DECODING OF PLAN AND PERI-MOVEMENT NEURAL SIGNALS IN PROSTHETIC SYSTEMS

Caleb T. Kemere[†*], *Gopal Santhanam*[†], *Byron M. Yu*[†], *Krishna V. Shenoy*[†§], *and Teresa H. Meng*[†]

Stanford University
[†]Department of Electrical Engineering
[§]Neurosciences Program
Stanford CA, 94305

ABSTRACT

In this paper we introduce a theoretical framework for improved processing of peri-movement neural activity for neurally controlled prosthetic systems through maximum likelihood sequence estimation. This framework further suggests a computational method for integrating plan and peri-movement neural activity. We show that combining plan activity, usually associated with target specification, with peri-movement neural activity yields more accurate estimates of the *trajectory* of an arm movement. The effectiveness of the method is demonstrated in simulation. Performance as a function of the specific number of plan and peri-movement neurons, as well as other system and design parameters is analyzed. The algorithm presented is also compared against previous, sample-based approaches, specifically a "point-process" filter for plan activity and a standard linear filter framework in the peri-movement regime.

1. INTRODUCTION

An exciting emerging field of signal processing is the decoding of neural signals drawn directly from the brain. One of the goals pursued by researchers in this field is to restore function to patients with paralyzed limbs through direct interface with the brain. To achieve this end, a signal processing system must be developed which correctly decodes neural activity. In this study, using simulated data from a reasonable model of the brain, we perform an equivalent task, reconstructing arm movements from their corresponding neural control signals.

A typical approach to investigating neural coding of motor control has been to use microelectrodes to record the activity of an ensemble of neurons while also recording the related arm movements ([1], [2], [3]). Neural activity immediately preceding or simultaneous with arm movements is termed "peri-movement." These signals are often highly correlated with electrically measured muscle activity, suggesting they correspond to muscle control signals. In some brain regions, there is also neural activity long before, or even without, actual movement. This is termed "plan" activity because of its association with intended movements. In the motor and pre-motor cortical regions of the brain, it is common to find neural activity of both types. Thus, in a situation where it is not possible to pre-select the type of neural activity, it is desirable to consider the optimal use of the data gathered, whether plan or peri-movement or both.

*Please address correspondances to ckemere@stanford.edu.

One might anticipate that combining plan activity with peri-movement activity would improve the accuracy of the reconstructed end-point of the movement since the plan activity provides additional information as to where the movement should come to rest. This study, however, provides an improved structure for decoding peri-movement neural activity and a framework for joint decoding of the two types of neural signals, both of which improve the accuracy of reconstructed movement trajectories. Simulation results for certain classes of movements provide an estimate of system performance, and yield insight into the sensitivity to various design parameters.

2. MODELS

2.1. Movement Model

During quick reaching movements, the hand travels to its target in stereotyped trajectories. While it is clear that having a target enforces a constraint on the trajectory taken, reasons for this stereotyping are not obvious. In trying to understand the brain's control algorithms, various constraints, e.g. minimizing jerk, torque change, transit time, have been proposed, and a recent unifying result suggests that the brain optimizes noisy force signals in order to minimize end-point error [4].

To ease this analysis, we simplify the somewhat complex reaching movements observed in nature. First, they are restricted to two dimensions – as when a hand moves on the surface of a touch screen. Second, movements are fully specified by their endpoint (i.e. curved trajectories and multiple speeds are precluded). Finally, trajectories are simplified to have the shape resulting from minimizing the "jerk" (time derivative of acceleration) of the movement [5]. This form is given as

$$\mathbf{x}(\mathbf{x}_f, t) = \mathbf{x}_f \cdot \left(6\left(\frac{t}{t_f}\right)^5 - 15\left(\frac{t}{t_f}\right)^4 + 10\left(\frac{t}{t_f}\right)^3 \right) \quad (1)$$

where $\mathbf{x}_f$ is the target location relative to the origin and t_f is the duration of movement, which is further constrained by a smoothness parameter, S, as below.

$$t_f = (60\|\mathbf{x}_f\|)^{\frac{1}{3}} S \quad (2)$$

The horizontal components of three sample arm trajectories and their corresponding time derivatives are shown in Figure 1.

