined by either the position sensor or by the movement predicted by the subject's cortical activity ("brain-control"). The cursor and target are shown with dotted outlines in the monkey's workspace, but do not physically exist. The monkey sees the cursor and target as being in the workspace but cannot see its own arm. The movement task is a 3D center-out task. The cursor is held at the central position until a target appears at one of eight radial locations which form the corners of an imaginary cube. The center of the cube is located distal to the monkey's right shoulder. (**B**) shows the approximate relative cube and target size during the hand-control task from the same observer view point as in (A).

In this study, we only pay attention to the closed-loop experiments, i.e. the “brain-control” movement task. At the beginning of each performance, the cursor automatically moves to the eight targets directly for some times, which can be seen by monkeys. Meanwhile neuronal activity in primates’ cortical areas is recorded and processed. We assume that monkeys had the intention to move the cursor to the same target as they saw. So the simultaneous cortical activity must reflect the intention and we can establish a numerical model to describe the relation between neural activity and cursor movement. This model is used to translate cortical signals into trajectories by the computer. After trained, monkeys can almost perform this task with a little help by the computer. That is, when the cursor did not move to the correct direction, the computer will add a little vector into the real computed cursor position from above-mentioned model so as to help it come to the right target.

3. MAIN RESULTS

Predictions of cursor position based on simultaneous neural ensemble firing were obtained by applying both a linear ARX model and a nonlinear NNARX model to these data. Here we only perform offline analysis of these two models. Our aim is to demonstrate that the nonlinear model will present more accurate prediction than the other one, if same random selected neurons are used in both of them.

3.1 System Data Recorded

In the above-mentioned closed-loop experiment, the cursor’s movement is generated by neuronal activities of monkeys’ cortex area, or by the intention of these animals. This brain control system is viewed as a black-box model. See Fig. 2. Outputs of the system are the measured movement trajectories of the cursor, which are represented in 3D coordinate data. Otherwise inputs of the system consist of cortical neuronal activities, which are recorded by multi-channel microelectrode arrays simultaneously.

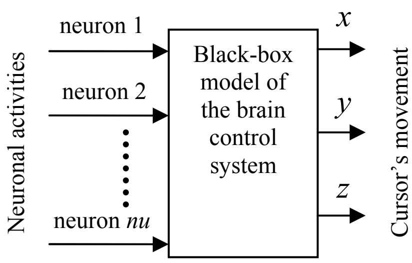


Fig. 2. The framework of the brain control system

Traditionally the input neural data is treated as discrete events marked at times when extra-cellular potentials indicate an action potential has fired in a neuron close to the electrode. This leaves a list of times when action potentials, commonly called ‘spikes’, were detected. Usually the list of times is transformed into a continuous variable in some way. Most methods that perform this transformation are estimating the mean firing rate of a neuron of over some time interval. The simplest one is employed here. In this method constant temporal bins are formed and the number of events that taken placed in a bin are counted. The mean firing rate example of one neuron is shown in Fig. 3. Assume there are *nu* input neurons. Thus we can arrange the total neuronal firing rate data into a vector $U(t)$, where $U(t) = [u_1(\mathrm{t}), u_2(t), \ldots, u_{nu}(t)]^{\mathrm{T}} \in R^{nu}$, $u_i(t)$ is the firing rate series of the ith neuron ($i = 1, 2, \ldots, nu$).

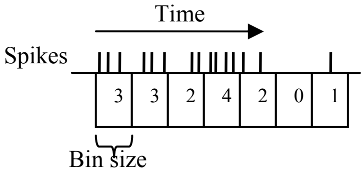


Fig. 3. Construction of the ith neuron firing rate series. Here $u_i(t) = \{\ldots, 3, 3, 2, 4, 2, 0, 1, \ldots\}$

On the other hand, the cursor's position (system output) is represented by a list of 3D coordinate data whose components are labeled as x, y and z separately. The 3D cursor trajectories of the whole experiment are depicted in Fig. 4. Similarly we arrange the output data into a vector $Y(t)$, where $Y(t) = [x(t),\ y(t),\ z(t)]^{\mathrm{T}} \in R^3$. Note that the synchronization between $U(t)$ and $Y(t)$ must be guaranteed when processing the recorded data.

