

Model-Based Neural Decoding of Reaching Movements: A Maximum Likelihood Approach

Caleb Kemere*, *Student Member, IEEE*, Krishna V. Shenoy, *Member, IEEE*, and Teresa H. Meng, *Fellow, IEEE*

Abstract—A new paradigm for decoding reaching movements from the signals of an ensemble of individual neurons is presented. This new method not only provides a novel theoretical basis for the task, but also results in a significant decrease in the error of reconstructed hand trajectories. By using a model of movement as a foundation for the decoding system, we show that the number of neurons required for reconstruction of the trajectories of point-to-point reaching movements in two dimensions can be halved. Additionally, using the presented framework, other forms of neural information, specifically neural “plan” activity, can be integrated into the trajectory decoding process. The decoding paradigm presented is tested in simulation using a database of experimentally gathered center-out reaches and corresponding neural data generated from synthetic models.

I. INTRODUCTION

RECENTLY, there has been a surge of interest in assisting individuals who are paralyzed or have other peripheral nervous system ailments by tapping directly into the undamaged motor centers to allow controlled prosthetic movements. Neural signals in the motor (M1) and pre-motor areas of cortex are known to generate voluntary movement. When recorded during movements repeated to the same target or along the same path, neurons produce a series of characteristic output pulses (action potentials) at a similar rate. Furthermore, the pattern of neural firing has been found to relate to aspects of the movement, speed and direction [1], or target location [2]. More generally, it is widely believed that M1 neurons act as control signals to drive movements, a complex dynamical system [3].

Functional MRI studies of brain activity have shown that neural activity in the relevant brain areas continues to be present when the signals are blocked from reaching muscles (e.g., due to spinal cord injury). Thus, the task of the neural prosthetic decoder is to deduce the intended movement from these neural signals. We will show that if we loosely restrict the prosthetic user’s reaching movements, the performance of the system can be enhanced dramatically.

Manuscript received July 15, 2003; revised January 16, 2004. This work was supported by MARCO Contract 2001-CT-888, by DARPA Contract MDA972-02-1-0004, by the National Science Foundation (NSF) Center for Neuromorphic Systems Engineering at Caltech, by the Burroughs Wellcome Fund Career Award in Biomedical Sciences, by the Center for Integrated Systems at Stanford University, and by the Sloan Foundation. *Asterisk indicates corresponding author.*

*C. Kemere is with the Department of Electrical Engineering, Stanford University, Stanford, CA 94305 USA (e-mail: ckemere@mojave.stanford.edu).

K. V. Shenoy is with the Department of Electrical Engineering and the Neurosciences Program, Stanford University, Stanford, CA 94305 USA (e-mail: shenoy@stanford.edu).

T. H. Meng is with the Department of Electrical Engineering, Stanford University, Stanford, CA 94305 USA (e-mail: meng@mojave.stanford.edu).

Digital Object Identifier 10.1109/TBME.2004.826675

A. Previous Approaches

The first generation of interface systems has relied primarily on a simple inversion of the mapping of neural signals to reach target or trajectory, where the mapping is discovered explicitly (i.e., as a separate step of the process) or implicitly from training data. A simple representation of this mapping, \mathcal{F} , from target (\mathbf{x}) or trajectory space (\mathbf{y}), to neural signal space, $\mathbf{s}_n(t)$ at time t is

$$\{\mathbf{x}, \mathbf{y}\} \xleftrightarrow{\mathcal{F}} \begin{bmatrix} \vdots \\ \mathbf{s}_n(t) \\ \vdots \end{bmatrix}_N \xrightarrow{\widehat{\mathcal{F}^{-1}}} \{\widehat{\mathbf{x}}, \widehat{\mathbf{y}}\} \quad (1)$$

where the neural signal vector has been expanded to emphasize the multiple neural channels it contains. The corresponding inverse $\widehat{\mathcal{F}^{-1}}$ is used to estimate $\widehat{\mathbf{x}}$ or $\widehat{\mathbf{y}}$.

One benefit of assuming a simple linear mapping is that the appropriate inverse is a linear filter, which may be implemented with low latency as in the recent demonstrations of cortical control with visual feedback [4]–[8]. The performance of these inverse-mapping prosthetic interfaces is limited primarily by the amount of neural information, specifically the number of neurons, that can be interfaced through chronically-implanted electrode arrays. Estimates of the number of neural channels required for “acceptable” performance have ranged from as high as 500 to as low as 20.

Prosthetic interfaces may currently be divided into two categories by the object of their decoding process, $\widehat{\mathbf{x}}$ in (1). The first decode the target or goal of a movement, and the second the time evolution of the arm’s trajectory.

1) *Target Location Estimation*: Target-decoding systems commonly rely on “plan” activity—neural activity which appears in the period between target onset and a movement cue, and directly encode aspects of a movement such as target direction or distance.

Recent decoding systems in this genre [8]–[10] have used probabilistic approaches (maximum likelihood (ML) or more general Bayesian methods) to generate estimates of the target locations of center-out reaches from plan activity. These types of decoders would be very useful in an environment in which the user wishes to pick between a set of discrete reach endpoints.

