A Switching Kalman Filter Model for the Motor Cortical Coding of Hand Motion

Wei Wu[†] Michael J. Black[‡] David Mumford[†] Yun Gao[†] Elie Bienenstock[†*] John P. Donoghue[*]

[†]Division of Applied Mathematics [‡]Dept. of Computer Science [*]Dept. of Neuroscience
Brown University, Providence, RI 02912

weiwu@cfm.brown.edu, black@cs.brown.edu, david_mumford@brown.edu,
gao@cfm.brown.edu, elie@dam.brown.edu, john_donoghue@brown.edu

***Abstract*— We present a Switching Kalman Filter Model (SKFM) for the real-time inference of hand kinematics from a population of motor cortical neurons. First we model the probability of the firing rates of the population at a particular time instant as a Gaussian mixture where the mean of each Gaussian is some linear function of the hand kinematics. This mixture contains a "hidden state", or weight, that assigns a probability to each linear, Gaussian, term in the mixture. We then model the evolution of this hidden state over time as a Markov chain. The Expectation-Maximization (EM) algorithm is used to fit this mixture model to training data that consists of measured hand kinematics (position, velocity, acceleration) and the firing rates of 42 units recorded with a chronically implanted multi-electrode array. Decoding of neural data from a separate test set is achieved using the Switching Kalman Filter (SKF) algorithm. Quantitative results show that the SKFM outperforms the traditional linear Gaussian model in the decoding of hand movement. These results suggest that the SKFM provides a real-time decoding algorithm that may be appropriate for neural prosthesis applications.**

I. Introduction

Recent research on neural prostheses has explored a variety of neural decoding methods that convert neural activity into a voluntary control signal [2], [6], [7], [8]. Recently, we proposed a control-theoretic Kalman filter model [9], in which hand movement is encoded by a population of cells with a linear Gaussian model and is decoded using the Kalman filter algorithm. Our results suggest that this simple Kalman filter model enables accurate and efficient decoding of continuous hand motion. The method is based on an approximate generative model of neural firing. In particular, it assumes that the observed firing rates are a linear function of hand kinematics (position, velocity, and acceleration) and that they are corrupted by Gaussian noise. This generative model is only a rough approximation and we seek to systematically extend the linear Gaussian model to non-linear and/or non-Gaussian models and evaluate their performance with respect to neural decoding. Unfortunately, these non-linear models are difficult to learn from training data and the associated decoding methods are computationally expensive [1].

This work was supported in part by: the DARPA Brain Machine Interface Program, NINDS Neural Prosthetics Program and Grant #NS25074, and the National Science Foundation (ITR Program award #0113679).

In this paper, we exploit a mixture of linear Gaussian models that provides a general probabilistic model relating neural activity to hand kinematics. The key insight is that, while such a model is more general than the simple linear Gaussian model, it admits an efficient, real-time, decoding algorithm. This mixture model is called a Switching Kalman Filter Model (SKFM) [4] and the parameters of the model can be learned from training data using the Expectation-Maximization (EM) algorithm. Decoding is achieved using the Switching Kalman Filter algorithm [4] which has computational efficiency similar to the Kalman Filter and provides real-time decoding. Quantitative results show that the SKFM outperforms the Kalman filter in the decoding of hand movement for the neural data recorded from an implanted microelectrode array. The method satisfies the goals of accurate decoding and real-time performance which are both necessary for direct neural control tasks [6].

II. Data Acquisition and Processing

Task: Simultaneous recordings are acquired from an array consisting of 100 microelectrodes chronically implanted in the arm area of primary motor cortex (MI) of a Macaque monkey. The monkey views a computer monitor while gripping a two-link manipulandum that controls the 2D motion of a cursor on the monitor [6]. We use the experimental paradigm of [6], in which a target dot appears in a random location on the monitor and the task requires moving a feedback dot with the manipulandum so that it hits the target. When the target is hit, it randomly jumps to a new location. Note that the hand motions in this task are more "general" and natural than those in the more common "center-out" tasks [7].

