Proceedings of the 25th Annual International Conference of the IEEE EMBS
Cancun, Mexico • September 17-21, 2003

Robust Neural Decoding of Reaching Movements for Prosthetic Systems

Caleb Kemere*, Maneesh Sahani†, and Teresa Meng*
*Department of Electrical Engineering, Stanford University, Stanford, CA
†W. M. Keck Foundation Center for Integrative Neuroscience, University of California, San Francisco, CA

Abstract— **A new neural prosthetic decoder architecture is presented which uses a hidden Markov model of typical arm movements to assist the reconstruction of intended trajectories from an ensemble of neural signals. The use of such a model results in a decoder which is robust to fewer or smaller neural signals. With limited information, the average error of the reconstructed trajectories produced by the robust decoder is half of that produced by the standard linear filter approach.**
Keywords— **Neural prosthetics, hidden Markov models**

I. Introduction

Recently, there has been a surge of interest in helping patients who are paralyzed or have other peripheral nervous system ailments by tapping into the undamaged motor centers to control prosthetic movements ([1], [2], [3]). In an ideal neural prosthetic interface, intended movements would be decoded from the combined signals of a large number of individual neurons, and then generated through muscle stimulation or other artificial means. In this paper, we address the decoding of reaches—goal-directed arm movements.

Current neural prosthetic interfaces are limited primarily by the amount of neural information, specifically the number of neurons, that can be interfaced through chronically-implanted electrode arrays. Earlier studies have suggested that accurate arm-movement reconstruction could be achieved by using simple algorithms to decode signals from approximately 500 neurons ([1]). While the advances of electrode technology may achieve interfaces on this scale within the decade, the further advance of prosthetics research into hand and finger motion will again present the challenge of limited neural information. Thus, what is needed is a robust decoder architecture that not only works well in the many-neuron regime, but still provides useful functionality when neural information is limited.

In this paper we introduce a scheme for improving neural decoder performance by using a hidden Markov model of arm trajectories to influence reconstruction.

A. Goal-Directed Movements

The basic premise of our decoder is that reaching movements are primarily goal directed. In other words, intended arm movements can be represented by a transformation from a point in goal space to a multidimensional trajectory in time.

$$\mathcal{T}(t; \mathbf{x}, \mathcal{C}) + \mathbf{n}(t) \rightarrow \mathbf{y}(t) \tag{1}$$

Here $\mathbf{x}$ is the goal location in some representational space, and $\mathbf{n}(t)$ represents variation in repeated movements to the same goal due either to internal noise or unidentified intention. $\mathcal{C}$ represents external constraints on the reaching movement (for instance an obstacle that must be avoided). Alternatively, the external constraints can be enfolded in the variation term, so that complex movements would be represented as "noisy" simple reaches. The black lines in Fig. 1 depict the average of many observed trajectories (the gray lines) to the same endpoint goal ($\mathcal{T}(t; \mathbf{x})$), and variation in these trajectories ($\mathbf{n}(t)$) is apparent.

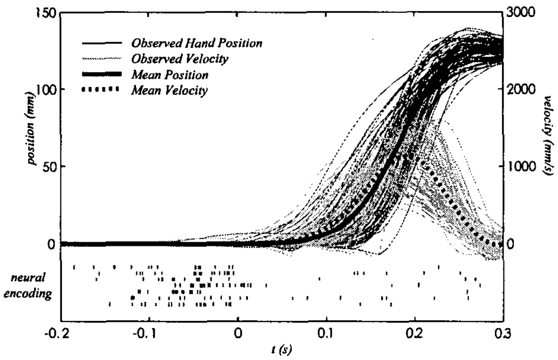

Fig. 1. Variation in observed movements and a typical realization of the associated neural signals

B. Robust Neural Decoding

In the prosthetic application, the intended movement cannot be directly observed. Rather, the decoder has signals from a large number of individual neurons which each encode some aspect of the intended movement. A good statistical characterization of these signals has proved to be the inhomogeneous Poisson process, in which an unobservable rate parameter controls the neural output ([4]). It has further been shown that, in motor cortical neurons, this mean firing can be reasonably well modeled as depending on the speed and direction of the intended movement.

$$\langle \lambda_k | \mathbf{v} \rangle = b_0 + \mathbf{b} \cdot \mathbf{v} \tag{2}$$

The mean firing rate of cell k is an affine transformation of the intended trajectory in velocity space, $\mathbf{v}$. In healthy individuals some neural signals precede the corresponding arm movement by a delay, typically on the order of 250 ms ([5]). Typical spike-sequences produced by a neuron ensemble are shown at the bottom of Fig. 1. In a prosthetic system, the movement decoder has access to information about the upcoming arm movement at least a quarter of a second in advance of the time the user expects it to take place—a decoder which does not take full advantage of this large amount of advance information will fail to achieve maximum performance.