2) *Trajectory Reconstruction*: Under the alternative decoding paradigm, the prosthetic interface seeks to generate an estimate of the intended trajectory of a user’s arm. These systems, have exclusively relied on “movement” activity which occurs during an ongoing movement and typically displays a time evolution which reflects aspects of the movement such as speed, direction, and/or force. While some studies have

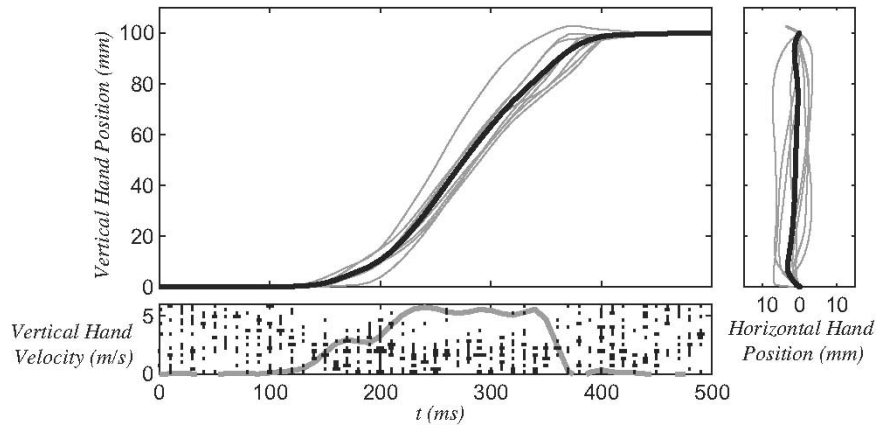


Fig. 1. *Top*: repeated reaching movements to a given endpoint share many similarities, but also display variation. Shown are the trajectories of eight center-out reaches to a point 10 cm above the origin, and the canonical trajectory formed by their average (thick line). To highlight the variation, the trajectories have been scaled and rotated slightly so that they line up exactly at beginning and end. *Bottom*: 20 channels of synthetic neural signals (one channel per row) overlaid on a the vertical component of the velocity of one of the reaches from above, both sampled at 10 ms. The width of the hash marks is proportional to the number of action potentials observed. Neurons produce pulse-like signals—a prosthetic interface uses the time of these pulses from an ensemble of neurons to decode movement intention. Here, the top ten neurons prefer downward movements while the bottom ten prefer upward ones.

examined continuous tracing movements [10], most focus on the reconstruction of point-to-point reaches. An argument may be made that these are the most commonly initiated behaviors, and thus an appropriate concentration [11].

Significant effort has been invested in reconstructing arm trajectories that most closely match the corresponding movements actually executed by behaving monkeys. Initial systems typically relied on linear filters or a related method known as the “population vector.” However, while a linear mapping between movement activity and arm movements in some cases provides an acceptable approximation [12], the relationship is inherently nonlinear, and therefore nonlinear decoding methods often provide an increase in performance [4], [13], [14].

B. First and Second Generation

Unfortunately, the trajectories reconstructed using what we term “first-generation” inverse mapping methods sometimes only loosely resemble the time course expected for the corresponding real reaching movements (as will be shown in Fig. 2). Furthermore, the amount of history used by the decoder is often large, on the order of 1 s or more, making the reconstruction of more abrupt movements difficult. “Second-generation” prosthetic decoding systems offer improved performance by including some model of the reaching arm. For example, in [15], a model of arm kinematics is used to enforce some aspects of reaching movements, such as smoothness, on the decoder. While this approach meets our second-generation criteria of including a prior model of arm motion, the assumption that the arm moves in a random walk—that the innovation in its state from one time period to another is controlled by a random process—is incomplete in the context of point-to-point reaches or any movement which proceeds in a directed manner. If we assume that the user wishes to make goal-directed movements, then what is required is not only a prior model of arm kinematics, but also a complete model of the reach.

Finally, little attention has been paid to using the plan activity to generate appropriate trajectories for reaches. In addition, the combination of plan and movement neural activity, in the context of reconstructing trajectories, is unexplored. In Section II,

with the assumption that the user wishes to make a goal-directed reach, we present a simple conceptual framework for achieving both of these goals while also delivering increased accuracy with only movement activity.

II. MODEL-BASED REACH DECODING

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. Fig. 1 depicts a set of hand trajectories recorded during repeated reaching movements to the same target. They are quite similar in several respects, ranging from a lack of loops or sudden stops and starts, to the actual time of flight and sigmoidal shape. These aspects prove to be quite general across different scales of movement and between individuals. However, there is variation in the observed trajectories. This variation is a combination of intentional variation caused by the whim of the test subject and unintentional variation caused by noise inherent in the neuro-muscular system. Thus, a given observed trajectory $\mathbf{y}(t)$ can be represented as

$$\mathcal{T}(t; \mathbf{x}) + \mathbf{n}(t) \rightarrow \mathbf{y}(t) \quad (2)$$

where \mathbf{x} is the goal location in some representational space and $\mathbf{n}(t)$ represents both intended and unintended variation. The thick black line in Fig. 1 is the “canonical trajectory” \mathcal{T} , an average of many observed trajectories (the gray lines) to a particular endpoint, and variation in these trajectories \mathbf{n} is apparent. If the mapping is well understood, and fully accounts for the subject’s goals, the variation term is small, and the canonical trajectory is a good approximation of the specific trajectory. Thus, we modify (1) to account for the goal-directed nature of the trajectory and anticipate the model-based decoding approach

$$\mathbf{x} \xrightarrow{\mathcal{T}} \mathbf{y}(t) \xleftarrow{\mathcal{F}} \begin{bmatrix} \vdots \\ s_n(t) \\ \vdots \end{bmatrix}_N \xrightarrow{(6)} \hat{\mathbf{x}} \xrightarrow{\hat{\mathcal{T}}} \hat{\mathbf{y}}(t). \quad (3)$$