0-7803-7587-4/02/$17.00 ©2002 IEEE

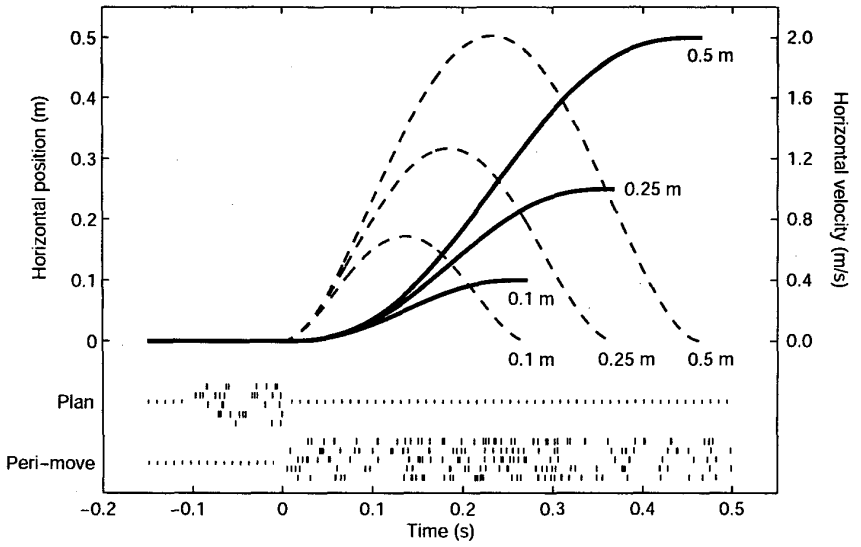

Fig. 1. Arm movement trajectories for reaches to three different end points. Movement trajectories are stereotyped with roughly sigmoidal position curves (solid lines) and roughly Gaussian velocity curves (dashed lines). Movement trajectories were generated using Equations (1) and (2). Also shown are responses of two neurons, one a plan neuron and one a peri-movement neuron associated with a 0.5 m rightward arm movement. Five representative repetitions, or trials, appear for each neuron with a vertical line indicating the time the simulated neuron emitted an action potential. The dotted line indicates when neural activity was not simulated since it was not needed for the algorithm. Note that plan neural activity is present throughout the simulated plan period (-100 to 0 ms) and the peri-movement neural activity increases and decreases according to the current arm movement velocity (rightward peri-movement preferred directions chosen to ensure positively correlated firing). Spike trains were generated with Equations (5) and (6).

2.2. Neural Signal Model

In experimental neurophysiology the standard technique is to record the time that a neuron emits a stereotypical electrical pulse, referred to as an action potential or "spike." The resulting data constitute a point process time series. The bottom panel of Figure 1 depicts data that might be gathered from two types of neurons during repeated reaches following the longest trajectory (0.5 m rightward reach) in the figure. It has been shown that modeling neurons as firing randomly in time as an inhomogeneous Poisson point process captures most of the statistical variation of neural firing [6]. Thus, the distribution of the number of action potentials, k, observed within a time window of duration T is given by

$$p(k) = \frac{1}{k!}\left(\int_t^{t+T} \lambda(\tau)d\tau\right)^k e^{-\int_t^{t+T}\lambda(\tau)d\tau} \quad (3)$$

$$= \frac{(f_T)^k}{k!}e^{-f_T} \quad (4)$$

where f_T is the integral of $\lambda(t)$, the instantaneous rate of the process, over the time window. In our model, the instantaneous rate encodes the parameters of interest, namely arm velocity or target location.

The variation of the rate at which a neuron produces spikes as a function of some external parameter is known as its "tuning." In some cases of interest, the parameter may represent a system state variable, for example a planned target. In these cases, the tuning is constant over some period of interest, and the decoding problem reduces to estimating the constant variable of interest from an observed time series of spikes. Alternatively, the tuning may vary with time, as, for example, when it is correlated with movement forces. In this case, the decoding problem is to estimate the time varying movement that was to be generated by the neural signal.

The tuning of the subclass of neurons that are involved in planning movements appears to be roughly constant over an interval during which the subject prepares to move. While the tuning of plan neurons has not been as extensively investigated as that of movement neurons (described below), it has been shown that the tuning varies with direction and extent of movement ([7], [8]). For simplicity, we model the tuning as Gaussian, where the firing of the neuron decreases radially from a preferred location. The functional form is given as

$$f_{planner}(\mathbf{x}_f) = T_{plan}\lambda_{max}\exp\left(-\frac{\|\mathbf{x}_f - \mathbf{u}_{preferred}\|^2}{2\sigma^2}\right) \quad (5)$$

where $f_{planner}$ is the mean of the Poisson process over the duration of the plan interval T_{plan}, λ_{max} specifies the maximum firing rate of the neurons, σ the standard deviation of the tuning, and $\mathbf{u}_{preferred}$ the location of maximal firing. It is interesting to note that, unlike the other parameters, which describe biological phenomena, the duration of the plan interval is variable at the system level. Appropriate assumptions allow λ_{max} and σ to be specified for our simulations – the same values are taken for every neuron in our population; $\mathbf{u}$ is randomly chosen within the workspace for each neuron.

