



Published in final edited form as:

*Conf Proc IEEE Eng Med Biol Soc.* 2009 ; 2009: 4535–4538. doi:10.1109/IEMBS.2009.5334129.

## Cortical Decoding of Individual Finger and Wrist Kinematics for an Upper-Limb Neuroprosthesis

**Vikram Aggarwal,**

Department of Biomedical Engineering at The Johns Hopkins University, Baltimore, MD

**Francesco Tenore,**

Johns Hopkins University Applied Physics Laboratory (APL), Laurel, MD

**Soumyadipta Acharya,**

Department of Biomedical Engineering at The Johns Hopkins University, Baltimore, MD

**Marc H. Schieber,** and

Department of Neurology, Neurobiology, and Anatomy, at the University of Rochester Medical Center, Rochester, NY

**Nitish V. Thakor**

Department of Biomedical Engineering at The Johns Hopkins University, Baltimore, MD

### Abstract

Previous research has shown that neuronal activity can be used to continuously decode the kinematics of gross movements involving arm and hand trajectory. However, decoding the kinematics of fine motor movements, such as the manipulation of individual fingers, has not been demonstrated. In this study, single unit activities were recorded from task-related neurons in M1 of two trained rhesus monkey as they performed individuated movements of the fingers and wrist. The primates' hand was placed in a manipulandum, and strain gauges at the tips of each finger were used to track the digit's position. Both linear and non-linear filters were designed to simultaneously predict kinematics of each digit and the wrist, and their performance compared using mean squared error and correlation coefficients. All models had high decoding accuracy, but the feedforward ANN ( $R=0.76-0.86$ ,  $MSE=0.04-0.05$ ) and Kalman filter ( $R=0.68-0.86$ ,  $MSE=0.04-0.07$ ) performed better than a simple linear regression filter ( $0.58-0.81$ ,  $0.05-0.07$ ). These results suggest that individual finger and wrist kinematics can be decoded with high accuracy, and be used to control a multi-fingered prosthetic hand in real-time.

### I. Introduction

Previous work has shown that neuronal ensemble activity from various motor areas can be used to continuously predict the kinematics for gross movements of a single effector, such as during reach [1,2] or control of a computer cursor [3,4]. However, in order to achieve neural control of advanced upper-limb neuroprostheses, there is also a need to develop Brain-Machine Interfaces (BMI) for dexterous movements, such as the manipulation of individual fingers.

We have recently demonstrated neural decoding of *discrete* flexions and extensions of individual fingers and the wrist [5,6], but for truly dexterous control of a multi-fingered prosthetic hand it will be necessary to *continuously* decode the kinematics of multiple digits. Single-unit activities were recorded from a population of neurons in the primary motor cortex

(M1) hand area of two male rhesus monkeys during individuated flexion and extension movements of each digit and the wrist. Simultaneous kinematics of each digit were obtained through strain gauges mounted on microswitches on the tip of each finger.

Popular methods for decoding activity from neuronal ensembles include population vectors (PVs) [7], linear filters, and artificial neural networks (ANNs). PVs have proven successful in decoding discrete movements in center-out tasks [7], and may not be well suited for continuous decoding. Linear regression filters and ANNs have been effective for real-time neural control of a 2D cursor [4] and prediction of hand trajectory [2], but both lack a clear probabilistic model and do not incorporate temporal information. More recently, Kalman filters have been used to decode hand position [8] and trajectory of a computer cursor [9] using a recursive, probabilistic approach.

In order to determine the best approach to model the data for this particular task, three different algorithms were used to decode the kinematics of each finger and the wrist from the same population of M1 neurons – a linear regression filter, a feedforward ANN, and a Kalman filter. The results of each model were then compared using the mean squared error and Pearson correlation coefficients.

This work demonstrates how neural activity can be used to simultaneously predict the kinematics of multiple end-effectors, and lays the foundation for dexterous manipulation of a multi-fingered hand neuroprosthesis.

## II. Methods

### A. Experimental Setup

Two male rhesus monkeys (monkey C, monkey K) were trained to individually flex or extend each digit and the wrist of the right hand by operating a pistol-grip manipulandum (see Fig. 1, inset). By flexing or extending each digit a few millimeters, the monkey closed microswitches at the tips of each finger as shown in Fig. 1. The position of each finger was obtained from switch-mounted strain gauges on either sides of the fingertip, and a potentiometer transduced wrist flexion and extension.