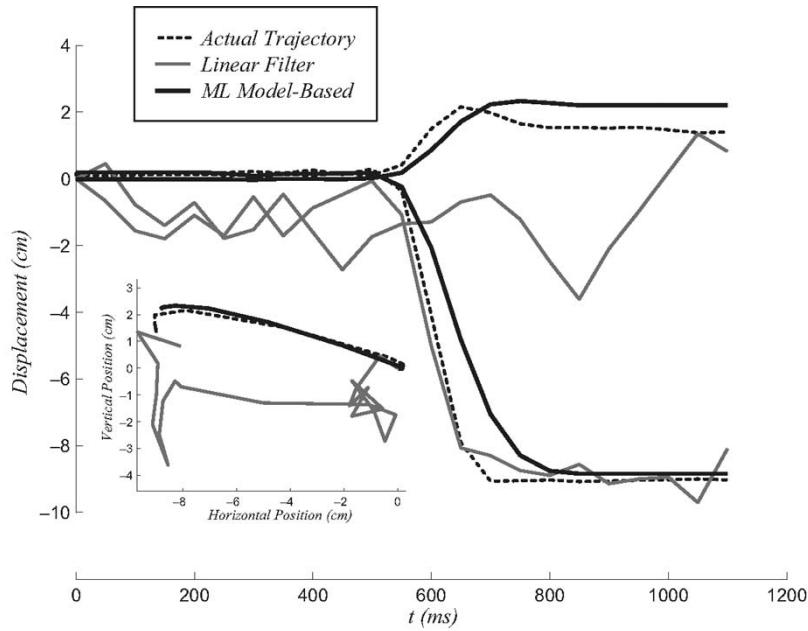


Fig. 2. Trajectories reconstructed using the model based paradigm presented in this work more closely resemble real trajectories than those reconstructed using simple inverse techniques (such as linear filters). Shown here is the reconstruction of an intended movement using the activity of 52 movement-tuned neurons. *Inset*: trajectory reconstruction in two dimensions is even more revealing.

Here, the movement goal \mathbf{x} has been converted into a unique trajectory $\mathbf{y}(t)$ which is encoded in peri-movement activity through the transformation \mathcal{F} . Below, we show how ML can be used to extract both goal $\hat{\mathbf{x}}$ and trajectory $\hat{\mathbf{y}}(t)$ estimates.

If we observe many reaching movements, we find that trajectory variation is primarily specified by the target location. This suggests a parameterization for the model discussed above, and thus that increased decoder accuracy may be obtained by using the neural data to estimate this parameter. We can write the likelihood of the arm position at any time during a movement as

$$\mathcal{L}(t; \mathbf{x}) = \prod_{n=1}^N \Pr(s_n(\mathbf{t}) | \mathcal{T}(t; \mathbf{x})) \quad (4)$$

where $s_n(\mathbf{t})$ contains the time of every action potential observed from neuron n from the beginning of the observation period up to time t .

Notice that this likelihood formulation enables the straightforward integration of plan activity into the estimation system. As discussed above, plan activity is defined as containing information about the goal or target of the reach. Thus, it introduces a new term to the likelihood function, as follows:

$$\mathcal{L}(t; \mathbf{x}) = \left[\prod_{m=1}^{N_m} \Pr(s_m(\mathbf{t}) | \mathcal{T}(t; \mathbf{x})) \right] \cdot \left[\prod_{p=1}^{N_p} \Pr(s_p(\mathbf{t}) | \mathbf{x}) \right]. \quad (5)$$

Here, $s_m(\mathbf{t})$ is the signal observed from movement neuron m and $s_p(\mathbf{t})$ that from plan neuron p .

Under many conditions, the optimal trajectory estimate will correspond to the endpoint which maximizes the likelihood. That is

$$\hat{\mathbf{x}} = \underset{\mathbf{x}}{\operatorname{argmax}} [\mathcal{L}(t; \mathbf{x})] \quad (6)$$

$$\hat{\mathbf{y}}(t) = \mathcal{T}(t; \hat{\mathbf{x}}). \quad (7)$$

Intuitively, the ML estimate of the current arm position is generated by evaluating which of a family of arm trajectories—indexed by the movement endpoint—best matches the data, and then choosing the current position along the best trajectory for the instantaneous trajectory estimate. In cases with an abundance of neural data and highly variable trajectories, a Bayesian solution, which essentially averages between potential trajectories in a probabilistically optimal manner, may be preferable [16].

It is important to recognize that neither the trajectory model \mathcal{T} , nor the probability distributions describing the mapping between neural signal and arm trajectory and/or goal location, are fully known. There are several approaches for approximating \mathcal{T} . One could refer to the literature and use models such as minimum jerk [17]; one could generate a complex statistical model from data [16]; or, one could generate a simple model from available data. This last approach is used in the following section. A statistical characterization of the neural signals could also be found empirically or assumed. Further consideration of these approximations can be found in Section V.