Data: The trajectory of the hand and the neural activity of 42 cells are recorded simultaneously. In particular, we compute the position, velocity, and acceleration of the hand every $70ms$. Neural data is recorded using a commercial Plexon system, units are isolated manually, and spikes are detected on-line using manually set thresholds. This "sorting" is approximate and the measured activity may include the activity of multiple cells. The activity of each unit is summed within non-overlapping $70ms$ time bins.

Pre-processing: Before fitting our model we apply a square-root transform to the firing data as suggested in [3]. The mean

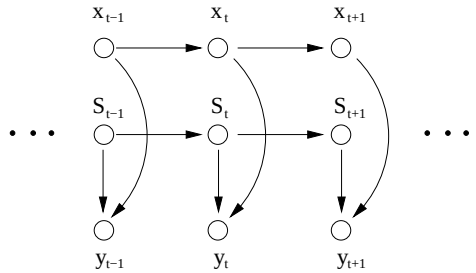

Fig. 1. Graphical model representation for SKFM: It is a mixture of State-space model and Hidden Markov model. Both states and switching labels are assumed Markov over time, and given states and labels, the observation is a linear Gaussian model.

firing rate for each unit is then subtracted to obtain zero-mean data.

In the work that follows we fit a Gaussian mixture model to the data in which each component of the model has a full covariance matrix (i.e. 42×42). Given the large number of units, correlations between their firing activity, and a limited amount of training data, fitting multiple covariance matrices can be computationally unstable. To deal with this, we reduce the dimensionality of the input firing rates using Principal Component Analysis (PCA). Here we project the firing rates onto a 39 dimensional subspace which results in a loss of less than 1% of the information. For simplicity, we still refer to these 39 principal components as "cells". This approach could be applied to larger populations to significantly compress the firing data making it feasible to fit full covariance matrices with limited training data.

III. Methods

In the Switching Kalman Filter Model, the hand movement (position, velocity and acceleration) is modeled as the system *state* and the neural firing rate is modeled as the *observation* (measurement). Let the *state* of the hand at the current instant in time be $\mathbf{x}_t = [x, y, v_x, v_y, a_x, a_y]_t^T \in \Re^6$, which represents x-position, y-position, x-velocity, y-velocity, x-acceleration, and y-acceleration at time $t\Delta t$ where $\Delta t = 70ms$ in our experiments. The observations $\mathbf{y}_t \in \Re^K$ which here represent a $K \times 1$ vector containing the firing rates at time t for K observed neurons within $70ms$.

Figure 1 shows the SKFM framework, where the joint probability distribution over states ($\{\mathbf{x}_t\}$), observations ($\{\mathbf{y}_t\}$) and switching variables ($\{S_t\}$) is

$$p(\{\mathbf{x}_t, \mathbf{y}_t, S_t\}) = [p(S_1)\prod_{t=2}^{T} p(S_t|S_{t-1})][p(\mathbf{x}_1)\prod_{t=2}^{T} p(\mathbf{x}_t|\mathbf{x}_{t-1})][\prod_{t=1}^{T} p(\mathbf{y}_t|\mathbf{x}_t, S_t)].$$

Conditioned on the hidden switching state, the probability of observing the firing rate vector is given by

$$p(\mathbf{y}_t|\mathbf{x}_t) = \sum_{j=1}^{N} p(S_t = j)p(\mathbf{y}_t|\mathbf{x}_t, S_t = j), \quad (1)$$

in which

$$p(\mathbf{y}_t|\mathbf{x}_t, S_t = j) = \mathcal{N}(\mathbf{H}_j\mathbf{x}_t, \mathbf{Q}_j), \quad (2)$$

where $j = 1, 2, \cdots, N, t = 1, 2, \cdots, T$. T is the total number of time steps in the trial and N is the number of different linear models in our mixture. $\mathcal{N}(\mathbf{H}_j\mathbf{x}_t, \mathbf{Q}_j)$ denotes a Gaussian distribution with mean $\mathbf{H}_j\mathbf{x}_t$ where $\mathbf{H}_j \in \Re^{K\times 6}$ is a matrix that linearly relates the hand state to the neural firing. The noise covariance matrix is $\mathbf{Q}_j \in \Re^{K\times K}$.