A row of LEDs above each switch were illuminated instructing the monkey to perform 12 distinct movements. Each instructed movement is abbreviated with the first letter of the movement type (f=flexion, e=extension), and number of the instructed digit (d1=thumb... d5=little finger, d6 or w=wrist; e.g. 'e4' indicates extension of ring finger). Fig. 1 shows the average analog traces from strain-gauges during instructed flexion movements for Monkey K. A detailed description of the behavioral task can be found in [10].

Well-isolated single units with task-related activity were recorded sequentially in the M1 hand area (anterior bank of the central sulcus), contralateral to the trained hand (monkey C,  $n=49$ ; monkey K,  $n = 115$ ). Up to 15 trials per movement type were recorded during daily 2-to-3-hour recording sessions. Simultaneously recorded activity of multiple single units was simulated by aligning the activity of each unit at the time of switch closure.

### B. Continuous Decoding of Finger and Wrist Kinematics

Three different models were used to decode the kinematics of each finger and the wrist from the same population of M1 neurons. All models were trained using the neuronal firing rate over a 100 ms window shifting every 20 ms. Mutually exclusive trials were used for training (100 sets), validation (50 sets), and testing (100 sets).

**1) Linear Regression Filter**—The first approach used a linear filter to represent each decoded parameter as a weighted sum of the firing rate [2],

$$y_k = \sum_{i=1}^N w_i x_i + b_k \quad (1)$$

where  $y_k$  is the kinematic parameter recorded from the strain gauge for digit  $k$ ,  $x_i$  is the firing rate of neuron  $i$ ,  $w_i$  is the weight for the neuron,  $N$  is the total number of neurons, and  $b_k$  is a bias term.

Multiple linear filters were used to *simultaneously* extract the kinematics of each digit and the wrist. The system of equations was set up as follows:

$$Y = WX \quad (2)$$

where  $Y$  is the matrix of kinematic parameters recorded from the strain gauges,  $X$  is the matrix of neuronal firing rates, and  $W$  are the weights of the model. The bias term was calculated by appending a row of ones to matrix  $X$ . The optimal weights were calculated using the least squares solution,

$$W = (X^T X)^{-1} X^T Y \quad (3)$$

**2) Feedforward ANN**—The second approach was to use a multilayer, feedforward Artificial Neural Network (ANN), which have been widely used in non-linear regression, function approximation, and classification [11]. As before, the relationship between finger position and neural activity can be modeled as,

$$y_k = \sum_{i=1}^N w_i g(x_i) + b_k \quad (4)$$

but where  $g(x)$  is a non-linear transformation of the firing rate activity. The ANN was designed with a single hidden layer containing 25 neurons with a tan sigmoidal transfer function. Dimensionality of the input space was reduced by performing Principle Component Analysis (PCA), and retaining those components that cumulatively contributed to >95% of the total variance. The networks were trained offline in MATLAB 7.4 (Mathworks Inc.), with the optimal weights calculated using the scaled conjugate gradient algorithm and early validation stop to prevent overfitting.

**3) Kalman Filter**—As opposed to the feedforward model, a Kalman filter [9] models the relationship between neural activity and finger and wrist position by using a probabilistic approach that incorporates prior events.

In the Kalman framework described in detail in [9], the position of each end-effector is modeled as the system state,  $Y$ , and the firing rate is modeled as the observation,  $X$ .

$$Y=[y_1 \dots y_k]^T \text{ where } y_k \text{ is strain gauge value for each digit}$$

$$X=[x_1 \dots x_N]^T \text{ where } x_n \text{ is firing rate of each neuron}$$

The Kalman model makes two important assumptions:

1. The observations are a linear function of the current system state,

$$X(t)=H(t)Y(t)+q(t) \quad (5)$$

where  $H(t)$  is the coefficient matrix and  $q(t) \sim N(0,Q)$ .

2. The current system state is a linear function of the previous system state,

$$Y(t+1)=A(t)Y(t)+w(t) \quad (6)$$

where  $A(t)$  is the coefficient matrix, and  $w(t) \sim N(0,W)$ .  $A$  and  $H$  were assumed to be constant and calculated offline using a least squares approach.