III. SIMULATION FRAMEWORK

To demonstrate the utility of our decoder paradigm, we simulated it using synthetic neural data. This is somewhat atypical, as the performance of neural prosthetic decoding algorithms is usually demonstrated using experimentally gathered neural data. However, in order to most clearly present the benefits of model-based decoding over first generation inverse-mapping approaches, we chose to entirely avoid the problem of estimating neural signal mappings. Rather, as will be shown, we synthesized neural data using the simple models described below. Using the following models for the relationship between neural activity and movement, we generated neural signals to correspond to movements in a database of 1369 two-dimen-

sional point-to-point reaches obtained from a single human subject. The human subject performed center-out reaches to visual point targets in a 20-cm square fronto-parallel workspace. The trajectory of the hand was captured using an optical tracking system and was considered acceptable if it satisfied very broad conditions, the primary one being that the reaction time was less than 0.5 s. In Section V, we address the impact of using synthetic neural data on the simulation results.

A. Neural Signals

In the prosthetic application, the intended movement cannot be directly observed. Rather, the decoder has signals from a large number of individual neurons each of which encode some aspect of the intended movement. Preprocessing of the signal on each interfaced electrode yields the times that a given neuron emits an action potential. Thus, the resulting data set constitutes a vector of observations of point process time series, one dimension for each neuron. The bottom panel of Fig. 1 depicts data that might be gathered from an ensemble of neurons during a reach along one of the pictured trajectories.

It has been shown that modeling neurons as firing randomly in time as an inhomogeneous Poisson point process captures much of the statistical variation of neural firing [18]. Thus, at time t , we model the distribution of the number of action potentials, $k^\Delta(t)$, observed within a window of duration Δ as

$$\begin{aligned} \Pr(k^\Delta(t) = \kappa) &= \frac{1}{\kappa!} \left(\int_t^{t+\Delta} \lambda(\tau) d\tau \right)^\kappa e^{-\int_t^{t+\Delta} \lambda(\tau) d\tau} \\ &= \frac{(f^\Delta)^\kappa}{\kappa!} e^{-f^\Delta} \end{aligned} \quad (8)$$

where f^Δ is the integral of $\lambda(t)$, the instantaneous rate of the process, over the time window.

1) *Movement Activity*: Motor cortical neurons have been shown to fire proportionally to many variables, including hand velocity, hand force, and muscle forces within the arm [3]. However, under many circumstances, observed firing rates approximately vary with the cosine of the angle between hand velocity and a preferred direction which differs from neuron to neuron [1]. This model for movement neural activity is often dubbed ‘‘cosine tuning.’’ As mentioned above, real neurons have more complex behavior than this simple linear tuning. However, since the purpose of this work is not to provide a better first generation inverse model, but rather to examine a new decoding paradigm, we choose the following simple tuning function for our synthetic neurons. After digital sampling, the mathematical representation of the sampled Poisson process mean for neuron m , f_m^Δ , is

$$f_m^\Delta(t - \delta) = \Delta \left(\lambda \cdot \frac{\hat{\mathbf{e}}_m \cdot \dot{\mathbf{y}}(t)}{\|\mathbf{v}_{\max}\|} + \beta \right) \quad (9)$$

where Δ is the time quantization, λ and β are constants, and $\dot{\mathbf{y}}(t)$ is the average velocity of the trajectory over $[t, t + \Delta)$. The δ term reflects the observed result that the neural signal precedes the corresponding arm movement by some delay, typically on the order of 100 to 200 ms. In our simulations, we fix this value at 100 ms. Finally, $\hat{\mathbf{e}}_m$ is a unit vector in the preferred motion

direction of the neuron; this is the only parameter that is varied on a per-neuron basis in our simulations.

2) *Plan Activity*: It has been reported that during a task in which a plan period is enforced by inserting a random-length period between target presentation and a go-cue, neural firing rates in certain regions vary both as a function of the direction and distance of the target from a central touch point [2]. Therefore, for similar reasons as used in modeling movement activity, we adopt a vector-cosine model for the tuning of plan activity. Furthermore, in agreement with the typical understanding, in our model, the firing rate of plan neurons remains constant throughout the pre-movement period when their activity is related to reaches. Thus, for neuron p

$$f_p(\mathbf{x}) = T_{\text{plan}} \left(\lambda \cdot \frac{\hat{\mathbf{e}}_p \cdot \mathbf{x}}{\|\mathbf{x}_{\max}\|} + \beta \right) \quad (10)$$

where f_p is the mean of the Poisson process over the duration of the pre-movement interval T_{plan} , λ , and β constants, and $\hat{\mathbf{e}}_p$ is a unit vector in the direction of preferred targets. Most experimental studies of the tuning of plan activity enforce a delay period. In natural reaching movements, the duration of plan activity appears to be on the order of 200 ms. In our simulations, we assume a constant plan period with a conservative duration of 150 ms.