Previous neural prosthetic decoding systems have been either excessively general or restrictive. Many of the current research approaches ([1], [2], [6]) decode intended arm movements by filtering a window of neural signals to find the incremental change in hand state (i.e., the velocity or position at the next time step). This approach is equivalent to modeling arm movements as a random walk in space, to which observed movements, especially reaching ones, bear little resemblance in practice. Alternatively, a stereotyped movement-based decoding system has been proposed ([7]) which strongly assumed that any variation in the trajectory of a reach was due to noise in the neural signals, rather than allowing for variations in the actual intended movement.

Thus, the goal of the robust neural prosthetic decoder is to use information about the time course of typical movements to aid, without over-constraining, the estimation of the desired trajectory from a noisy ensemble of neural signals related to it. For this study, we choose to limit ourselves to one dimensional trajectories, and simple neurons which fired as an inhomogeneous Poisson point process whose rate was an affine transformation of arm velocity. Though the algorithms will be discussed in this specific context, the essential concept can be extended to multiple dimensions and more complicated neural signaling models. Further discussion of these extensions can be found in the final section.

II. Decoder Algorithm

A. Basic Linear Filter

In the one dimensional, velocity-tuned neuron case, the optimal decoder can be expressed in a Bayesian sense. Given the observed neural firings from an ensemble of neurons, $\mathbf{f}_\tau = \left[f_\tau^1, f_\tau^2, \ldots, f_\tau^N\right]^T$, up to discrete time $t+\Delta$,

$$\mathbf{f}_\infty = \bigcup_{\tau=-\infty}^{t+\Delta} \mathbf{f}_\tau,$$

where Δ represents the amount of advance neural information available to the decoder, the *a posteriori* distribution of the average velocity at time t is

$$\Pr(v_t \mid \mathbf{f}_\infty) = \frac{\Pr(\mathbf{f}_\infty \mid v_t)\Pr(v_t)}{\Pr(\mathbf{f}_\infty)}. \tag{3}$$

However, as discussed above, most current approaches to decoding intended movements take an incremental or windowed approach. The optimal *windowed* decoder found in current approaches is obtained by using only the neural signals observed over a time interval, i.e., substituting $\mathbf{f}_t$ for $\mathbf{f}_\infty$ in equation (3). Furthermore, current windowed approaches also make no assumption about the distribution of v_t, leading to various approaches (linear filter, neural network), which approximate the maximum likelihood estimate

$$\hat{v}_t = \arg\max\{\Pr(\mathbf{f}_t \mid v)\}. \tag{4}$$

If the firing rate of the kth neuron in the ensemble is modeled as

$$f_t^k = \beta(v_t + \alpha) + n_t^k, \tag{5}$$

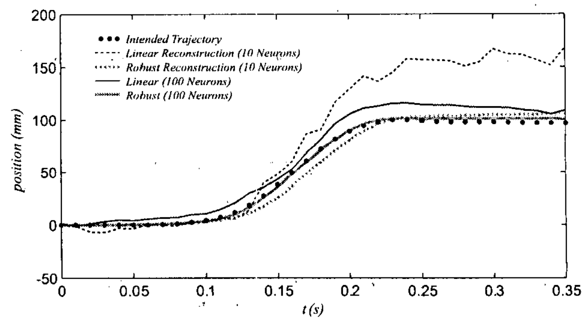

Fig. 2. Intended trajectory and reconstructions using traditional linear filter and new robust methods

where $n_t^k \sim \mathcal{N}\left(0, \sigma^2\right)$, then with no prior information about v_t, the maximum likelihood solution is just the average of the affinely scaled f_t^k's

$$\begin{aligned}\hat{v}_t &= \beta^{-1}\left(\mathbf{1}^T\mathbf{1}\right)^{-1}\mathbf{1}^T\mathbf{f}_t - \alpha \\ &= \beta^{-1}\overline{f_t} - \alpha\end{aligned} \tag{6}$$

The decoded arm trajectory at any point in time is then the sum of all the $\hat{v}_t$'s up to the current time. Fig. 2 depicts a sample trajectory with the results of this linear filter reconstruction with the signals from 10 and 100 neurons captured in 10 ms windows. Neurons were modeled as described above, with their firing rates scaled so that the maximum rate was 50 per second and the minimum was zero . One essential characteristic of this class of simple decoders is that they perform poorly when few neural signals are available, but scale well with the number of neurons interfaced, as is apparent in the figure. Note that variation in the number of neurons and the dynamic range of the spike rate are both measures of the amount of neural information available to the decoder. While the analysis in this work is done relative to the number of neurons interfaced, it could equivalently have been done by increasing or decreasing this parameter.

