CORTICAL CONTROL OF A ROBOT USING A TIME-DELAY NEURAL NETWORK

M. Burrow[1], J. Dugger[1], D. R. Humphrey[2], D. J. Reed[2], L. R. Hochberg[2]
[1]Georgia Institute of Technology, Biomedical Interactive Technology Center, Atlanta, GA USA
[2]Emory University, Department of Physiology, Atlanta, GA, USA

Abstract

Research into cortically controlled prosthetic devices for paralyzed individuals has resulted in some preliminary success. A system in which a robot arm tracks the actual wrist position of a monkey given corresponding neural signals from the animal's motor cortex has been developed. The system uses a Time-Delay Neural Network (TDNN) to transform a time sequence of multichannel neural activity into a sequence of wrist positions.

Introduction

The ability of an individual with severe spinal cord injury to control a prosthetic device, such as a robot arm, by thinking of the desired movement depends on a control interface that can successfully convert activity in the motor cortex into a desired prosthetic movement. The control interface described in this paper uses a neural network to perform this conversion. Use of a neural network allows the conversion to be made independent of an explicit, physiologically based system model, since a neural network can learn the desired input-output relationship. The particular neural network used in this research was a TDNN. A TDNN is simply a feedforward neural network whose input vector is a tapped delay line. The tapped delay line allows the feedforward network to learn spatiotemporal patterns.

Cortical Input Signals

Microelectrode arrays implanted in the motor cortex of a rhesus monkey provide 24 channels of neural input for the prosthesis control system. Each of the signal channels from this array records the level of activity in the area of the motor cortex corresponding to movement of the arm or hand.

Experimental data was obtained by recording motor cortex activity from the monkey while it performed trained movements. The trained movement involved following a target on a computer screen with a cursor. The monkey, whose arm was restrained, controlled the cursor by moving a joystick. A potentiometer attached to the joystick measured the monkey's wrist position, which was then digitized and stored in a file with the recorded neural data. Thirteen flexion and

This work is supported by contract N01-NS-1-2308 of the NIH Neural Prosthesis Program, D. R. Humphrey, P.I.

thirteen extension trials were extracted from this data stream for training and testing the prosthesis control system. These trials were selected as the best examples of their respective movements.

Data reduction procedures were used to determine which of the 24 channels would be most relevant to wrist movement. First, channel variation during wrist movement was examined using a signal formed by alternately concatenating all of the extension and flexion trials to mimic the original data stream. Channels with the highest variance over time were chosen as those most relevant to movement. Second, the repeatability of each channel for the same type of movement was measured by selecting those channels having the highest average autocorrelations over multiple trials of the same movement. The repeatability test validated the results of the variance test. These tests thus provided reasonable measures of channel relevance. Only two of the original 24 channels were selected to serve as training and testing data.

Prosthesis Control System

Input signal preprocessing operations and the TDNN comprise the prosthesis control system. A block diagram of the system appears in Figure 1. Eight of the thirteen extension/flexion trial concatenations were used to train the TDNN and the remaining five were used as testing data.

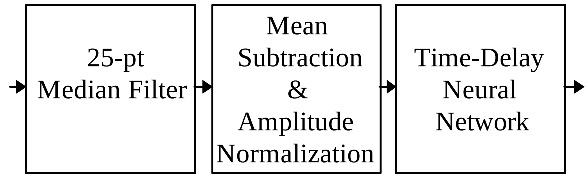

Figure 1: Neural Prosthesis Control System

Preprocessing of the input signals consists of median filtering, mean subtraction, and amplitude normalization. Median filtering eliminates spikes in the data while preserving the general shape of the signal. Mean subtraction and amplitude normalization were found to improve TDNN training. Input signals to the prosthesis control system are normalized using the means and amplitude factors of the training data.

A schematic of the TDNN appears in Figure 2.

