Neural Decoding of Cursor Motion using Kalman Filter

W. Wu, M. J. Black, Y. Gao, E. Bienenstock,
M. Serruya, A. Shaikhouni, J. P. Donoghue

NIPS 15, 2003

CSE 599E: Brain-Computer Interfaces
Presented by: Jean Wu
19 April 2006

Overview

- Direct neural control of external devices requires the accurate decoding of neural activity representing continuous movement
- Develop a control-theoretic approach that models the probabilistic relationship between hand motions and neural firing rates
- Kalman filter to encode/decode the neural data

Overview

Using a mathematical decoding method to produce an estimate of the system “state” from a sequence of “observations”

“State” – hand movement
(position, velocity, and acceleration)

“Observation” – measurement of the neural firing rates

Overview

Decoding method should:

☺ Have good probabilistic foundation

☺ Model noise in the data explicitly

☺ Indicate uncertainty in state estimations

☺ Make minimal assumptions about the data

Overview

Decoding method should:

☺ Require minimal ‘training’ data

☺ On-line estimation with short delay (< 200*ms*)

☺ Provide insight into the neural coding of movement

à Kalman Filtering Method

Experimental Setup

- A 100-microelectrode array implanted in the arm area of primary motor cortex of a monkey

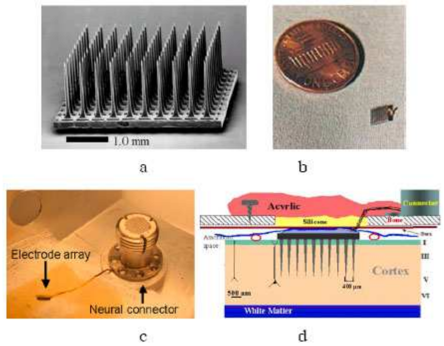
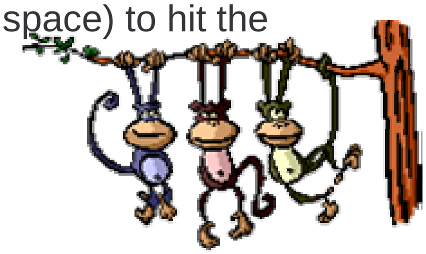


Fig: http://donoghue.neuro.brown.edu/pubs/capri%20IEEE%20review.pdf

Experimental Setup

- The monkey views a computer screen while gripping a two-link manipulandum that controls 2D motion of a cursor on the screen
- **Task:** move the manipulandum on a 30x30cm^2 tablet (20x20cm^2 working space) to hit the randomly placed targets



Experiment Setup

- Record the trajectory of the hand and neural activity of 42 cells simultaneously
- Firing rate 70*ms*
- Assume the observation (firing rate) is a linear function of the state + Gaussian noise*

Fixed Linear Filter

$$x_k = a + \sum_v \sum_{j=0}^{N} r_{k-j}^v f_j^v$$

Compute hand position as a linear combination of neural firing rates over some fixed time period

Fixed Linear Filter

$$x_k = a + \sum_v \sum_{j=0}^{N} r_{k-j}^v f_j^v$$

x_k : x-position at time $t_k = k\Delta t$ (Δt = 70ms),
$k = 1, \ldots, M$ and M is the number of time steps in a trial

a : constant offset

r_{k-j}^v : firing rate of neuron v at time t_{k-j}

f_j^v : filter coefficients (learn from training data using least-square)

Kalman Filter *(Encoding)*

Generative model of neural firing

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{q}_k$$

H = a matrix that linearly relates hand state to neural firing

Assume the noise in observations is zero mean and normally distributed

! Current state linearly causes the observed firing rate

Kalman Filter *(Encoding)*

Generative model of neural firing

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

A = coefficient matrix

! the state at time *k*+1 is linearly related to the state at time *k*

Kalman Filter *(Decoding)*

Discrete time update equation:

$$\hat{\mathbf{x}}_k^- = \mathbf{A}\hat{\mathbf{x}}_{k-1},$$
$$\mathbf{P}_k^- = \mathbf{A}\mathbf{P}_{k-1}\mathbf{A}^T + \mathbf{W}$$

6 Prediction of the *a priori* state estimate

6 obtain the estimate at time t_k from time t_{k-1} then compute its error covariance matrix P_k^-

Kalman Filter *(Decoding)*

Measurement update equation:

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_k^-),$$
$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{H})\mathbf{P}_k^-,$$

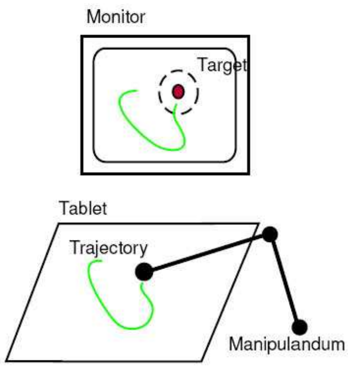
B Update the estimate with new measurement data to produce *a posteriori* state estimate