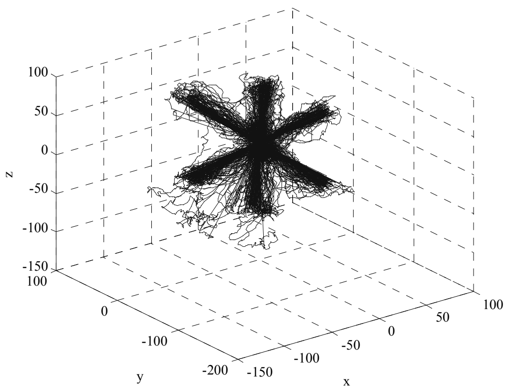


Fig. 4. Cursor positions' trajectories of the whole "brain-control" experiment

3.2 Model Structure

A *model* reflecting the relationship among observed signals is required for further study. Since there is little prior knowledge or physical laws about the cortical neurons, a black-box model is established. As a result little pre-processing is needed to the recorded data, which is unlike the traditional population vector (PV) method. Here we select both linear and nonlinear model structure and assess the performance of them.

In the linear case, the selected model structure between the neuronal discharge in the vector $U(t)$, and the 3D cursor position in $Y(t)$ is

$$A(q)Y(t) = B(q)U(t) + \varepsilon(t) \qquad (1)$$

where $U(t) \in R^{nu}$, $Y(t) \in R^3$, $A(q) \in R^{3\times 3}$ is an three-by-three matrix whose entries are polynomials in the delay operator q^{-1}. You can represent it as

$$A(q) = I_3 + A_1 q^{-1} + \cdots + A_{n_a} q^{-n_a}$$

where I_3 means the three-by-three identity matrix. Similarly, $B(q) \in R^{3\times nu}$ is an *three*-by-*nu* matrix satisfies

$$B(q) = B_0 + B_1 q^{-1} + \cdots + B_{n_b} q^{-n_b},$$

$\varepsilon(t)$ are the residual errors. For more detailed ARX model descriptions see the reference (Ljung et al., 2004).

Otherwise in the nonlinear case, the artificial neural network NNARX model structure (Norgaard et al., 2001) was selected. This model is described by

$$\begin{cases} \text{Regression vector:} \\ \varphi(t) = [Y(t-1) \cdots Y(t-n_a)\ \ u(t-n_k) \\ \qquad \cdots u(t-n_b-n_k+1)] \\ \text{Predictor:} \\ \hat{Y}(t\,|\,\theta) = \hat{Y}(t\,|\,t-1,\theta) = g(\varphi(t),\theta) \end{cases} \qquad (2)$$

where $\varphi(t)$ is a vector containing the regressors, θ is a vector containing the weights and g is the function realized by the neural network.

3.3 Parameter Identification

Offline analysis was performed to test the validity of this model. Neuronal discharges were counted in 20ms bins. By randomly selected 7 to 10 neurons, n_a=1, n_b=1, n_k=1, both linear and nonlinear models were identified. There are about 35,000 samples of the observed data from the closed-loop experiment. We use the preceding 20,000 samples (i.e. U(:,1:20000) and Y(:,1:20000) in MATLAB representation) to approximate the model parameters. The rest part is left for the model validation.

In the linear case, parameter matrices $A(q)$ and $B(q)$ were calculated in the MATLAB. One-step ahead of prediction of the cursor position was achieved by directly using the obtained linear model. For example, selecting nine neurons (neuron 1 to neuron 9) as the inputs of the system model (1), we can compute $A(q)$ and $B(q)$ as follows by applying the MTALAB identification toolbox.