The tuning of motor cortical neurons is a matter of some controversy among researchers in the field. It has been argued that these neurons fire proportionally to many variables, including hand velocity, hand force, and muscle forces within the arm [9]. However, under many circumstances, observed firing rates vary with the cosine of the angle between hand velocity and some preferred direction. The rates also scale linearly with hand speed. This model for peri-movement neural activity is often dubbed "cosine-tuning" [10]. Note that unlike the plan neurons, the firing of the peri-movement neurons is time-varying. After digital sampling, the mathematical representation of the sampled Poisson process mean, f_{mover}, is

$$f_{mover}(\mathbf{x}_f, n) = \Delta_t \frac{\lambda_{max} - \lambda_{min}}{2} \cdot \left(\frac{\hat{\mathbf{e}}_{preferred} \cdot \dot{\mathbf{x}}(\mathbf{x}_f, n)}{\|\mathbf{v}_{max}\|} + 1 \right) + \Delta_t \lambda_{min} \quad (6)$$

where Δ_t is the time quantization, λ_{min} specifies the minimum firing and $\dot{\mathbf{x}}(\mathbf{x}_f, n)$ is the average velocity of the trajectory over $[n\Delta_t, (n+1)\Delta_t)$, as given by the time derivative of Equation (1). Finally, $\hat{\mathbf{e}}_{preferred}$ 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 for peri-movement neurons. Studies of peri-movement neural activity in the motor cortex have shown that the directional tuning can be better modeled with more complex functional forms [11], but cosine tuning appears to capture much of the complexity.

3. DECODING

3.1. Prior Approaches

Previous work in decoding the neural activity associated with arm movements has focused primarily on the peri-movement neural signals. Several approaches have been taken; the most popular ones estimate the velocity (or position) of the arm from affine combinations of the observed firing of the neurons during windows in time [2]. For comparison with the algorithm presented here, we use the minimum mean-square error filter derived from the preferred directions of the peri-movement neurons. If one rewrites Equation (6) for neuron i as

$$f_i(n) = A\hat{\mathbf{e}}_i \cdot \mathbf{v}(n) + B \quad (7)$$

then, given the observed firing of N neurons concatenated into a column vector $\mathbf{f}$, the standard linear unbiased estimator for $\mathbf{v}$ (the time derivative of Equation 1) is given by

$$\hat{\mathbf{v}} = \frac{1}{A}(\mathbf{E}^T\mathbf{E})^{-1}\mathbf{E}^T(\mathbf{f} - B) \quad (8)$$

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. A strength and weakness of this type of algorithm is that it is agnostic to stereotyping of arm movements, such as those observed in nature and described above. Therefore, it generates an estimate based only on the currently observed (or in more complicated versions, nearby) samples of data.

For plan activity one promising sample-based algorithm is the "point-process" filter [12]. This algorithm closely resembles the philosophy embodied in the well-known Kalman filtering framework. The point-process filter uses a recursive algorithm, with

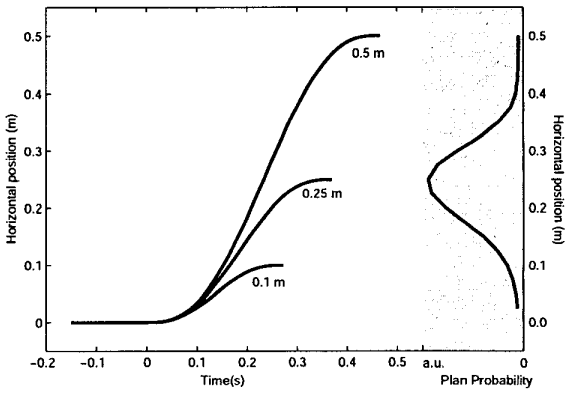

Fig. 2. Neural plan activity imposes an a priori distribution on movement trajectory. Three movement trajectories, as shown in Figure 1, appear to the left. Neural plan activity can be used to compute a probability distribution function for where the reach will end (curve to the right). The probability distribution is maximum for a 0.25 meter rightward reach, thus at each time step, the point along the trajectory marked 0.25 m is more likely to be correct than the points corresponding to that time step on the other two trajectories.

procedures similar to the Kalman time and measurement updates, to incorporate the last sample estimate with the spike data from the current time point. In this particular case, the time update is tied to a model in which the next movement *increment* is stochastically distributed as a 2-dimensional Gaussian centered at the past estimate. Furthermore, the measurement update takes into account the point-process nature of the Poisson model of the neurons. This filter design is derived for Gaussian tuned neurons; hence, we have studied its utility in the plan period. For more summary information regarding this approach, please see the Appendix.

Additional sample-based strategies for Gaussian neural activity have been investigated including non-linear (Bayesian, neural network based, etc.) sample-based decoders, which in some cases perform better than the linear ones. One comprehensive review of linear filters and non-linear maximum likelihood techniques is found in [13].

3.2. Maximum Likelihood

The previous observation that certain classes of movements are stereotyped suggests that greater accuracy may be achieved by holistically treating the neural firing as a temporal sequence of values specified by the endpoint of the movement rather than isolated samples. For peri-movement neurons, we can write the log-likelihood of arm position at any time as

$$LL(\mathbf{x}_f, n) = \log \left[\prod_{j=1}^{M} \prod_{\tau=1}^{n} p(k_{mover,j}(\tau) | \mathbf{x}_f) \right] \quad (9)$$

where $k_{mover,j}$ is the number of spikes observed from peri-movement cell j and $p(\ldots)$ is found by substituting Equation (6) into (4). Due to the assumption that endpoints fully specify trajec-

tories, the maximum likelihood estimate is

$$\hat{\mathbf{x}}_{est}(n) = \mathbf{x}\left(\underset{\mathbf{x}_f}{\text{argmax}}\left[LL(\mathbf{x}_f, n)\right], n\Delta_t\right) \quad (10)$$

where $\mathbf{x}(\mathbf{x}_f, t)$, the trajectory inverse function that maps from an endpoint to the point along the trajectory at time t, is found in Equation (1). Equations (9) and (10) illustrate that the estimate of the current arm position is generated by evaluating which of a family of arm trajectories – indexed by the movement endpoint – best fit the current data, and then choosing the current position of *that* trajectory for the current estimate.