B. Decoding Algorithms

A major reason for the proliferation of first generation inverse-mapping decoding systems is the complexity of the mapping between arm movements and neural activity. In order to avoid modeling and learning this mapping, as mentioned above, we have chosen a simple linear tuning model for movement activity. A further benefit of this design choice is the resulting simple inverse-mapping decoder. For linear tuning, the minimum mean square error decoder is just a linear filter. If one rewrites (9) for neuron m as

$$f_m(t) = A \hat{\mathbf{e}}_m \cdot \dot{\mathbf{y}}(t + \delta) + B \quad (11)$$

then, given the observed firing of M neurons concatenated into a column vector \mathbf{f} , the standard linear unbiased estimator for $\dot{\mathbf{y}}(t + \delta)$ is given by

$$\dot{\mathbf{y}}_{\text{est}} = \frac{1}{A} (\mathbf{E}^T \mathbf{E})^{-1} \mathbf{E}^T (\mathbf{f} - B) \quad (12)$$

where \mathbf{E} is a matrix formed from the concatenation of the preferred directions of the neurons. The trajectory of the arm can be reconstructed by summing the estimated velocities. This result closely resembles that of [4] except that our added knowledge of the neural tuning permits the direct formation of the matrix \mathbf{E} rather than requiring its estimation.

We tested the model-based decoding paradigm by estimating the ML trajectory as described above. Previously ([17]), we used an analytic minimum-jerk model to represent canonical trajectories to a particular endpoint. In this work, we make use of a potentially more powerful approach—using the data set itself to generate the movement model. Canonical trajectories are generated from our database by averaging the reaches, aligned by movement onset, whose endpoints are close (within a certain

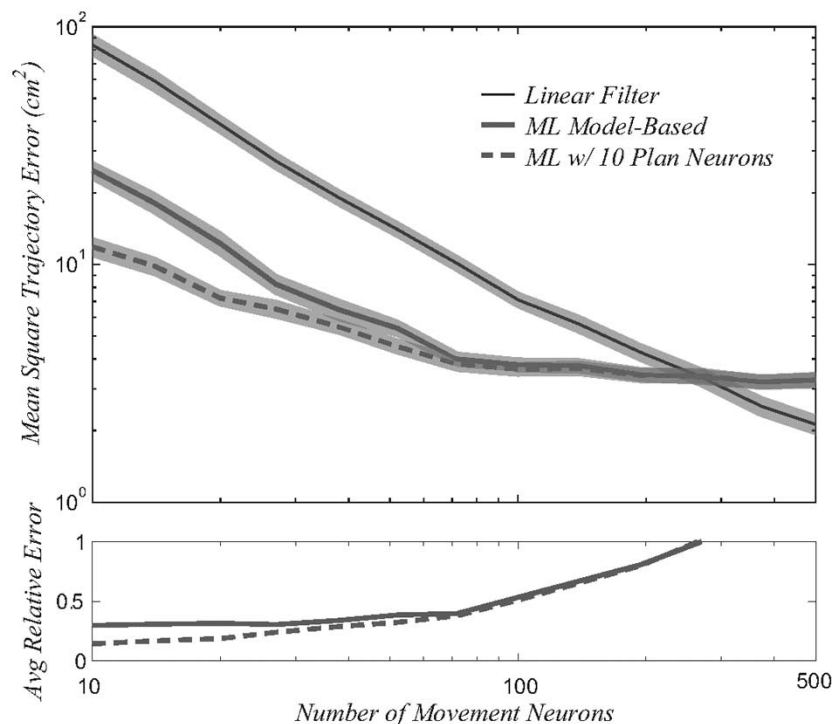


Fig. 3. *Top*: Average mean square error for a ML model-based decoder with 36 canonical trajectories as the number of movement-tuned neurons increases. The shaded regions show the standard errors of the averages. Notice how the addition of plan activity has a significant impact on system performance when the number of movement neurons is low to moderate. *Bottom*: In the regime of current neural interfaces (about 100 electrodes), the simple 36 canonical trajectory decoder on average has half of the error of a linear filter. As the amount of neural data increases, the complexity of the ML decoder would have to be increased to maintain this performance differential.

L_∞ -norm distance). While this approach is attractive in its generality, for a fixed data set the spatial density and statistical accuracy of the canonical trajectories obtained clearly trade off; we chose the actual number of canonical trajectories for this work in an *ad hoc* manner.

Also previously, no provision was made for estimating the start of the movement. In our data set, this task is made nontrivial by variation in the duration of the interval between the time the subject was cued to move and began moving (the “reaction time”). While the formulation of [16] proves quite helpful for this particular problem, for this work we simply expand the ML calculation to include a variable initial delay. For each canonical trajectory, the ML delay is calculated, and then the ML trajectory is chosen from the optimally delayed canonical trajectories.

IV. SIMULATION RESULTS

We generated synthetic neural activity using the vector-cosine tuning models for plan and movement activity described in Section III. Preferred directions are chosen at random, and each neuron modulates its firing rate between 0 and 20 spikes/s, with the maximum and minimum rates corresponding to a maximum velocity of 0.6 m/s (for movement neurons) or a maximum displacement of 50 cm (for plan neurons) in the preferred and anti-preferred directions respectively. The canonical trajectories are indexed by endpoint, with the endpoints evenly spaced throughout the workspace. Each canonical trajectory is calculated by averaging those trajectories in the database whose endpoints lie within a set distance of the canonical endpoint, excluding the trajectory to be reconstructed. A spike train corresponding to the test trajectory is generated for each neuron in

our ensemble using (8) with a sampling period of 50 ms. Then, using (9), (10), and (5), for each canonical trajectory, the conditional likelihood of the ensemble of neural signals is calculated. Finally, the ML estimate is generated using (7). We repeated this process about 200 times to generate the averaged results of Figs. 3 and 4.