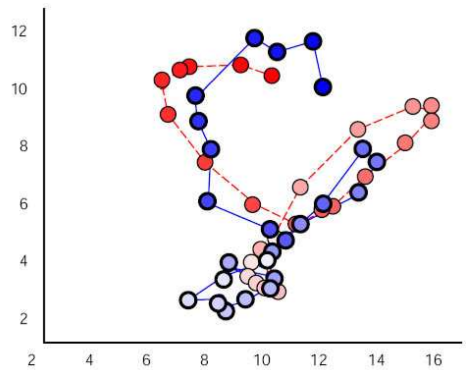
B $\mathbf{P}_k$ = state error covariance after taking into account the neural data

B $\mathbf{K}_k$ = Kalman *gain* matrix

Experiment

Reconstructing 2D Hand Motion





Training Decoding

Results

- ~3.5min of training data (same as linear filtering method)
- Results use ~1min test data
- **Optimal Lag** ~140*ms* (two time steps)

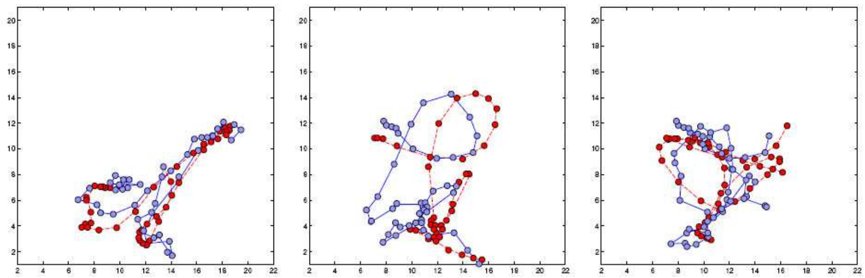
Results

Reconstruction Results

Method	Correlation Coefficient (x, y)	MSE (cm^2)
Kalman ($0ms$ lag)	(0.768, 0.912)	7.09
Kalman ($70ms$ lag)	(0.785, 0.932)	7.07
Kalman (140ms lag)	**(0.815, 0.929)**	**6.28**
Kalman ($210ms$ lag)	(0.808, 0.891)	6.87
Kalman (no acceleration)	(0.817, 0.914)	6.60
Linear filter	(0.756, 0.915)	8.30

Results

Reconstructed Trajectories



Red: true target trajectory

Blue: reconstruction using Kalman filter

Comparison with linear filtering

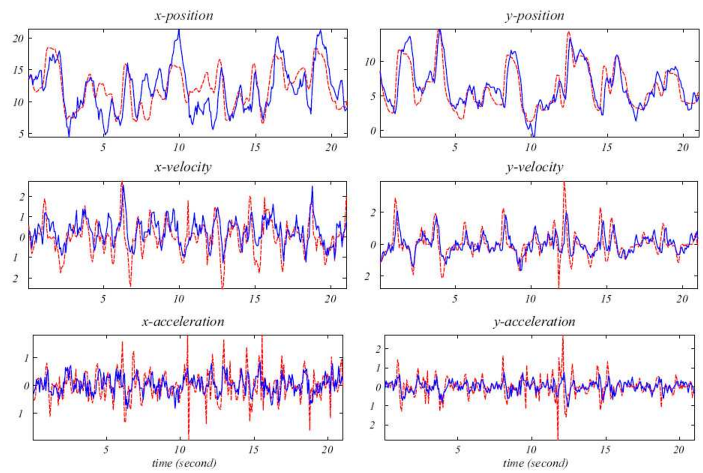
Linear filter:

D not benefit from use of time-lagged data

D not explicitly reconstruct velocity or acceleration

C **Kalman filter** gives higher correlation coefficient and lower mean-squared error à more accurate reconstruction

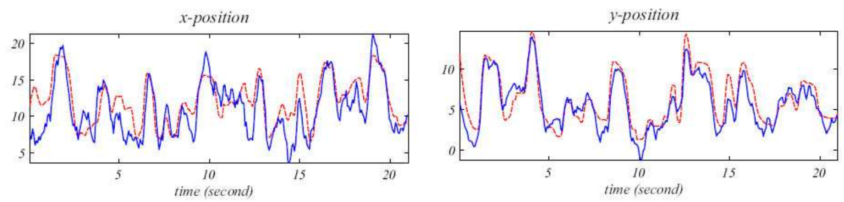
Reconstruction of Position using Kalman Filter



Red: true target trajectory

Blue: reconstruction using Kalman filter

Reconstruction of Position using Linear Filter



Red: true target trajectory

Blue: reconstruction using linear filter

Conclusions

The Kalman filter model can be easily learned using a few min of training data and provides real-time estimates of hand position every 70*ms* given the firing rates of 42 cells in primary motor cortex

Conclusions

The estimated trajectories are more accurate than the fixed linear filtering results

The Kalman filter provides a rigorous probabilistic approach with well understood theory.

the End