At each time step, we first calculate an *a priori* estimate  $\hat{Y}$  using Eq. 5 and calculate its error covariance matrix,  $P'$ ,

$$\hat{Y}'(t)=A\hat{Y}(t-1) \quad (7)$$

$$P'(t)=AP(t-1)A^T+W \quad (8)$$

We then calculate the Kalman gain,  $K$ , update the estimate with an *a posteriori* estimate using new measurement data and calculate the posterior error covariance matrix,  $P$ ,

$$K(t)=P'(t)H^T(HP'(t)H^T+Q)^{-1} \quad (9)$$

$$\hat{Y}(t)=\hat{Y}'(t)+K(t)(X(t)-H\hat{Y}'(t)) \quad (10)$$

$$P(t)=(I-K(t)H)P'(t) \quad (11)$$

The Kalman gain produces a state estimate that minimizes the mean squared error [9].

### III. Results

The sample decoding results in Fig. 3 from monkey K, demonstrate that the predicted output (red) was highly correlated with the actual hand kinematics (blue) using all three decoding

models. Each plot shows the simultaneous reconstructed kinematics for all five digits and the wrist during instructed flexion movements.

Two different performance metrics were used to evaluate the decoding accuracy: mean squared error (MSE) and Pearson correlation coefficients (R). Table I and Table II summarize the decoding results for monkey C ( $n = 49$ ) and monkey K ( $n = 115$ ) respectively. Fig. 4 compares the MSE and correlation coefficients for all three decoding approaches, and across each end effector. Fig. 4A and 4B show the results for monkey C, while Fig. 4D and 4E show the results for monkey K.

As can be seen, the feedforward ANN appears to have the best performance for both monkey C ( $R=0.76$ ;  $MSE=0.05$ ) and monkey K ( $R=0.86$ ;  $MSE=0.04$ ). Furthermore, for both monkeys the poorest performance appears with decoding of the wrist kinematics (two-way ANOVA,  $p<0.05$ ). This is consistent across all three decoding models (monkey C: linear,  $R=0.25$ ; ANN,  $R=0.54$ , Kalman,  $R=0.38$  – monkey K: linear,  $R=0.66$ ; ANN,  $R=0.77$ , Kalman,  $R=0.72$ ) which suggest that this movement parameter is especially difficult to decode given the available neuron population.

Fig. 4C (monkey C) and Fig. 4F (monkey K) show the average correlation coefficient (R) as a function of randomly selected subpopulations of neurons for the linear regression (blue), ANN (red), and Kalman (green) filters. The results were averaged across five random subsets of a given number of neurons. Both the feedforward ANN and Kalman filter perform statistically significantly better than the linear regression filter (two-way ANOVA,  $p<0.05$ ).

#### IV. Discussion and Conclusion

This work demonstrates that it is indeed possible to decipher the neural coding of individual finger and wrist kinematics, which paves the way for dexterous manipulation of a multi-fingered hand prosthetic hand. The results do indicate, however, a negative bias towards decoding of wrist movements for both monkeys. This is likely due to the fact that the recording location of electrodes may have been biased to the lateral M1 hand area, and thus did not include as many neurons coding for wrist movements [5]. Furthermore, mechanical stops on the manipulandum may have prevented complete flexion and extension of the wrist (note saturation of wrist measurement during 'fw' in Fig. 2).

Furthermore, it appears that for this particular task a nonlinear decoding filter may be appropriate. Although the feedforward ANN and Kalman filter show comparable correlation coefficients, the ANN has smaller MSE values-particularly for fewer neurons (data not shown). Given the dexterity of the motor movements, and the fact that we are tracking multiple end-effectors simultaneously, it is not surprising that a nonlinear filter is better able to decode finger and wrist kinematics.

It is important to note, however, that even a simpler linear regression algorithm still provides respectable decoding accuracy (see Fig. 4C,F), and thus would be an acceptable low-cost alternative for embedding in the hardware controller of a prosthetic arm.

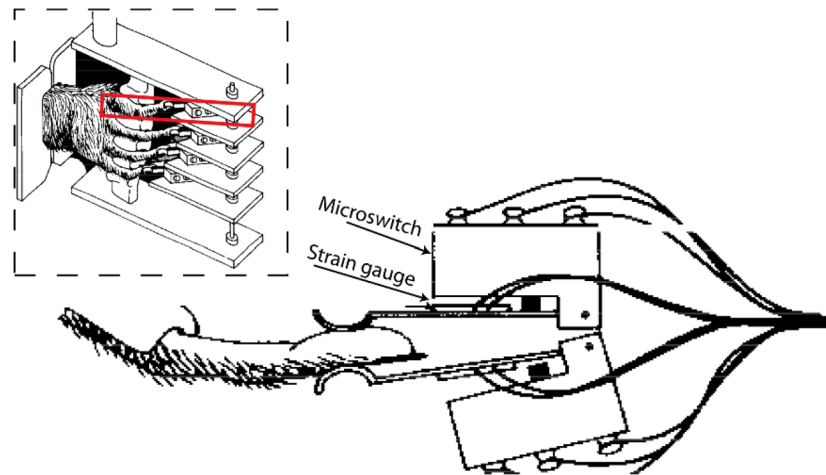
Future experiments aim to extend the techniques developed in this paper, in order to decode entire kinematics of the hand and arm as subjects perform less constrained movements. By demonstrating the decoding of individual finger primitives, these findings can be extended to more complex dexterous tasks involving multiple fingers and different grasp conformations of the hand.