Furthermore, if the data presented to the decoding system is composed of both plan and peri-movement neural activity, the integration of the plan activity is seamless. As illustrated in Figure 2, because of our stereotypical movement assumptions, plan activity (right panel), which is tuned for the endpoint of a movement, effects an a priori distribution on the possible trajectories (left panel) that may be decoded from peri-movement neural activity. The new log likelihood function is

$$LL_{full}(\mathbf{x}_f, n) = \log\left[\prod_{i=1}^{P} p(k_{planner,i}|\mathbf{x}_f)\right] + LL_{mover} \quad (11)$$

where $k_{planner,i}$ is the number of spikes observed from plan cell i and the likelihood surface corresponding to this neural activity has been added to the log-likelihood of Equation (9). To evaluate the maximum likelihood, Equation (11) is substituted into (10).

By using small time windows, the analysis can be simplified. In the short interval limit, a Poisson process becomes a Bernoulli process, i.e. produces only zero or one as an outcome. Inserting the probability distribution in this case yields the following likelihood function.

$$\begin{aligned} LL(\mathbf{x}_f, t) &= C \\ &+ \sum_{i=1}^{P} \left(k_i \log\left[f_{planner,i}(\mathbf{x}_f)\right] - f_{planner,i}(\mathbf{x}_f)\right) \\ &+ \sum_{\tau=1}^{n}\sum_{j=1}^{M} \left(I_j(\tau) \log\left[f_{mover,j}(\mathbf{x}_f, \tau)\right] - f_{mover,j}(\mathbf{x}_f, \tau)\right) \end{aligned} \quad (12)$$

where for neuron i, k_i is the number of spikes observed, and f_i is the tuning as a function of target location (given in Equations (5) and (6)), I is an indicator function for the firing of a cell, and P and M are the numbers of plan and peri-movement neurons, respectively. Because there is no closed form solution to this maximization problem, the actual solution is approximated in our simulations through an exhaustive search through discretized space.

4. RESULTS

4.1. Simulation Architecture

Neural signals were generated for movements to targets chosen at random in a unit square (arbitrary units consistent throughout simulations) centered on zero as in Figure 3. As discussed above, discretizing the number of potential targets into a grid of endpoints allows for simplified calculation of the maximum likelihood. As decoding errors decrease, the estimated endpoints begin to snap to the grid, causing an abnormal acceleration in the performance of the algorithm. For most data in the study, we avoided this issue by utilizing a grid of 1600 points. Neuron parameters (e.g. preferred locations for plan neurons, preferred directions for peri-movement neurons) were randomized with the decoding process typically repeated over at least 20 sets. For each set of parameters, at least 200 random targets were typically selected. Reaches to these endpoints were constructed by Equations (1) and (2) where the smoothness parameter, S, was chosen such that a reach to the farthest target in the grid took 0.5 seconds. Random neural firing data were generated with a time quantization of 1 millisecond using the methods and probability distributions described previously.

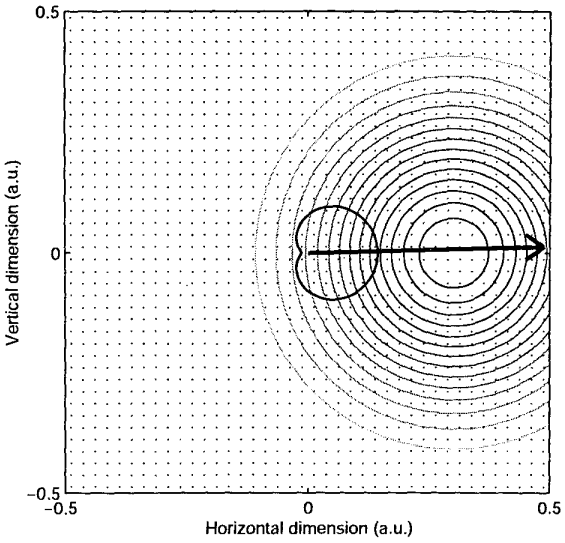

Fig. 3. Workspace used in computer simulations. Reaching arm movements were simulated from the center of the workspace to each of the 1600 possible endpoints (array of points). The arrow depicts a rightward and slightly upward reach. The workspace is square and is one arbitrary unit (a.u.) on a side. All distance parameters are consistently measured against this unit. A plan neuron 2-dimensional Gaussian receptive field centered at 0.3 a.u. to the right and is shown as a family of iso-intensity rings. The cosine tuning of a peri-movement neuron, centered in the workspace with a rightward preferred direction, is also shown (solid line). Note that the plan neuron receptive field is drawn to scale while only the shape, not size, of the peri-move neuron movement field has meaning since arm movement velocity modulates the response.