We assume the hidden states $S_1, S_2, \cdots, S_T$ form a first order Markov chain as illustrated in Figure 1; that is,

$$p(S_t = j) = \sum_{i=1}^{N} p(S_t = j|S_{t-1} = i)p(S_{t-1} = i), \quad (3)$$

where we denote

$$c_{ij} = p(S_t = j|S_{t-1} = i), \quad 1 \le i, j \le N. \quad (4)$$

We represent these state transition probabilities as a *transition matrix* $\mathbf{C} = \{c_{ij}\}$.

The kinematic state is also assumed to form a Markov chain represented by the system model:

$$p(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{A}\mathbf{x}_{t-1}, \mathbf{W}), \quad (5)$$

where $\mathbf{A} \in \Re^{6\times 6}$ is the coefficient matrix and the noise covariance matrix is $\mathbf{W} \in \Re^{6\times 6}$.

Encoding

In practice, we need to estimate all the parameters $\mathbf{A}, \mathbf{W}, \mathbf{H}_{1:N}, \mathbf{Q}_{1:N}, \mathbf{C}$ from training data, in which both hand kinematics $\{\mathbf{x}_t\}$ and firing rates $\{\mathbf{y}_t\}$ are known, but the switching labels $\{S_t\}$ are hidden. Therefore, we estimate all the parameters by maximizing likelihood $p(\{\mathbf{x}_t, \mathbf{y}_t\})$:

$$\begin{aligned}&\text{argmax}_{A,W,H_{1:N},Q_{1:N},C} p(\{\mathbf{x}_t, \mathbf{y}_t\}) \\ &= \text{argmax}_{A,W,H_{1:N},Q_{1:N},C} p(\{\mathbf{x}_t\})p(\{\mathbf{y}_t\}|\{\mathbf{x}_t\}) \\ &= \text{argmax}_{A,W} p(\{\mathbf{x}_t\})\text{argmax}_{H_{1:N},Q_{1:N},C} p(\{\mathbf{y}_t\}|\{\mathbf{x}_t\})\end{aligned}$$

Using the linear Gaussian property of $p(\{x_t\})$, we have

$$\text{argmax}_{A,W} p(\{\mathbf{x}_t\}) =$$

$$\text{argmin}_{A,W} \sum_{t=2}^{T} [\log(\det \mathbf{W}) + (\mathbf{x}_t - \mathbf{A}\mathbf{x}_{t-1})^T \mathbf{W}^{-1}(\mathbf{x}_t - \mathbf{A}\mathbf{x}_{t-1})].$$

The above minimization has a closed form solution:

$$\begin{aligned}\mathbf{A} &= \sum_{t=2}^{T} \mathbf{x}_t\mathbf{x}_{t-1}^T \left(\sum_{t=2}^{T} \mathbf{x}_{t-1}\mathbf{x}_{t-1}^T\right)^{-1}, \\ \mathbf{W} &= \frac{1}{T-1}\left(\sum_{t=2}^{T} \mathbf{x}_t\mathbf{x}_t^T - \mathbf{A}\sum_{t=2}^{T} \mathbf{x}_{t-1}\mathbf{x}_t^T\right).\end{aligned}$$

The other term $p(\{\mathbf{y}_t\}|\{\mathbf{x}_t\}) = \sum_{\{S_t\}} p(\{\mathbf{y}_t, S_t\}|\{\mathbf{x}_t\})$ contains hidden variables $\{S_t\}$. While no closed form solution exists, the EM algorithm offers an effective way to estimate all the parameters. Denote $\theta = (\mathbf{H}_{1:N}, \mathbf{Q}_{1:N}, \mathbf{C})$ and $p(\cdot|\cdots) = p(\cdot|\{\mathbf{x}_t, \mathbf{y}_t\}; \theta_k)$, we update θ_k to θ_{k+1} as

$$\text{argmax}_\theta E_{p(\{S_t\}|\cdots)} \log p(\{\mathbf{y}_t, S_t\}|\{\mathbf{x}_t\}; \theta).$$