## Acknowledgments

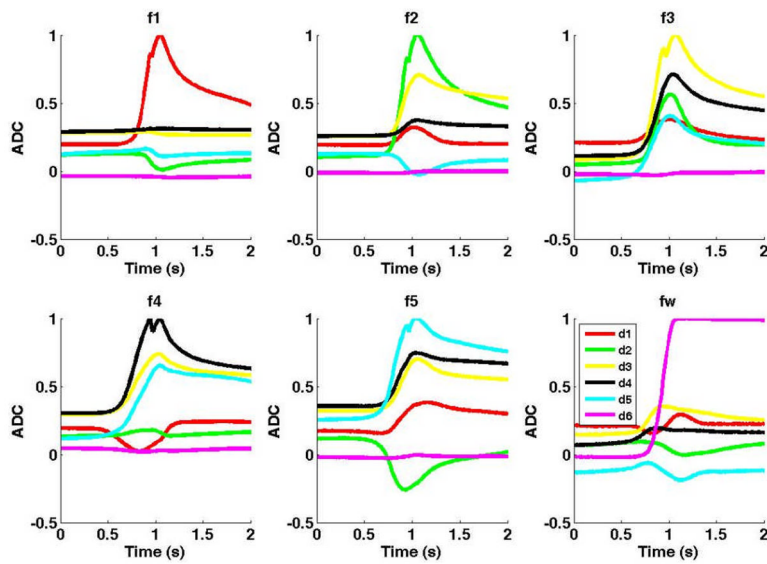
This work was sponsored in part by the Johns Hopkins University Applied Physics L under the DARPA Revolutionizing Prosthetics program, contract N66001-06-C-8005, by the National Science and Engineering Research Council of Canada (NSERC), and by NIH R01/R37 NS27686.

## References

1. Carmena JM, Lebedev MA, Crist RE, O'Doherty JE, Santucci DM, Dimitrov DF, Patil PG, Henriquez CS, Nicolelis MA. Learning to control a brain-machine interface for reaching and grasping by primates. *PLoS Biol* Nov;2003 1:E42. [PubMed: 14624244]
2. Wessberg J, Stambaugh CR, Kralik JD, Beck PD, Laubach M, Chapin JK, Kim J, Biggs SJ, Srinivasan MA, Nicolelis MA. Real-time prediction of hand trajectory by ensembles of cortical neurons in primates. *Nature* Nov;2000 408:361–5. [PubMed: 11099043]
3. Hochberg LR, et al. Neuronal ensemble control of prosthetic devices by a human with tetraplegia. *Nature* 2006;v442:164–71. [PubMed: 16838014]
4. Serruya MD, Hatsopoulos NG, Paninski L, Fellows MR, Donoghue JP. Instant neural control of a movement signal. *Nature* Mar 14;2002 416:141–2. [PubMed: 11894084]
5. Aggarwal V, Acharya S, Tenore F, Shin HC, Etienne-Cummings R, Schieber MH, Thakor NV. Asynchronous decoding of dexterous finger movements using M1 neurons. *IEEE Trans on Neural Sys and Rehab Eng* Feb;2008 16(1):3–14.
6. Acharya S, Tenore F, Aggarwal V, Etienne-Cummings R, Schieber MH, Thakor NV. Decoding finger movements using volume-constrained neuronal ensembles in M1. *IEEE Trans on Neural Sys and Rehab Eng* Feb;2008 16(1):15–23.
7. Georgopoulos AP, Schwartz AB, Kettner RE. Neuronal population coding of movement direction. *Science* Sep 26;1986 233:1416–9. [PubMed: 3749885]
8. Wu W, Shaikhouni A, Donoghue J, Black M. Closed-loop neural control of cursor motion using a Kalman filter. *Conf Proc IEEE Eng Med Biol Soc* 2004;6:4126–9. [PubMed: 17271209]
9. Wu, W.; Black, MJ.; Gao, Y.; Bienenstock, E.; Serruya, M.; Shaikhouni, A.; Donoghue, JP. *Advances in Neural Information Processing Systems* 15. MIT Press; 2003. Neural decoding of cursor motion using a Kalman filter; p. 133.140
10. Schieber MH. Individuated finger movements of rhesus monkeys: a means of quantifying the independence of the digits. *J Neurophysiol* Jun;1991 65:1381–91. [PubMed: 1875247]
11. Bishop, C. *Neural Networks for Pattern Recognition*. Oxford University Press; 1995.