Fig. 2 compares a trajectory reconstructed using the inverse-mapping approach with the ML estimate generated from the model-based decoding paradigm using only movement-tuned neural activity. In this particular example, the endpoint of the linear filter reconstruction is somewhat close to that of the actual movement. However, as mentioned earlier, what is apparent is the unrealistic nature of the trajectories that are typical of first generation systems. The ML model-based algorithm, on the other hand, with only 36 canonical trajectories, produces a movement which is quite natural, as well as more closely following the actual trajectory.

Fig. 3 depicts the dependence of trajectory error on the number of neurons available to the decoder. Because medical and electrode technology limit the number of neurons to be interfaced, an interesting comparison of decoding techniques involves the number of neurons required to achieve acceptable performance. If we arbitrarily select the maximum acceptable mean square trajectory error to be 10 cm², then the ML approach achieves this level of accuracy with less than half of the neurons necessary for the standard linear filter approach (26 versus 67). The dashed gray line in Fig. 3 shows how the performance of a decoder based only on movement activity is increased through the addition of the plan activity of ten neurons. Because the plan activity maps directly to goal space,

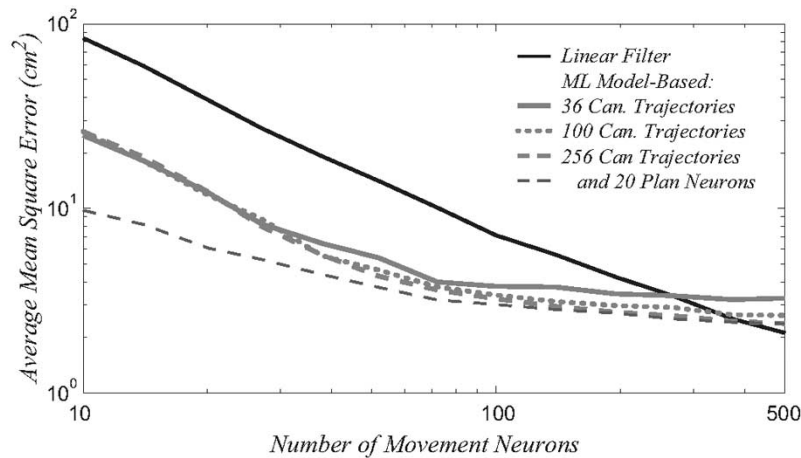


Fig. 4. Performance of a ML decoder is limited by the trajectory model underlying it. Depicted are the average mean square error for decoders of increasing complexity as the number of movement-tuned neurons increases.

with limited overall neural information the addition of these ten neurons decreases decoder error significantly. Specifically, notice that with the ten additional plan neurons, the number of movement neurons required for the 10 cm^2 level of accuracy drops to about 12.

The addition of plan activity, however, highlights the error floor that becomes apparent with increasing amounts of neural information. As the variation term, $\mathbf{n}(t)$ of (2), begins to overtake neural Poisson noise as the main source of reconstruction error, the ML framework of the model-based estimation begins to degrade. Recall that the variation term (\mathbf{n} in (2)) is inversely related to the efficacy of the canonical trajectory mapping. Thus, if the number of canonical trajectories is increased, the error floor should be lowered. This is apparent in Fig. 4, where the performance of ML decoders with 36, 100, and 256 canonical trajectories are compared. However, in the limited neural information case, fewer canonical trajectories provide less opportunity for reconstructing noise, hence we observe no increase in performance. Importantly, the addition of even a few (e.g., 10–20) plan neurons significantly improves performance in this regime. An alternate approach which avoids these problems is to form the problem in a Bayesian framework, allowing for the reconstructed trajectory to deviate from an expected canonical one [16]. In this work, however, as we were interested in the more common case of somewhat limited neural information (typically on the order of 100 neurons), further optimization was unnecessary.

V. DISCUSSION

While the results of the simulation appear promising, the practical implementation of a model-based neural prosthetic interface requires consideration of details that are partially avoided in the simulation framework.

A. Estimating Neural Tuning and Canonical Trajectory Models

As discussed above, the actual mapping between neural signals and movement is undoubtedly nonlinear. This clearly will have a negative effect on any system that lacks a perfect model for the mapping. In the model-based decoding paradigm, there

are three approaches to dealing with the problem of estimating neural tuning from data. First, using traditional techniques, an analytical linear or nonlinear mapping can be estimated, and simply used instead of the simple linear model used in this work. Second, the probability densities $\Pr(s_m(\mathbf{t})|T)$ or $\Pr(s_p(\mathbf{t})|\mathbf{x})$ can be directly estimated from data for each neuron. The ensuing result is an ensemble of neural firing templates corresponding to each canonical trajectory or endpoint. The final approach is to use an appropriate first generation inverse-mapping estimator to generate an initial trajectory estimate, and then refine it using a reach model in a mathematically similar procedure to that presented here.

A related problem is that the mapping may depend on the initial condition of the hand (i.e., initial posture). This is a more troubling issue due to the exponential difficulty of evaluating the neural-movement mapping for all possible postures, and remains an open problem. However, as long as the movement model is sufficiently accurate, a model-based system will always require fewer neurons than first generation inverse-mapping decoders to achieve an equivalent performance level.