4.2. Simulation Results

For unbiased estimators, the variances of estimates from independent observations add inversely. In this particular case, one can think of each neuron as providing an independent observation. Thus, in the biologically plausible range of parameters, we would expect an error model such as

$$\begin{aligned} &\frac{1}{E[(\hat{\mathbf{x}} - \mathbf{x})^2]} \propto \\ &\frac{1}{\sigma^2_{plan}} N_{plan} + \frac{1}{\sigma^2_{move}} N_{move} - C(N_{plan}, N_{move}) \end{aligned} \quad (13)$$

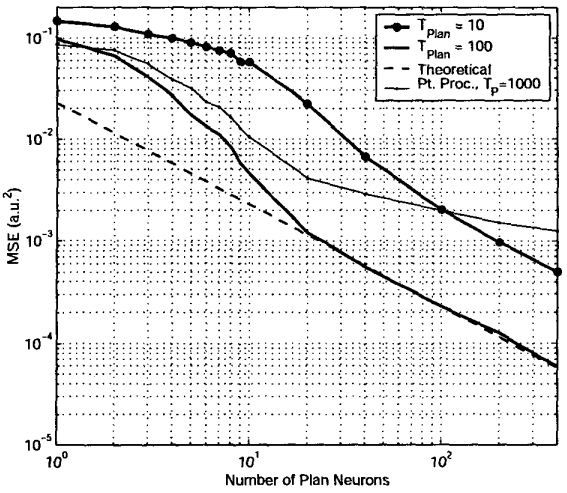

Fig. 4. Decoded trajectory error as a function of the number of plan neurons. At least 100 random endpoints were chosen for each of at least 20 randomly parameterized ensembles of neurons (σ = 0.2 a.u., λ_{max} = 100 and λ_{min} = 10 spikes/second, consistently throughout this study). The dotted line depicts a fit of the many neuron limit of system performance. The thin line shows the performance of a point process filter on the same data (random walk variance = 0.001 a.u.^{2}).

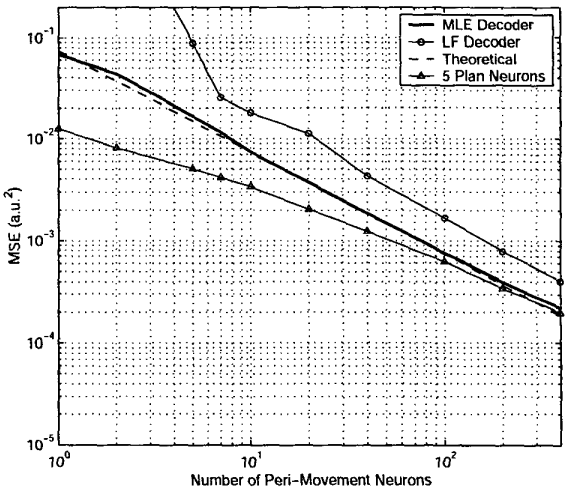

Fig. 5. Decoded trajectory error as a function of the number of peri-movement neurons. The performance of the maximum likelihood decoder is compared with the theoretical bound, a linear filter, and a system in which the decoder uses not only peri-movement neural activity, but also information from 10 plan neurons.

where N_{plan} and N_{move} are the number of plan neurons and peri-movement neurons, respectively, and σ^2_{plan} and σ^2_{move} represent the contribution of a single neuron to the mean square estimation error. The final term represents the non-linear portion of the error (discussed below). It can be shown that the variance of the maximum-likelihood estimate of the parameter of a Poisson process varies inversely with the length of the estimation window [14]. Thus, it is expected that the single-neuron variance of plan neurons will be inversely proportional to the duration of plan interval.

For small numbers of neurons chosen randomly, the typical distribution of neurons in the workspace (preferred locations or directions) will be non-uniform (e.g. the preferred directions will be closer to each other than to orthogonal, or the preferred locations will be unbalanced in workspace coverage). The result is higher than expected error. This is the source of the $C(\ldots)$ term in Equation (13). The performance of the prosthetic system with limited numbers of neurons is of special interest, since current instrumentation only permits interfacing with small neural populations (10s-100s of cells). Furthermore, the performance of systems controlled by even small numbers of neuron may be further enhanced by the brain's ability to adapt through time [1].

Figure 4 shows simulation results as the number of plan neurons in the system increases. The error metric is the trajectory error measured as the square distance between estimated and actual hand positions averaged over the movement time. Notice that the error performance is well approximated by Equation (13), not only by decreasing inversely to neuron count, but also by scaling inversely with the length of the plan interval. If we take the value for 200 neurons as characteristic of the many neuron limit, the data suggest that σ^2_{plan} is approximately 0.0025 a.u.^{2} sec – i.e. 0.25 a.u.^{2} for 10 msec plan or 0.025 a.u.^{2} for 100 msec plan. We found that, in the many neuron limit, this value was inversely related to the tuning width of the neurons. For example, for a standard deviation of 0.4 a.u.^{2}, twice that used in our simulations, the per-neuron variance was measured as 0.0051 a.u.^{2} sec. For a standard deviation of 0.1 a.u.^{2}, the per-neuron variance was measured as 0.0021 a.u.^{2} sec.