The detail of the maximization process can be found in [4]. We only show the updating result here:

$$
\begin{aligned}
c_{ij} &= \sum_{t=2}^{T} p(S_t = j, S_{t-1} = i|\cdots) / \sum_{t=2}^{T} p(S_{t-1} = i|\cdots), \\
\mathbf{H}_j &= [\sum_{t=1}^{T} p(S_t = j|\cdots)\mathbf{y}_t\mathbf{x}_t^T][\sum_{t=1}^{T} p(S_t = j|\cdots)\mathbf{x}_t\mathbf{x}_t^T]^{-1}, \\
\mathbf{Q}_j &= \sum_{t=1}^{T} [p(S_t = j|\cdots)(\mathbf{y}_t\mathbf{y}_t^T - H_j\mathbf{x}_t\mathbf{y}_t^T)] / \sum_{t=1}^{T} p(S_t = j|\cdots).
\end{aligned}
$$

where $i, j = 1, \cdots, N$ and the conditional probabilities of S_t, S_{t-1} can be calculated using standard Dynamic Programming techniques.

Experimentally we find that approximately 3.5 minutes of training data suffices for accurate reconstruction (this is similar to the result for fixed linear filters reported in [6]). Training the model takes approximately 1 minute on a Pentium III 866.

Decoding (Estimation)

Given the probabilistic encoding model defined above, we turn to the problem of decoding; that is, reconstructing hand motion from the firing rates of the cells. Let $\mathbf{x}_{1:t}$ denote $\mathbf{x}_1, \cdots, \mathbf{x}_t$, and the same for $\mathbf{y}_{1:t}$ and $S_{1:t}$. We seek the *a posteriori* mean $\hat{\mathbf{x}}_t = \mathrm{E}(\mathbf{x}_t|\,\mathbf{y}_{1:t})$ that minimizes the mean square error $\mathrm{E}((\mathbf{x}_t - \hat{\mathbf{x}}_t)^2|\,\mathbf{y}_{1:t})$. We achieve this using the efficient Switching Kalman Filter algorithm which is briefly described here (see [4] for details).

Under the SKFM framework, the posterior distribution of the state is also a mixture of Gaussians, but the mixture number grows exponentially with time, i.e. assume initial $p(\mathbf{x}_1|\,\mathbf{y}_1)$ is a mixture of N Gaussians (one for each value of S_1), then $p(\mathbf{x}_t|\,\mathbf{y}_{1:t})$ is a mixture of N^t Gaussians (one for each sequence of $S_1, \cdots, S_t$). The Switching Kalman Filter (SKF) algorithm [4] approximates these N^t Gaussians with a mixture of N Gaussians at each time step t. The fixed number N over time is maintained by "collapsing" N Gaussians into one using moment matching, which can be shown to be the optimal approximation under the criterion of minimization of relative entropy between the Gaussians.

We need the following notation:

$$
\begin{aligned}
\mathbf{x}_t^j &= \mathrm{E}[\mathbf{x}_t|\,\mathbf{y}_{1:t}, S_t = j], \\
\mathbf{V}_t^j &= \mathrm{Cov}[\mathbf{x}_t|\,\mathbf{y}_{1:t}, S_t = j], \\
\mathbf{x}_t^{ij} &= \mathrm{E}[\mathbf{x}_t|\,\mathbf{y}_{1:t}, S_t = j, S_{t-1} = i], \\
\mathbf{V}_t^{ij} &= \mathrm{Cov}[\mathbf{x}_t|\,\mathbf{y}_{1:t}, S_t = j, S_{t-1} = i], \\
w_t^j &= p(S_t = j|\,\mathbf{y}_{1:t}), \\
w_t^{ij} &= p(S_t = j, S_{t-1} = i|\,\mathbf{y}_{1:t}), \\
g_t^{ij} &= p(S_{t-1} = i|\,\mathbf{y}_{1:t}, S_t = j), \\
l_t^{ij} &= p(\mathbf{y}_t|\,\mathbf{y}_{1:t-1}, S_t = j, S_{t-1} = i).
\end{aligned}
$$