$$A_1 = \begin{bmatrix} -0.9836 & 0 & 0 \\ 0 & -0.9856 & 0 \\ 0 & 0 & -0.9822 \end{bmatrix},$$

$$B_0 = [0]_{3\times 9}$$

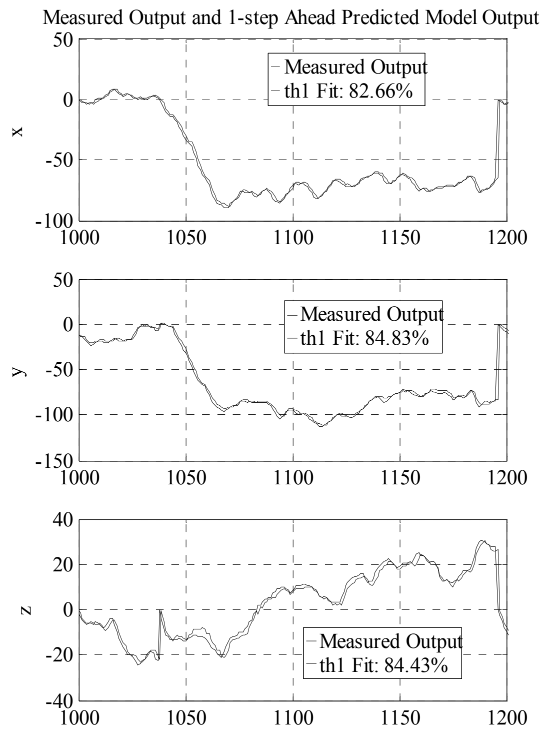
$$B_1 = \begin{bmatrix} 0.1178 & -0.0186 & 0.0305 & \cdots \\ -0.0385 & 0.5082 & 0.0125 & \cdots \\ 0.0071 & 0.2085 & -0.1587 & \cdots \end{bmatrix}_{3\times 9}.$$

After training the selected neural network with the well-known BP algorithm or L-M method, an appropriate NNARX model was identified in the nonlinear case. The same procedure was performed on this model as we did in the linear case. Here we select 6 hidden-layer units for the training neural network. The trained weight matrix W_1 and W_2 are list as follows.

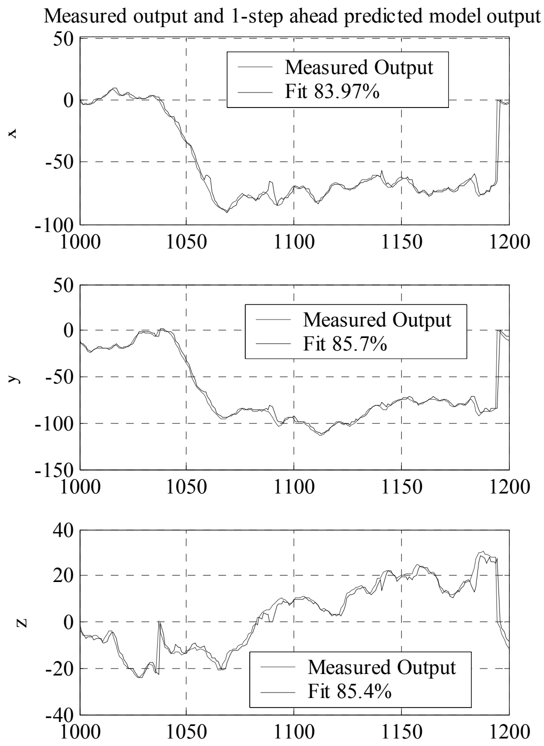
$$W_1 = \begin{bmatrix} -0.0315 & -0.0603 & -0.0199 & \cdots \\ 0.3139 & -0.0916 & 0.0001 & \cdots \\ -0.0062 & 0.0481 & 0.0309 & \cdots \\ 0.1739 & -0.0481 & -0.0345 & \cdots \\ 0.0252 & -0.0510 & -0.0257 & \cdots \\ -4.1573 & 0.8681 & 0.4971 & \cdots \end{bmatrix}_{6\times 13},$$

$$W_2 = \begin{bmatrix} -2.2255 & 0.9807 & 2.8436 & \cdots \\ -1.9410 & 0.0008 & 3.0455 & \cdots \\ 2.4499 & -0.6070 & 3.0471 & \cdots \end{bmatrix}_{3\times 7}$$

In Fig. 5, the upper three plots (a) refer to trajectories and the *fit* value in the linear model and the lower three plots (b) the ANN nonlinear model. From the plots we can see that both models have a good prediction performance. Though the nonlinear one using artificial neural network gains a little advantage over the ARX model. Even if the advantage is only about one percentage in the *fit* value.