B. HMM Trajectory Reconstruction

We can model the velocity of the hand at discrete time t, v_t, as being the output of some unobservable "state" of the model system, $s_t \in \{q_1, q_2, \ldots, q_M\}$. A hidden Markov Model (HMM) is a special case of the general class of latent variable models, which share the basic similarity that observed data is produced by some unobservable stochastic process. The HMM has the special property that the progression of states in time is a first-order Markov process, or

$$\begin{aligned}\Pr(s_{t+1} = q_j \mid s_t, s_{t-1}, \ldots) &= \Pr(s_{t+1} = q_j \mid s_t) \\ &= a_{ij}.\end{aligned}$$

Furthermore, the observed data at time t, O_t, depends only on the current state, that is

$$\Pr(O_t \mid s_t, s_{t-1}, \ldots) = \Pr(O_t \mid s_t = q_j).$$

Thus, the HMM is fully specified by three parameters, the matrix of transition probabilities, $A = \{a_{ij}\}$, the state dependent output density, $b_j = \Pr(O_t \mid s_t = q_j)$, and the initial state distribution, $\pi_j = \Pr(s_0 = q_j)$.

To model hand velocity trajectories, we constrained the general HMM. First, the states are connected in a "left-right" manner: in time, a state can only transition to itself, or a higher indexed state. In other words, $a_{ij} = 0$ for all $i < j$. Second, the state output density is modeled as Gaussian,

$$b_j = \Pr(v_t \mid s_t = q_j) \sim \mathcal{N}\left(\mu_{q_j}, \sigma_{q_j}^2\right). \tag{7}$$

The model parameters, $\{A, \mathbf{b}, \boldsymbol{\pi}\}$, are trained using the Baum-Welch method with a large number of reach trajectories (see, e.g., [8]).

Given our hidden Markov model of trajectories, we can rewrite (3) as

$$\begin{aligned}\Pr(v_t \mid \mathbf{f}_\infty) &= \sum_{s_t} \Pr(v_t, s_t \mid \mathbf{f}_\infty) \\ &= \sum_{s_t} \Pr(v_t \mid s_t, \mathbf{f}_\infty) \Pr(s_t \mid \mathbf{f}_\infty).\end{aligned} \tag{8}$$

If $\Pr(\mathbf{f}_t \mid v_t)$ is as defined in (5), then

$$\begin{aligned}\Pr(v_t \mid s_t, \mathbf{f}_\infty) &= \Pr(v_t \mid s_t, \mathbf{f}_t) \\ &= \frac{\Pr(\mathbf{f}_t \mid v_t) \Pr(v_t \mid s_t)}{\int \Pr(\mathbf{f}_t \mid v_t) \Pr(v_t \mid s_t) dv} \\ &\sim \mathcal{N}(\nu_{s_t}, C_t).\end{aligned}$$

The covariance of this density, C_t, is unimportant to the derivation. The mean, ν_{s_t} is

$$\nu_{s_t} = \mu_{s_t} + \frac{\sigma_{s_t}^2}{\sigma_{s_t}^2 + \frac{\sigma^2}{N}} \left(\hat{v}_t - \mu_{s_t}\right), \tag{9}$$

where $\hat{v}_t$ is as defined in (6), and N is the number of available neurons. $\Pr(s_t \mid \mathbf{f}_\infty)$ can be evaluated using the forward-backward algorithm using modified HMM parameters. The conditional mean estimate is then simply the mean of (8).

$$\mathrm{E}(v_t \mid \mathbf{f}_\infty) = \sum_{s_t = q} \nu_{s_t, t} \Pr(s_t \mid \mathbf{f}_\infty) \tag{10}$$

To maximize our ability to match observed trajectories, we use an ensemble of HMMs, each trained on a subset of movements, selected by some clustering criteria (here we divided the training movements using the endpoint extent). The Bayesian estimate can then be further extended across the separately trained HMMs. Thus the final robust estimate is

$$\mathrm{E}(v_t \mid \mathbf{f}_\infty) = \sum_m \mathrm{E}(v_t \mid m, \mathbf{f}_\infty) \Pr(m \mid \mathbf{f}_\infty), \tag{11}$$

where $\mathrm{E}(v_t \mid m, \mathbf{f}_\infty)$, the *a posteriori* estimate given model m, is computed in (10) and $\Pr(m \mid \mathbf{f}_t)$ is also found through the forward backward algorithm.