The plan interval parameter (T_{plan} in Equation (5)) is important to system designers since it can be used to reduce the contribution of the planner neurons to overall estimator variance. As seen in Equation (13), the planner population variance can be reduced in two ways: by decreasing σ^2_{plan} or increasing the number of neurons (N_{plan}). The former can be achieved by increasing T_{plan}; training the user to extend the period during which a movement is planned. The number of neurons interfaced cannot be easily increased after a fixed number of electrodes have been implanted in the subject.

Figure 5 depicts the dependence of trajectory error on the number of peri-movement neurons. As expected, the inverse relationship of Equation (13) holds. In this case, σ^2_{move} is approximately 0.076 a.u.^{2}. Thus, in the limit of many neurons, the information gained from a plan neuron with a plan interval of about 30 ms is equivalent to that gained from a peri-movement neuron. Also shown is a plot of the performance of a system in which the activity from 5 plan neurons is integrated with peri-movement activity. As expected, for small numbers of peri-movement neurons, trajectory error is significantly reduced by the addition of plan activity.

A key difference between the error performance of plan- and peri-movement-based decoding is in their convergence characteristics. The $C(\ldots)$ term of Equation (13) represents the greater

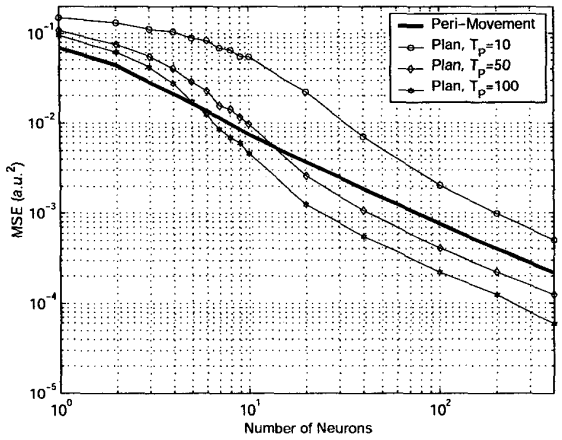

Fig. 6. Plan- and peri-movement-only decoder performance. Notice different error convergence characteristics of decoders using only plan or peri-movement neural activity. In the limit, only neuron parameters and the number of neurons matter.

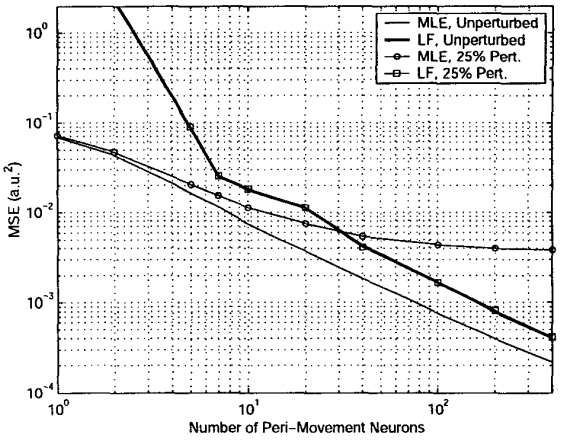

Fig. 7. ML and linear filter decoder performance with perturbed movements. Trajectories were perturbed in the velocity domain by one cycle of a sine wave in one of four randomly chosen directions. Maximum perturbation is 25% of the maximum velocity.

error that occurs when there are only a small number of neurons. As the number of plan and peri- movement neurons increases, $C(\ldots)$ tends to zero. As seen in Figure 5, in peri-movement neurons, when there are more than two, each neuron contributes nearly its full amount of information. Hence, the error for the peri-movement neurons is linear throughout nearly the whole regime of neuron densities.

The error convergence of plan neurons is closely related to the size of workspace area in which the neurons provide significant signal differentiation. Thus, unlike the broadly tuned peri-movement neurons, for the tuning widths used in this study, the system error does not converge immediately to its many neuron limit. However, this effect is significantly affected by the tuning width of the neurons. Intuitively, neurons with wide tuning are less specific, hence their limiting variance is higher than those with narrower tuning. For the same reason, for smaller numbers of neurons, those with wide tuning cover more of the workspace, and thus the error converges to the many neuron limit more quickly. Comparing the limiting cases, for infinitely wide tuning the number of neurons has no effect on the error; for infinitely narrow tuning, an infinite number of neurons is needed to decode reaches in a continuous workspace.