(a)
Linear model (th 1)



(b)
ANN model

Fig. 5. The intercepted 3D trajectories of measured outputs and both the two models' one-step ahead predicted outputs when nine neurons were selected as inputs.

3.4 Model Validation

We assess the accuracy of the identified model by calculating the percentage of the output variation, which is explained by

$$fit = 100 \cdot \left(1 - \left\|\hat{Y} - Y\right\| / \left\|Y - \overline{Y}\right\|\right) \% \qquad (3)$$

where $\hat{Y}$ is the predicted output vector, $\overline{Y}$ is the mean value of the elements in Y. The more this calculated *fit* is closer to 1, the more accurate this linear model is.

By randomly selected 7 to 10 neurons, both linear and nonlinear models were established and compared. The *fit* value which describes the accuracy of the two models are listed in Table 1. It is can be seen that the accuracy of the NNARX model is somewhat higher than the linear one.

Table 1. Comparing the accuracy of two types of models

	fit (N=7)		fit (N=8)	
	ARX	NNARX	ARX	NNARX
x	82.65	83.89	82.65	83.82
y	84.8	85.67	84.81	85.53
z	84.7	85.24	84.33	85.42

	fit (N=9)		fit (N=10)	
	ARX	NNARX	ARX	NNARX
x	82.66	83.97	82.66	83.95
y	84.83	85.7	84.84	85.7
z	84.43	85.4	84.43	85.37

4. CONCLUSION

This study applies both linear and nonlinear methods to model the deliberate close-loop "brain-control" experiment. Offline analysis was performed only. Results showed that the prediction accuracy of the NNARX model is a little higher than the linear ARX method. Although the advantage is small, it emerged in three components of the 3D cursor trajectories simultaneously. Therefore we can conclude that using nonlinear methods to model the cortical control system may prevail over linear ones. These nonlinear extraction methods hold hope for a possibly more accurate interface capable of adding quality to the life of paralyzed people.

ACKNOWLEDGEMENT

This work was partly supported by NSF China under Grant No. 69974017, 60274020, 60340420431. This is greatly acknowledged.

REFERENCES

Chapin J.K., Moxon K.A., Markowitz R.S. and Nicolelis M.A.L. (1999). "Real-time control of a robot arm using simultaneously recorded neurons in the motor cortex" *Nature Neurosci*, Vol. 2, pp. 664-670, 1999.

Georgropoulos A.P., Schwartz A.B. and Kettner R.E. (1986). "Neuronal population coding of movement direction" *Science*, Vol. 233, pp. 1416-1419, Sep. 1986.

Ljung L. (1987) "*System identification: theory for the user*" Prentice-Hall, Englewood Cliffs, NJ.

Ljung L. (2004). "*System identification toolbox user's guide"* COPYRIGHT 1988 - 2004 by The MathWorks, Inc.

Norgaard M., Ravn O. and Poulsen N.K. (2001). "NNSYSID and NNCTRL tools for system identification and control with neural networks" *Computing & Control Engineering Journal*, Vol. 12, pp. 29-36, Feb. 2001.

Schwartz A.B., Taylor D.M. and Tillery S.I. Helms (2001). "Extraction algorithms for cortical control of arm prosthetics" *Current Opinion in Neurobiology*, Vol. 11, pp. 701-708, Dec. 2001.

Sjöberg J., Zhang Q.H. and Ljung L. et al. (1995). "Nonlinear black-box modeling in system identification: a unified overview" *Automatica*, Vol. 31, No. 12, pp. 1691-1724, 1995.

Taylor D.M., Tillery S.I. Helms and Schwartz A.B. (2002). "Direct cortical control of 3D neuroprosthetic devices" *Science*, Vol. 296, pp. 1829-1832, Jun. 2002.

Wessberg J. et al. (2000). "Real-time prediction of hand trajectory by ensembles of cortical neurons in primates" *Nature*, Vol. 408, pp. 361-365, Nov. 2000.