The decoding algorithm that follows shows how the posterior distribution $p(\mathbf{x}_t|\,\mathbf{y}_{1:t})$ is approximated by a mixture of N Gaussians $\sum_j w_t^j \mathcal{N}(\mathbf{x}_t^j, \mathbf{V}_t^j)$ for each time step t:

TABLE I

RECONSTRUCTION ACCURACY USING KALMAN FILTER AND SKF

Method	Corr-Coef (x, y)	MSE (cm^2)
Kalman	(0.82, 0.93)	5.87
SKF	(0.84, 0.93)	5.39

From time step $t-1$ to t:

$$
\begin{aligned}
[\mathbf{x}_t^{ij}, \mathbf{V}_t^{ij}, l_t^{ij}] &= \textbf{filter}(\mathbf{x}_{t-1}^i, \mathbf{V}_{t-1}^i, \mathbf{y}_t, \mathbf{H}_j, \mathbf{Q}_j, \mathbf{A}, \mathbf{W}) \\
w_t^{ij} &= l_t^{ij} c_{ij} w_{t-1}^i / \sum_{ij} l_t^{ij} c_{ij} w_{t-1}^i \\
w_t^j &= \sum_i w_t^{ij} \\
g_t^{ij} &= w_t^{ij} / w_t^j \\
[\mathbf{x}_t^j, \mathbf{V}_t^j] &= \textbf{collapse}(\{\mathbf{x}_t^{ij}, \mathbf{V}_t^{ij}, g_t^{ij}\}_i)
\end{aligned}
$$

where we use the standard Kalman Filter subroutine **filter** (shown in the appendix). Assume at time $t-1$, the posterior distribution $p(\mathbf{x}_{t-1}|\,\mathbf{y}_{1:t-1})$ is a mixture of N Gaussians (which is true at first time step), then **filter** propagates them to time t to have a mixture of N^2 Gaussians (one for each array of (S_{t-1}, S_t)), i.e.

$$
p(\mathbf{x}_t|\,\mathbf{y}_{1:t}) = \sum_j w_t^j p(\mathbf{x}_t|\,\mathbf{y}_{1:t}, S_t = j) = \sum_{ij} w_t^{ij} \mathcal{N}(\mathbf{x}_t^{ij}, \mathbf{V}_t^{ij}).
$$

For each $j \in \{1, \cdots, N\}, p(\mathbf{x}_t|\,\mathbf{y}_{1:t}, S_t = j) = \sum_i g_t^{ij} \mathcal{N}(\mathbf{x}_t^{ij}, \mathbf{V}_t^{ij})$ is a mixture of N Gaussians. The subroutine **collapse** (also shown in the appendix) approximates this mixture by one Gaussian by matching the mean and covariance, which produces a mixture with N components at time step t. Therefore, the posterior distribution $p(\mathbf{x}_t|\,\mathbf{y}_{1:t}) \approx \sum_j w_t^j \mathcal{N}(\mathbf{x}_t^j, \mathbf{V}_t^j)$, and the final state estimation $\hat{\mathbf{x}}_t$ and its error covariance $\hat{\mathbf{V}}_t$ are as follows:

$$
\begin{aligned}
\hat{\mathbf{x}}_t &= \sum_j w_t^j \mathbf{x}_t^j, \\
\hat{\mathbf{V}}_t &= \sum_j w_t^j (\mathbf{V}_t^j + (\mathbf{x}_t^j - \hat{\mathbf{x}}_t)(\mathbf{x}_t^j - \hat{\mathbf{x}}_t)^T).
\end{aligned}
$$

At the beginning of the test trial we let the predicted initial condition equal the average state in training data, then the SKF algorithm is applied over time. Table I shows that, the SKF gives a more accurate reconstruction than the Kalman Filter (which uses a single linear Gaussian model).