Figure 6 depicts three regimes of operation for systems composed of plan and peri-movement neurons. When there are few neurons, peri-movement neurons provide more per-neuron estimating accuracy than plan neurons. This is regime 1 – roughly 1–10 neurons in Figure 6. When there are a large number of neurons, both the peri-movement and plan neurons will cover the entire workspace well. Thus, comparison between the per-neuron variance of plan and peri-movement neurons can be done solely on the basis of system parameters. As depicted, a system based only on plan or peri-movement neurons will provide higher decoder accuracy depending on whether σ_{plan} or σ_{move} is lower. This is regime 3 – roughly 10 or more neurons in Figure 6. When the per-neuron variances are comparable (regime 2 – around 10 neurons in Figure 6), the exact distribution of neuron centers and preferred directions will heavily influence system performance.

In practice, arm movements do not exactly follow prescribed paths. In fact, ballistic trajectories (e.g. reaching for a glass) are often modified to avoid obstacles. Thus, to test the significance of the assumption of stereotyped trajectories, the performance of the system was measured in an environment where arm movements occurred close to the same path, but were perturbed by a sinusoidal velocity term in a random orientation. Figure 7 compares the performance of the linear filter with the maximum likelihood model in this situation. As might be expected, the maximum likelihood estimate, which is constrained to selecting from a family of trajectories that does not contain the real one, has a rather significant error floor. Future work will seek to combine the advantages of maximum likelihood (or point-process) for plan periods with the flexibility that linear filters provide for movement estimation.

Furthermore, it is quite possible that an individual cannot, or does not, steadily plan on a stationary target for the entire plan period. For example, one can imagine a scenario where a subject is tracking a dancing target before deciding to reach to it. The maximum likelihood algorithm will suffer in performance since it inherently assumes that the plan will be static in its sequence based architecture. On the other hand, a sample based approach can potentially perform better in this arena. To simulate this type of variation, the plan was modeled as the baseline firing rate previously described along with a sinusoidal spatial perturbation component. Figure 8 compares the performance of the point-process filter with the maximum likelihood under this condition. As expected, with tests of simple oscillatory plans, the point-process filter produces higher accuracy trajectories than the maximum likelihood planning filter. In this case, the maximum likelihood decoder has an error floor related to the relationship between the size of the plan interval and the frequency of the variation. One benefit of the point process filter structure is that it produces an estimate variance that could be used to generate a likelihood surface in a similar vein to

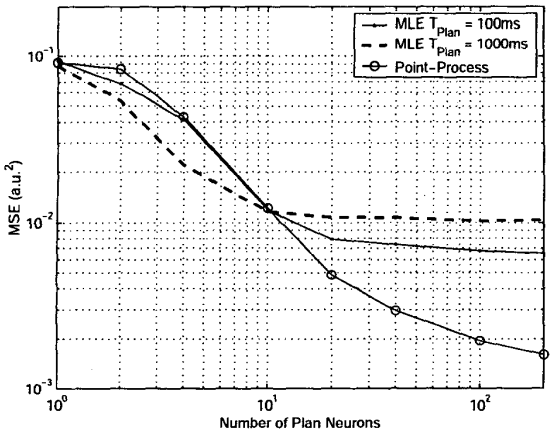

Fig. 8. ML and point-process filter decoder performance with dynamic plan activity. The planned endpoint was perturbed in a random direction by a cosine wave biased so that the actual endpoint occurred at the minimum of the perturbation, fixed to occur at the end of the plan period. Results are given for a perturbation with magnitude 0.4 a.u., and Frequency 4 Hz.

the maximum likelihood estimate. This potential avenue for integration with peri-movement data merits further investigation.

5. CONCLUSION

We have presented a mathematical framework for improved decoding of peri-movement neural activity in the case of stereotyped arm movements. It also enables plan neural activity, related to target location, to be integrated in such a way that it improves system performance along the trajectory of arm movements. A maximum likelihood decoder has been simulated and its performance as a function of the number of neurons in the system was presented. Decoder system performance has also been discussed as a function of system parameters such as the duration of the plan interval and the tuning width of plan neurons. Finally, the simulation assumptions of fully stereotyped movements and static plan activity were relaxed and the performance of the maximum likelihood decoder was compared to previous methods in less constrained circumstances.

One model of movement control in the brain is that peri-movement activity results from optimizing potential trajectories subject to reaching the location encoded by the plan activity. If the only movements considered are those that follow stereotypical trajectories, the decoding system we have presented hitherto decreases error by reproducing this optimization external to the brain. Though movements in nature actually display a much richer variety than those studied in this work, the framework presented can be expanded as the control algorithms utilized by the brain are further understood. Thus, by combining plan and peri-movement neural activity as presented here, it should be possible to use these signals regardless of activity type, to more accurately estimate arm movements and hereby control prosthetic devices with fewer implanted electrodes.

6. ACKNOWLEDGMENTS

The authors would like to thank Dr. Mark Churchland for his helpful advice regarding the effects of grid density on system performance. Furthermore, Figure 6 might aptly be termed a "Churchland" plot. This work is supported by MARCO contract #2001-CT-888, DARPA Contract #Mda972-02-1-0004, the Center for Integrated Systems, Stanford University, the Sloan Foundation, the Burroughs Wellcome Fund, and the NSF Center for Neuromorphic Systems Engineering at Caltech.