Thus, the basis of the refined HMM decoder architecture we propose is quite simple. The movement decoding is done in two stages. First, the neural signals are combined using a classic direct approach to estimate the current physical state of the arm. In Fig. 3, this is represented by the "Linear Decoder" block (though in principle this estimator need not be linear). Then, the forward-backward algorithm is used to evaluate the best fit of the HMM states to this initial estimate. Finally, at each point in time the conditional state densities are used in conjunction with (11) to calculate the robust estimate.

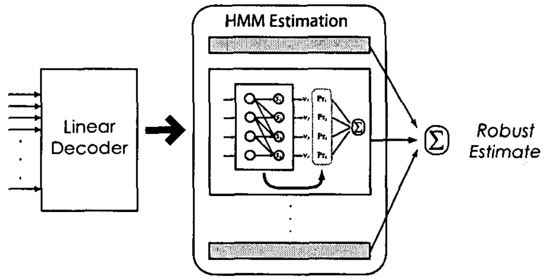

Fig. 3. Robust decoder architecture

As discussed above, in the simulations presented in this paper, the neurons are assumed to signal as an inhomogeneous Poisson point process. Thus, the model presented for neuron outputs in (5) is incorrect, and optimal decoding methods based on it will be suboptimal for Poisson neurons. However, as we show, significant gains can still be made using the approximation of additive noise. In the case of a doubly stochastic Poisson process (as the output of the model neurons are), the mean and variance are well defined. Thus, for the Gaussian approximation, the mean firing rate will still be as defined in (5),

$$\langle f_t^k \rangle = \beta(\mu_{s_t} + \alpha),$$

and the variance will be the sum of the mean of the velocity distribution and its variance, or

$$\sigma^2 = \beta(\mu_{s_t} + \alpha) + \sigma_{s_t}^2.$$

In Fig. 2 the solid gray lines are the robust estimates generated with 10 and 100 neurons. Notice that in the 10 neuron case, unlike the linear filter, the robust decoder is able to suppress the effect of noise during the middle of the movement. In the 100 neuron case, the algorithm is still useful, but its effect is smaller.

III. Simulation Results

Figure 4 shows the performance of the robust HMM decoder relative to that of the basic linear filter as the number of neurons available increases. The dotted lines in the figure represent the average error of reconstructions of simulated trajectories (generated using the minimum jerk model of [7] with additional noise added in the velocity domain). The solid lines represent the average error of reconstructions of actual trajectories captured on the infrared tracking system of our collaborators. (A subset of these are depicted in Fig. 1.) In both cases, neural signals are generated randomly using the previously described stochastic model. While we have used the term "trajectory" interchangeably to represent the time courses in velocity and position spaces, it is important to note that the error metric presented in Fig. 4 is the average square distance

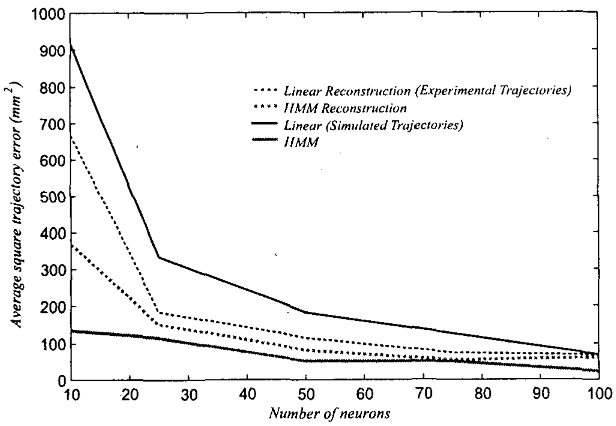

Fig. 4. Decoder performance as number of neurons available increases

between the intended and reconstructed arm *positions*, because this is most relevant to the patient.