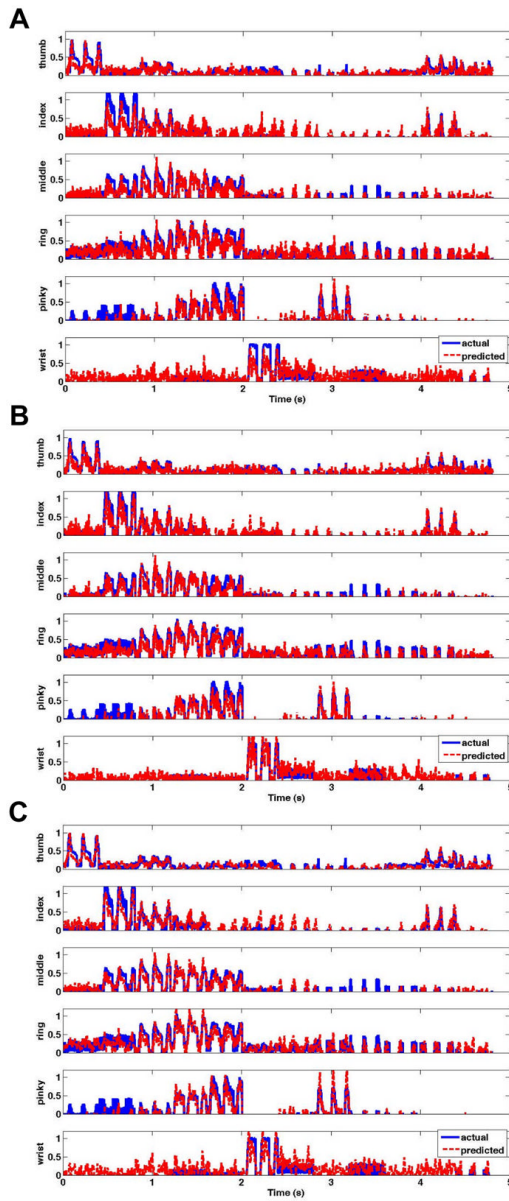
**Figure 1.** Illustration of strain gauges mounted at microswitches at the tip of each finger of the manipulandum (from [10]).



