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

Kalman Filtering Method

Experimental Setup

1. 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

1. The monkey views a computer screen while gripping a two-link manipulandum that controls 2D motion of a cursor on the screen.

2. **Task**: move the manipulandum on a 30x30 cm² tablet (20x20 cm² working space) to hit the randomly placed targets.

Experiment Setup

1. Record the trajectory of the hand and neural activity of 42 cells simultaneously.

2. Firing rate 70 ms

3. Assume the observation (firing rate) is a linear function of the state + Gaussian noise.
Fixed Linear Filter

\[ x_k = a + \sum_{\nu} \sum_{j=0}^{N} r_{k-j}^{\nu} f_{j}^{\nu} \]

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

Fixed Linear Filter

\[ x_k = a + \sum_{\nu} \sum_{j=0}^{N} r_{k-j}^{\nu} f_{j}^{\nu} \]

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

\( a \): constant offset

\( r_{k-j}^{\nu} \): firing rate of neuron \( \nu \) at time \( t_{k-j} \)

\( f_{j}^{\nu} \): filter coefficients (learn from training data using least-square)
**Kalman Filter (Encoding)**

**Generative model of neural firing**

\[ z_k = H_k x_k + q_k \]

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

Assume the noise in observations is zero mean and normally distributed

\[ \text{Current state linearly causes the observed firing rate} \]

\[ x_{k+1} = A_k x_k + w_k \]

\( A \) = coefficient matrix

\[ \text{the state at time } k+1 \text{ is linearly related to the state at time } k \]
Kalman Filter (Decoding)

Discrete time update equation:

\[
\hat{x}_k^- = A\hat{x}_{k-1},
\]
\[
P_k^- = AP_{k-1}A^T + W.
\]

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:

\[
\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-),
\]
\[
P_k = (I - K_kH)P_k^-,
\]

- Update the estimate with new measurement data to produce a posteriori state estimate
- \( P_k \) = state error covariance after taking into account the neural data
- \( K_k \) = Kalman gain matrix
Experiment
Reconstructing 2D Hand Motion

Results

• ~3.5min of training data (same as linear filtering method)

• Results use ~1min test data

• Optimal Lag ~140ms (two time steps)
Results

Reconstruction Results

<table>
<thead>
<tr>
<th>Method</th>
<th>Correlation Coefficient ((x, y))</th>
<th>MSE ((cm^2))</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kalman (0(ms) lag)</td>
<td>((0.768, 0.912))</td>
<td>7.09</td>
</tr>
<tr>
<td>Kalman (70(ms) lag)</td>
<td>((0.785, 0.932))</td>
<td>7.07</td>
</tr>
<tr>
<td><strong>Kalman (140(ms) lag)</strong></td>
<td><strong>((0.815, 0.929))</strong></td>
<td><strong>6.28</strong></td>
</tr>
<tr>
<td>Kalman (210(ms) lag)</td>
<td>((0.808, 0.891))</td>
<td>6.87</td>
</tr>
<tr>
<td>Kalman (no acceleration)</td>
<td>((0.817, 0.914))</td>
<td>6.60</td>
</tr>
<tr>
<td>Linear filter</td>
<td>((0.756, 0.915))</td>
<td>8.30</td>
</tr>
</tbody>
</table>

Results

Reconstructed Trajectories

**Red:** true target trajectory

**Blue:** reconstruction using Kalman filter
Comparison with linear filtering

**Linear filter:**
- not benefit from use of time-lagged data
- not explicitly reconstruct velocity or acceleration

- **Kalman filter** gives higher correlation coefficient and lower mean-squared error
  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 70ms 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