7. REFERENCES

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

[2] 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.

[3] J.K. Chapin et al., "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. Harris and D. Wolpert, "Signal-dependent noise determines motor planning," *Nature*, vol. 394, pp. 780–784, Aug. 1998.

[5] B. Hoff, "A model of duration in normal and perturbed reaching movement," *Biological Cybernetics*, vol. 71, no. 6, pp. 481–488, Oct. 1994.

[6] C. Koch, *Biophysics of Computation*, Oxford University Press, 1999.

[7] A. P. Batista and R. A. Andersen, "The parietal reach region codes the next planned movement in a sequential reach task," *Journal of Neurophysiology*, vol. 85, no. 2, pp. 539–544, Feb. 2001.

[8] J. Messier and J. Kalaska, "Covariation of primate dorsal premotor cell activity with direction and amplitude during a memorized-delay reaching task," *J. Neurophysiology*, vol. 84, no. 1, pp. 152–165, July 2000.

[9] E. Todorov, "Direct cortical control of muscle activation in voluntary arm movements: a model," *Nature Neuroscience*, vol. 3, no. 4, pp. 391–398, April 2000.

[10] A.P. Georgopoulos, A.B. Schwartz, and R.E. Kettner, "Neuronal population coding of movement direction," *Science*, vol. 233, no. 4771, pp. 1416–1419, Sept. 1986.

[11] B. Amirikian and A. Georgopoulos, "Directional tuning profiles of motor cortical cells," *Neuroscience Res.*, vol. 36, no. 1, pp. 73–79, Jan. 2000.

[12] E. Brown et al., "A statistical paradigm for neural spike train decoding applied to position prediction from the ensemble firing patterns of rat hippocampal place cells," *J. Neuroscience*, vol. 18, no. 18, pp. 7411–7425, Sept. 1998.

[13] K. Zhang et al., "Interpreting neuronal population activity by reconstruction: Unified framework with application to hippocampal place cells," *J. Neurophysiology*, vol. 79, no. 2, pp. 1017–1044, Feb. 1998.

[14] N. Twum-Danso and R. Brockett, "Trajectory estimation from place cell data," *Neural Networks*, vol. 14, no. 6–7, pp. 835–844, July 2001.

8. APPENDIX

The derivation of the point-process filter is covered in detail by [12]. One of the main advantages of this filter is that it produces an estimate of the plan position as well as a variance in that estimate. The error in the estimate can be approximated as Gaussian distributed around the estimated position. The Gaussian distribution can then be substituted for the maximum likelihood surface that would have been originally produced at the end of the plan period. Equations (14)–(16) provide a flavor for the one-step prediction phase of the point-process filter (a la the Kalman time update). The posterior mode $\hat{x}(t_k|t_k)$ and posterior variance $W(t_k|t_k)$ equations – analogous to the Kalman measurement update – are not included here. These and further derivations can be found in [12].

$$x(t_k) - x(t_{k-1}) \sim N(0, W_x(\Delta_k)); \quad (14)$$

$$\hat{x}(t_k|t_{k-1}) = \hat{x}(t_{k-1}|t_{k-1}); \quad (15)$$

$$W(t_k|t_{k-1}) = W_x(\Delta_k) + W(t_{k-1}|t_{k-1}); \quad (16)$$

The point-process filter relies on a continuity assumption (Equation (14)) so as to constrain the magnitude of each spatial increment. This continuity constraint can be thought of as the distribution of velocities at which a subject can move his plan. There are very little data on the rate at which plans in cerebral cortex can change. However, it is possible to make estimates of this quantity using data from [7], specifically the slew rate of the planning neurons. In this way, there is an attempt to (loosely) tie a biological variable to the algorithm. Of course, further physiological experiments along this path could be rather enlightening.

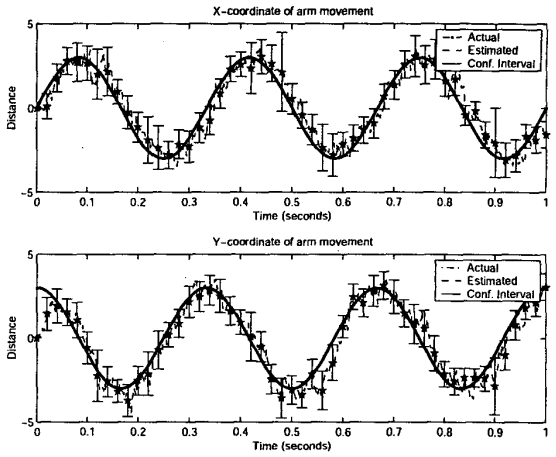

Fig. 9. Tracking Sinusoidal Dynamic Plan (Point-Process Filter)

Lastly, Figure 9 shows one observation of how the point-process filter might estimate the dynamic endpoint during a single plan period. The plan is oscillating quickly and widely over a large distance and this algorithm performs reasonably in tracking the change.