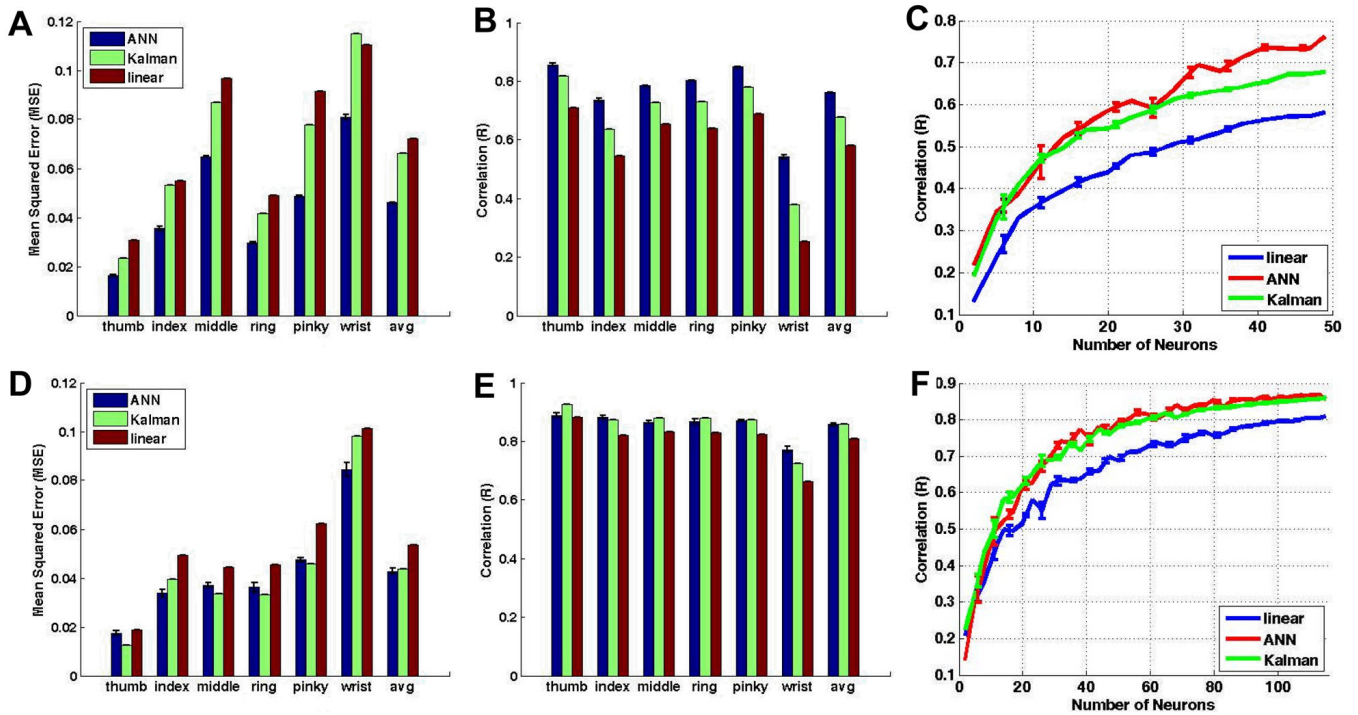
**Figure 2.**

Average analog traces from the strain gauges during instructed flexion movements (e.g. top left plot shows simultaneous traces of all five fingers and wrist during an instructed flexion of the thumb). Similar traces were obtained for instructed extension movements.





**Figure 3.** Reconstruction for the position of each end-effector using a) linear model, b) feedforward ANN, and c) Kalman filter. The predicted kinematic output (red) for each digit and the wrist was highly correlated with the actual hand kinematics (blue) using all three decoding models. Results are shown for monkey K ( $n = 115$ ).



**Figure 4.** Mean square errors and correlation coefficients for monkey C (top row) and monkey K (bottom row). All movements are decoded with high accuracy, although all three decoding models performed worst with wrist movements. Plots of the average correlation coefficient as a function of number of neurons (right column) show that the linear filter (blue) performed worse than both the ANN (red) and Kalman filter (green).

**TABLE I**

Summary of Decoding Results for Monkey C (n=49)

|         | Linear |      | ANN  |      | Kalman |      |
|---------|--------|------|------|------|--------|------|
|         | R      | MSE  | R    | MSE  | R      | MSE  |
| thumb   | 0.71   | 0.03 | 0.86 | 0.02 | 0.82   | 0.02 |
| index   | 0.55   | 0.05 | 0.74 | 0.04 | 0.64   | 0.05 |
| middle  | 0.65   | 0.10 | 0.78 | 0.06 | 0.73   | 0.09 |
| ring    | 0.64   | 0.05 | 0.80 | 0.03 | 0.73   | 0.04 |
| pinky   | 0.69   | 0.09 | 0.85 | 0.05 | 0.78   | 0.08 |
| wrist   | 0.25   | 0.11 | 0.54 | 0.08 | 0.38   | 0.11 |
| average | 0.58   | 0.07 | 0.76 | 0.05 | 0.68   | 0.07 |

**TABLE II**

Summary of Decoding Results for Monkey K (n=115)

|         | Linear |      | ANN  |      | Kalman |      |
|---------|--------|------|------|------|--------|------|
|         | R      | MSE  | R    | MSE  | R      | MSE  |
| thumb   | 0.88   | 0.02 | 0.89 | 0.02 | 0.93   | 0.01 |
| index   | 0.82   | 0.05 | 0.88 | 0.03 | 0.87   | 0.04 |
| middle  | 0.83   | 0.04 | 0.86 | 0.04 | 0.88   | 0.03 |
| ring    | 0.83   | 0.05 | 0.87 | 0.04 | 0.88   | 0.03 |
| pinky   | 0.82   | 0.06 | 0.87 | 0.05 | 0.87   | 0.05 |
| wrist   | 0.66   | 0.10 | 0.77 | 0.08 | 0.72   | 0.10 |
| average | 0.81   | 0.05 | 0.86 | 0.04 | 0.86   | 0.04 |