One of the more interesting current problems in the field is designing the decoding system to enhance the adaptation ability of the neural circuits that interface it. In a situation in which two systems are simultaneously adapting to each other, there is the potential for either great gains in the rate of improvement or, alternatively, significant delays. While both canonical trajectory and neural tuning models could easily be made to adapt over time, we are still considering optimal methods for approaching the doubly adaptive system problem from the model-based decoding perspective.

In addition to neural signal mapping, the estimation of canonical trajectories can be difficult, especially if one wishes to expand the space of possible movements. Since a purely data-driven approach requires larger and larger data sets to learn more complex sets of movement, when a more general model is sought the best solution may be a synthesis of analytical and data-driven approaches. Perhaps the most rigorous approach would be a Bayesian method similar to that found in [16]. These types of modifications are especially necessary in situations in which the basic restriction of the model-based decoding paradigm—that movements are point-to-point reaches—is violated.

If, for instance, there is the potential for nonreach-like movements, or dramatically perturbed trajectories, a model-based decoder must be able to interpolate novel unusual movements from its basic family of canonical ones.

B. Computational Cost

The second aspect of the model-based decoding framework that was not addressed in our simulation was that of computational efficiency. In fact, one aspect in which the ML model-based approach is inferior to simple first-generation systems is in the amount of computational resources required. Computational complexity translates into a limiting factor for an implementation through the time required to generate an estimated trajectory. Ideally, a neural prosthetic interface should not introduce a delay into the users movements. Thus, its latency should be less than the approximate transit time of neural signals to the arm (about 100 ms), and its throughput (i.e., the number of neural channels that can be processed at a greater-than-real-time rate) should be larger than the number of available neurons. Using Matlab, we could decode a one second trajectory from the signals of an ensemble of 100 neurons in approximately 1 s. Previous experience suggests that the computation time of the ML algorithm can be reduced by a factor of ten using optimized C-code on a commercial off-the-shelf desktop computer, or a factor of 100 through the use of an ASIC, which would have the further advantage of being portable. Thus, we expect to use the system as described here for real-time prosthetic feedback in the same manner linear decoders are used. More fundamentally, however, in a prosthetic system, computational costs for increased performance are currently, and will most likely continue to be, much lower than costs involved with alternative methods for increasing performance, such as increasing the number of chronically implanted electrodes or training the user over a long period of time.

VI. CONCLUSION

We have shown that using a goal-directed model of reaching movements as a basis for a prosthetic decoder can result in significant performance increases over first generation inverse-mapping decoders. Additionally, this paradigm provides for straightforward integration of movement activity with target-tuned plan activity. Furthermore, the basic method of model-based decoding could be used to integrate as yet undiscovered types of neural activity, or even the local field potentials into an estimate of the instantaneous hand trajectory [19].

One of the main reasons for using the ML mechanism for generating the optimal model-based trajectory estimate in this work was its clarity and intuitive appeal. Its performance also proves to be competitive with the current alternatives, more than halving the number of neurons required for a given level of reconstruction error. While more development of the actual technique is necessary, we have demonstrated that decoding of goal-directed movements can and should take their nature into account. Therefore, we would suggest that this is the

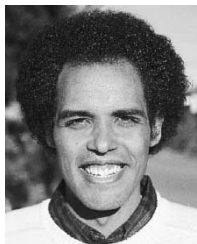
appropriate paradigm for extracting maximum benefit for patients with neural interfaces.

ACKNOWLEDGMENT

The authors would like to thank Dr. M. Churchland for providing the movement data we have used for this study. We would also like to thank G. Santhanam and B. Yu for their advice and assistance.