Figure 2 shows the SKF reconstruction of the first 20 seconds of test data (distinct from the training data) for each component of the state variable (position, velocity and acceleration in x and y). We see that the reconstructed trajectories are smooth and visually similar to the true ones (i.e. high Correlation Coefficient). We also observe that the SKF produces a more accurate reconstruction in terms of the Mean Square Error (MSE).

For the SKF, the posterior distribution of the state is assumed to be a mixture of Gaussians and the uncertainty can be estimated by the error covariance. The 95% confidence

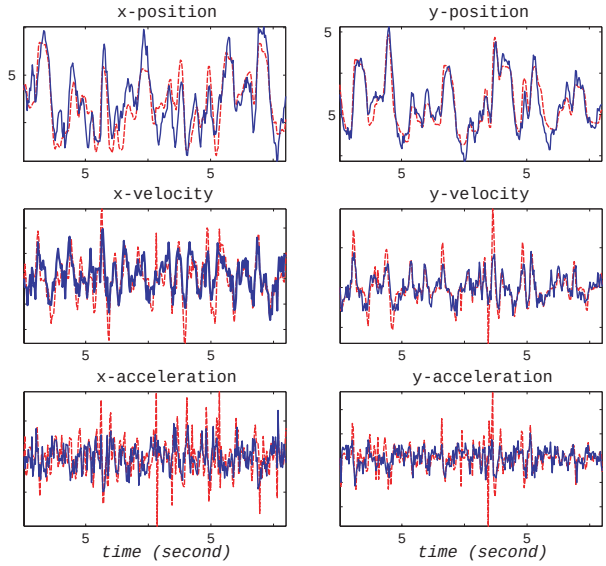

Fig. 2. Reconstruction of each component of the system state variable: true hand motion (dashed (red)) and reconstruction using the SKF (solid (blue)). $20seconds$ from a 1*minute* test sequence are shown.

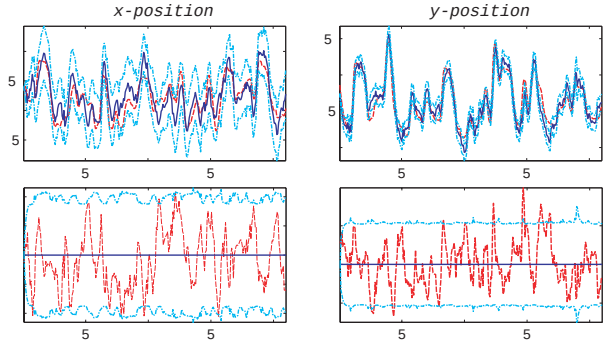

Fig. 3. Confidence estimation for the x and $y-position$: The first row shows the true (dashed (red)), reconstructed trajectories (solid (blue)) and their 95% confidence range (dashdot (cyan)). The second row is the normalized version by subtracting the corresponding reconstruction, which shows more clearly the confidence intervals.

interval is shown for both x and y-position in Figure 3. We see that most of time the true positions are within of them, which shows the validity of the confidence.

IV. Conclusions

Based on previous Kalman filter work, we proposed a natural non-linear extension which is more appropriate for the neural control of 2D cursor motion. The new approach is focused on the observation model, which can be efficiently learned by the EM algorithm using a few minutes of training data and provides real-time estimates of hand position every $70ms$ given the firing rates of 42 cells in primary motor cortex. The estimated trajectories are more accurate than the standard linear Kalman filter results for this data set. The SKFM has many of the desirable properties of the Kalman filter (e.g. linear Gaussian models (when conditioned on the switching state), full covariance model, efficient encoding and decoding), while being more versatile and accurate. It can deal with violations (to some extent) of both the assumption of linearity and Gaussian noise.

Finally, our future work will evaluate its performance for on-line neural control of cursor motion and compare with Kalman filter and other linear regression methods. Additionally, we are exploring alternative measurement noise models, non-linear system models, adaptive learning techniques, and non-linear particle filter decoding methods.

Acknowledgment

We thank M. Serruya, A. Shaikhouni, J. Dushanova, C. Vargas, L. Lennox, and M. Fellows for their assistance.