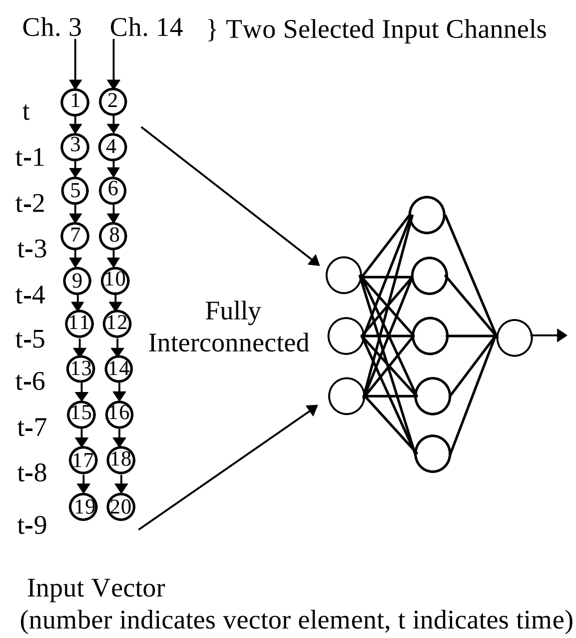

Figure 2: Time-Delay Neural Network

The TDNN is essentially a feedforward neural network whose input vector is a tapped delay line. In this case, the TDNN has an input buffer of 10 successive samples for both channels. The output is calculated using the current values of both channels, as well as the past nine values of both channels. Thus the input vector to the feedforward neural network is 20 dimensional. The tapped delay line input allows the feedforward neural network to learn temporal information in the neural data. The TDNN has two hidden layers with 3 nodes in the first hidden layer and 5 in the second. Two hidden layers are sufficient to approximate any function to a given accuracy provided there are enough units per layer [1]. Experimentation yielded the numbers of hidden layer nodes used in this network. There is only one output node which provides a normalized wrist position between +1 representing full extension and -1 representing full flexion. The output values can be scaled to the desired range of angular position. The network was trained using backpropagation with momentum.

Figure 3 illustrates the normalized monkey wrist position versus wrist position computed by the TDNN from the testing data set. Although the TDNN output exhibits some glitches, its general pattern fits well to that of the actual wrist position.

Software has been developed in Visual Basic to perform these signal processing tasks. This software plots the results shown in Figure 3 while simultaneously controlling the wrist motor of an Alpha II robot arm to flex and extend from -45 to +45 degrees.

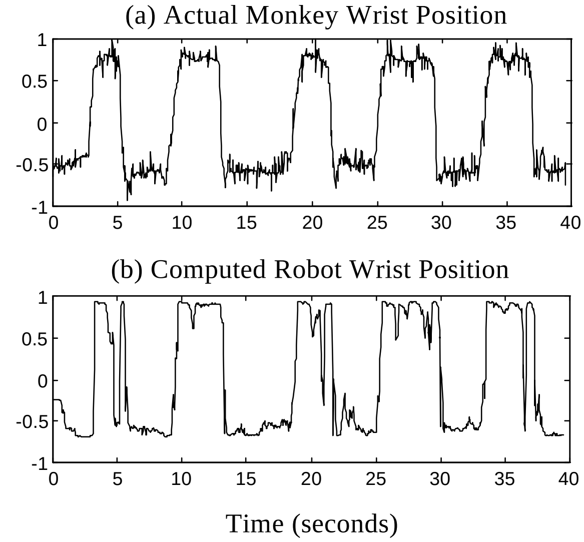

Figure 3: Actual vs. Computed Wrist Position

Conclusion

The results illustrated in Figure 3 illustrate that time-delay neural networks have the ability to convert time-series motor cortex data into robot arm control signals. Possible improvements in network performance will be examined using different hidden layer sizes when training with additional data from the same monkey. Network behavior and training using data recorded from a different monkey will also be examined. If these studies with wrist movement yield positive results, then similar methods will be examined for control of the entire arm.

Reference

[1] Hertz, J., Krogh, A., Palmer, R. G., Introduction to the Theory of Neural Computation. Redwood City, CA: Addison-Wesley, 1991.

Address

Mike Burrow
250 14^{th} St. NW
GCATT Rm. 238
Atlanta, GA 30318
USA