REFERENCES

- [1] D. Moran and A. Schwartz, "Motor cortical representation of speed and direction during reaching," *J. Neurophysiol.*, vol. 82, no. 5, pp. 2676–2692, Nov. 1999.
- [2] J. Messier and J. Kalaska, "Covariation of primate dorsal premotor cell activity with direction and amplitude during a memorized-delay reaching task," *J. Neurophysiol.*, vol. 84, no. 1, pp. 152–165, July 2000.
- [3] E. Todorov, "Direct cortical control of muscle activation in voluntary arm movements: a model," *Nature Neurosci.*, vol. 3, no. 4, pp. 391–398, Apr. 2000.
- [4] 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.
- [5] 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.
- [6] M. D. Serruya, N. G. Hatsopoulos, L. Paninski, M. R. Fellows, and J. P. Donoghue, "Instant neural control of a movement signal," *Nature*, vol. 416, no. 6877, pp. 141–142, Mar. 2002.
- [7] J. Carmena *et al.*, "Learning to control a brain-machine interface for reaching and grasping by primates," *PLoS Biol.*, vol. 1, no. 2, pp. 193–208, Nov. 2003.
- [8] K. V. Shenoy *et al.*, "Neural prosthetic control signals from plan activity," *Neuroreport*, vol. 14, no. 4, pp. 591–596, Mar. 2003.
- [9] D. Meeker, S. Cao, J. W. Burdick, and R. A. Andersen, "Rapid plasticity in the parietal reach region demonstrated with a brain-computer interface," presented at the *Soc. Neurosci. Conf. 357.7*, 2002, Abstr. Viewer and Itinerary Planner Program No. 357.7.
- [10] M. Serruya, N. Hatsopoulos, M. Fellows, L. Paninski, and J. Donoghue, "Robustness of neuroprosthetic decoding algorithms," *Biol. Cybern.*, vol. 88, no. 3, pp. 219–228, Mar. 2003.
- [11] M. S. A. Graziano, D. F. Cooke, C. S. R. Taylor, and T. Moore, "Distribution of hand location in monkeys during spontaneous behavior," *Exp. Brain Res.*, vol. 155, pp. 30–36, Mar. 2004.
- [12] B. Amirikian and A. Georgopoulos, "Directional tuning profiles of motor cortical cells," *Neurosci. Res.*, vol. 36, no. 1, pp. 73–79, Jan. 2000.
- [13] J. C. Sanchez *et al.*, "Input-output mapping performance of linear and nonlinear models for estimating hand trajectories from cortical neuronal firing patterns," in *Int. Workshop Neural Networks for Signal Processing*, Martigny, Switzerland, Sept. 2002, pp. 139–142.
- [14] 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 Neurosci.*, vol. 2, no. 7, pp. 664–670, July 1999.
- [15] W. Wu *et al.*, "Neural decoding of cursor motion using a kalman filter," in *Neural Inform. Processing Syst.*, 2003, vol. 15.
- [16] C. T. Kemere, M. Sahani, and T. H. Meng, "Robust neural decoding of reaching movements for prosthetic systems," in *Proc. IEEE Engineering in Medicine and Biology Society 25th Ann. Conf.(EMBS '03)*, Cancun, Mexico, Nov. 2003, pp. 2079–2082.
- [17] C. T. Kemere, G. Santhanam, B. M. Yu, K. V. Shenoy, and T. H. Meng, "Decoding of plan and peri-movement neural signals in prosthetic systems," in *Proc. IEEE Workshop Signal Processing Systems (SIPS '02)*, San Diego, CA, Oct. 2002, pp. 276–283.
- [18] C. Koch, *Biophysics of Computation*. New York: Oxford Univ. Press, 1999.
- [19] G. Santhanam, M. M. Churchland, M. Sahani, and K. V. Shenoy, "Local field potential activity varies with reach distance, direction, and speed in monkey pre-motor cortex," presented at the *Soc. Neurosci. Conf.*, New Orleans, LA, 2003.



Caleb Kemere (S'00) received the B.S. degree in electrical engineering (Hons.) in 1998 from the University of Maryland, College Park, and the M.S. degree in electrical engineering in 2000 from Stanford University, Stanford, CA, where he is currently working toward the Ph.D. degree in electrical engineering.

After joining Stanford University, he worked in the Space Systems Development Laboratory and the Magnetic Resonance Systems Research Laboratory. Following a brief stint with Datapath Systems (now part of LSI Logic), San Jose, CA, he returned to Stanford University. His research interests center on the development of novel signal-processing algorithms and architectures for use in neural interfaces.



Krishna V. Shenoy (S'87–M'01) received the B.S. degree in electrical engineering from the University of California at Irvine in 1990, and the M.S. and Ph.D. degrees in electrical engineering and Computer Science from the Massachusetts Institute of Technology, Cambridge, in 1992 and 1995, respectively.

He was a systems neuroscience postdoctoral Fellow in the Division of Biology at Caltech, Pasadena, CA, from 1995–2001. He joined the faculty at Stanford University, Stanford, CA, in 2001, where he is an Assistant Professor in the Department of Electrical Engineering and Neurosciences Program. His current research activities include neurophysiological investigations of sensorimotor integration and coordination, neural prosthetic system design, and neural signal processing and electronics.

Dr. Shenoy received the 1996 Hertz Foundation Doctoral Thesis Prize, a Burroughs Wellcome Fund Career Award in Biomedical Sciences, the William George Hoover Faculty Scholar in Electrical Engineering from Stanford University, the Robert N. Noyce Family Scholar from the Stanford University School of Engineering, an Alfred P. Sloan Research Fellow, and a Defense Science Research Council Fellow.



Teresa H. Meng (S'82–M'83–SM'93–F'99) received the Ph.D. degree in electrical engineering and computer science from the University of California at Berkeley in 1988.

She is currently the Reid Weaver Dennis Professor of Electrical Engineering at Stanford University, Stanford, CA. Her research activities during the first ten years at Stanford University included low-power circuit and system design, video signal processing, and wireless communications. In 1999, founded Atheros Communications, which delivers core technology for high-performance wireless communication systems.

She returned to Stanford University in 2000 to continue her research and teaching. Her current research interests focus on circuit optimization, neural signal processing, and computation architectures for future scaled CMOS technology. She has given plenary talks at major conferences in the areas of signal processing and wireless communications. She is the author of one book, several book chapters, and over 200 technical articles in journals and conferences.

Dr. Meng has received numerous awards and honors for her research work at Stanford University, including a Presidential Young Investigator Award, an ONR Young Investigator Award, an IBM Faculty Development Award, a Best Paper Award from the IEEE Signal Processing Society, the Eli Jury Award from the University of California at Berkeley, and awards from AT&T, Okawa Foundation, and other industry and academic organizations. As a result of her work as founder of Atheros Communications, she was named one of the Top 10 Entrepreneurs in 2001 by Red Herring, Innovator of the Year in 2002 by MIT Sloan School eBA, and received the CIO 20/20 Vision Award in 2002.