References

[1] Gao, Y., Black, M. J., Bienenstock, E., Wu, W., Donoghue, J. P., "A quantitative comparison of linear and non-linear models of motor cortical activity for the encoding and decoding of arm motions," *1st International IEEE/EMBS Conference on Neural Engineering*, pp. 189–192, March 20-22, 2003.

[2] Helms Tillery, S., Taylor, D., Isaacs, R., Schwartz, A. (2000) Online control of a prosthetic arm from motor cortical signals. *Soc. for Neuroscience Abst.*, Vol. 26.

[3] Moran, D. and Schwartz, B. (1999). Motor cortical representation of speed and direction during reaching. *J. of Neurophysiology*, 82(5):2676–2692.

[4] Murphy, K. P. (1998). Switching Kalman Filter. Technical Report 98-10, Compaq Cambridge Research Laboratory.

[5] Paninski, L., Fellows, M., Hatsopoulos, N., and Donoghue, J. P. (2001). Temporal tuning properties for hand position and velocity in motor cortical neurons. *submitted, J. of Neurophysiology.*

[6] Serruya, M. D., Hatsopoulos, N. G., Paninski, L., Fellows, M. R., and Donoghue, J. P. (2002). Brain-machine interface: Instant neural control of a movement signal. *Nature*, (416):141–142.

[7] Taylor. D., Tillery, S., Schwartz, A. (2002). Direct cortical control of 3D neuroprosthetic devices. *Science*, Jun. 7;296(5574):1829-32.

[8] Wessberg, J., Stambaugh, C., Kralik, J., Beck, P., Laubach, M., Chapin, J., Kim, J., Biggs, S., Srinivasan, M., and Nicolelis, M. (2000). Real-time prediction of hand trajectory by ensembles of cortical neurons in primates. *Nature*, 408:361–365.

[9] Wu, W., Black, M. J., Gao, Y., Bienenstock, E., Serruya, M., Shaikhouni, A., and Donoghue, J. P. (2003). Neural Decoding of Cursor Motion using a Kalman Filter. *Advances in Neural Information Processing Systems 15, The MIT Press*.

Appendix

The subroutine **filter** is the standard Kalman filter:

$$
\begin{aligned}
[\tilde{\mathbf{x}}, \tilde{\mathbf{V}}, l] &= \mathbf{filter}(\mathbf{x}, \mathbf{V}, \mathbf{y}, \mathbf{H}, \mathbf{Q}, \mathbf{A}, \mathbf{W}) \\
\mathbf{x}_m &= \mathbf{A}\mathbf{x}, \\
\mathbf{V}_m &= \mathbf{A}'\mathbf{V}\mathbf{A}' + \mathbf{W}, \\
\mathbf{S} &= \mathbf{H}\mathbf{V}_m\mathbf{H}' + \mathbf{Q}, \\
\mathbf{K} &= \mathbf{V}_m\mathbf{H}'\mathbf{S}^{-1}, \\
l &= \mathcal{N}(\mathbf{y} - \mathbf{H}\mathbf{x}_m; 0, \mathbf{S}), \\
\tilde{\mathbf{x}} &= \mathbf{x}_m + \mathbf{K}(\mathbf{y} - \mathbf{H}\mathbf{x}_m), \\
\tilde{\mathbf{V}} &= (I - \mathbf{K}\mathbf{H})\mathbf{V}_m.
\end{aligned}
$$

The subroutine **collapse** approximates mixture of Gaussians as one Gaussian by matching the mean and covariance:

$$
\begin{aligned}
[\mathbf{x}, \mathbf{V}] &= \mathbf{collapse}(\{\tilde{\mathbf{x}}^i, \tilde{\mathbf{V}}^i, g^i\}_i) \\
\mathbf{x} &= \sum_i g^i \tilde{\mathbf{x}}^i \\
\mathbf{V} &= \sum_i g^i (\tilde{\mathbf{V}}^i + (\tilde{\mathbf{x}}^i - \mathbf{x})(\tilde{\mathbf{x}}^i - \mathbf{x})^T)
\end{aligned}
$$