The dynamic range and peak velocity of the actual trajectories is higher than that of the simulated ones. As a result, the linear filter reconstructions of these trajectories have, on average, less error. Furthermore, because the experimentally gathered arm trajectories have considerably more variation, and limited data with which to train, the robust trajectory estimates have higher error than those corresponding to simulated reaches. Despite this, even on actual arm trajectories, robust decoding results in an average of 25 percent less error (70 percent for the simulated movements). An alternate performance metric would be the number of neurons required for useful performance. The endpoints ranged from 50 to 140 mm in extent; thus, we can define "acceptable" average trajectory error as less than 10 percent of the final endpoint, or about 10 mm (100 mm^2 average square error). In this case, for the actual arm trajectories, the typical linear filter approach would require about 75 neurons. The robust decoder needs only about 50! In the case of the simulated trajectories, the situation is even more extreme 30 neurons compared to 90 for the linear filter approach.

IV. Conclusion

Using an appropriate model as a prior for decoding reaching movements is clearly preferable to movement-ignorant methods in almost every situation. The technique we present for combining the outputs of a HMM that has been trained to represent typical reaching movements is a novel way of achieving this goal. The performance increase, an average of 25 percent reduction in error, or a one-third reduction in the required number of neural signals (70 percent and two-thirds for the larger data set of simulated movements), highlights the utility of this technique. Furthermore, our method's scalability and computational efficiency is an improvement over the maximum likelihood method presented in [7].

We have modeled trajectories using a HMM with little discussion of the primary assumption behind the technique, that the hidden process has first-order Markov characteristics. It has been shown that variation in endpoint-directed movements is larger in dimensions that do not affect the endpoint error than those dimensions which do. Thus, for example, higher velocity early in a movement may be counteracted with lower velocity later in order to maintain endpoint accuracy. This type of relationship violates the Markov assumption if our models consider only the velocity domain. However, in general, we expect to find that an intelligent choice of the state variable(s) can account for such variation while still providing the computational efficiency that the HMM structure allows.

It was suggested earlier that a reduction to one dimension and simple velocity tuned neurons can be made without loss of generality. In this work, we presented the problem of estimating the mean of the *a posteriori* density $\Pr(v_t \mid \mathbf{f}_\infty)$ because of its convenience. Extending the model to reconstruct multidimensional trajectories can be accomplished by instead using the *a posteriori* density $\Pr(\mathbf{x}_t \mid \mathbf{f}_\infty)$, where $\mathbf{x}_t$ is a vector representing physical state, including multiple dimensions and relevant derivatives. The result, which also incorporates a model of arm dynamics, becomes a form of dynamic Bayesian network for which similar, but more complex, estimation techniques exist.

Finally, one consequence of using only one-dimensional trajectories is an illusory increase in performance compared to the two-dimensional results referenced earlier. Because the second and third dimension are essentially independent of the first, an additional set of neurons are required to decode them. Thus, the robust decoding scheme is quite relevant to current interface systems as the regime in which its performance dramatically exceeds that of current methods contains the range of current electrode technology.

Acknowledgment

The authors would like to thank Dr. Mark Churchland for the arm trajectory data he collected and generously provided.

References

[1] J. Wessberg, *et al.*, "Real-time prediction of hand trajectory by ensembles of cortical neurons in primates," *Nature*, vol. 208, pp. 361–365, Nov. 2000.

[2] D. Taylor, S.I.Helms-Tillery, and A. Schwartz, "Direct cortical control of 3d neuroprosthetic devices," *Science*, vol. 296, no. 3, pp. 1829–1832, June 2002.

[3] J. Chapin, K. A. Moxon, R. S. Markowitz, and M. A. L. Nicolelis, "Real-time control of a robot arm using simultaneously recorded neurons in the motor cortex," *Nature Neuroscience*, vol. 2, no. 7, pp. 664–670, July 1999.

[4] C. Koch, *Biophysics of Computation*. New York, NY: Oxford University Press, 1999.

[5] D. Moran and A. Schwartz, "Motor cortical representation of speed and direction during reaching," *Journal of Neurophysiology*, vol. 82, no. 5, pp. 2676–2692, Nov. 1999.

[6] M. D. Serruya, N. G. Hatsopoulos, L. Paninski, M. R. Fellows, and J. P. Donoghue, "Brain-machine interface: Instant neural control of a movement signal," *Nature*, vol. 416, pp. 141–142, 14 Mar. 2002.

[7] C. T. Kemere, G. Santhanam, B. M. Yu, K. V. Shenoy, and T. H. Meng, "Decoding of plan and perlmovement neural signals in prosthetic systems," in *Proc. IEEE Workshop on Signal Processing Systems (SIPS '02)*, San Diego, CA, Oct. 2002, pp. 276–283.

[8] L. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.