

# Controlling a Neuroprosthetic Arm: Real Time Estimation and Prediction

Cari Kaufman and Valérie Ventura

July 7, 2006

## Abstract

The development of neuroprosthetic devices promises to allow previously immobile patients to control the movement of an external device using only their brains' electrical activity. Prediction algorithms used to control such devices rely on models relating the firing rates of a population of neurons to intended movement variables such as direction. However, since no data on real arm movement will be available prior to use of the prosthetic, and recent research has shown that neurons may change their firing patterns in response to visual feedback, an algorithm is needed that in addition to predicting movement can also perform real time estimation of the model. This article proposes statistical methods for performing these related tasks and demonstrates the methods' effectiveness using data taken from a series of experiments using a rhesus monkey.

## 1 Introduction

A neuroprosthetic device is a type of brain-computer interface (BCI) that uses the brain's electrical signals to control the movement of a mechanical device such as a robot arm. Until recently, most BCIs have used electroencephalogram (EEG) recordings taken from electrodes held against a human scalp [15]. However, these noninvasive techniques do not provide the type of detailed signal recording needed to be able to predict movement. Recently, small microelectrode arrays implanted directly into the brains of live monkeys have allowed the recording of many individual neurons simultaneously, making the development of neuroprosthetic devices more feasible [9].

In order to allow the user of a neuroprosthetic device to control its movement, one needs to design an algorithm which at any given time has as its input the firing rates of a population of neurons and has as its output the movement the prosthetic should make, in terms of direction, speed, acceleration, or other variables. As a statistical problem, this means (i) determining an appropriate model for the the relationship between the user's intended movement at a given time and the associated firing rates, (ii) estimating the parameters of this model, and (iii) using the fitted model, along with the current firing rates, to

determine how the prosthetic should move. Generally, model selection, task (i), may be a time consuming process and is probably best undertaken before the prosthetic is in use. Movement prediction, task (iii), needs to be done repeatedly in order to control the prosthetic, and it must be done fast and often enough that the movement can mimic natural movement. When the user of the prosthetic doesn't have the use of his or her own arm, no prior data on actual arm movement will be available to carry out step (ii), parameter estimation. Instead, this also needs to be done while the prosthetic is in use, although not necessarily as frequently as the movement prediction. Some further reasons for preferring a real time estimation method are given in the next section. Note also that the model selection and parameter estimation steps require knowing what movements the user intended. This can be achieved by using a "calibration" period in which the user attempts to make a series of specific movements. The tasks performed by the monkey in the experiments described in section 2 are of this nature.

### 1.1 The Need for Real Time Estimation Methods

Previous algorithms have used data taken from experiments in which real arm movements and neuronal firing rates were recorded simultaneously [5, 16]. Using some initial period of time as "training data," researchers fit various models which are then used to predict the trajectory of the arm during the rest of the experiment. The predicted trajectories are compared to the actual trajectories in order to assess the quality of the statistical methods used.

There are several problems with this approach. The most obvious one is clearly that such "training data" will not be available before the prosthetic is used for the first time by a human without the use of his or her own arm. But even if the data were available, these methods make the assumption that the relationship between neuronal firing rates and movement parameters is constant across time and does not depend on whether the arm is real or prosthetic. These assumptions are probably not supported and may greatly effect the accuracy of the predicted movement. For instance, although real arm position has been shown to be related to the firing rates of primary motor cortex neurons, it is less obvious whether the position of a prosthetic device will have the same effect, when information about the position is only obtained visually, and not by way of any sensory input from the arm itself. Also, recent research suggests that neurons can change their firing patterns depending on whether they are controlling a real or prosthetic arm, and that when using a prosthetic arm the firing patterns continue to evolve in the presence of visual feedback [14]. Therefore, an algorithm that periodically updates estimates of the tuning parameters in real time should give improved control over the neuroprosthetic device.

Such an algorithm is described by Taylor et al. [14]. In a series of experiments, researchers recorded the firing rates of neurons in two rhesus monkeys using microelectrode arrays implanted into the motor cortex of each. While their neuronal activity was being recorded, the monkeys performed virtual reality variants of the standard 3-D center-out task, which has been used extensively

by neuroscientists trying to determine the relationship between movement and neuronal activity [1, 6, 12]. This task involves moving a cursor from a central location to one of several targets in space whenever the target appears on the screen. In these particular experiments, the movement of the cursor was controlled in one of two ways. Under “hand control,” the cursor was controlled by the movements of the monkey’s own arm in space; the firing rates of the neurons were simultaneously recorded but not used for on-line movement prediction. Under ‘brain control,’ the cursor was controlled only by the neuronal firing rates as they were recorded, using an algorithm that performed both iterative parameter estimation and movement prediction as discussed above. Meanwhile, the monkey’s arms were restrained. With practice, the algorithm allowed the monkeys to control the direction of the cursor’s movement with reasonable accuracy.

The algorithm used by Taylor et al. was not based on an explicit statistical model, although various constants used by the algorithm were chosen to minimize the error in past movement prediction when compared to an idealized trajectory (moving immediately to the target). Within a statistical framework, one can design an algorithm to perform the iterative process of parameter estimation and movement prediction in a more effective and flexible way. In the remainder of this article, we propose statistical methods for carrying out this process and demonstrate that they substantially outperform the existing algorithm.

## 2 The Data

All of the methods described below were tested on one or more of three datasets taken from the experiments of Taylor et al. The three datasets are taken from experiments using the same monkey and all carried out on the same day. Each dataset contains recordings of various movement variables as well as the times each electrode in the array recorded a rapid voltage change. For simplicity’s sake, in the remainder of this article we will refer to the recordings taken from a single electrode as though they came from one neuron, although for many of the electrodes it is unclear whether the observed electrical discharges were associated with a single neuron or instead a group of neurons.

The datasets come from three different experimental paradigms.

### 2.1 Hand Control

In the first experiment, the monkey faced a computer screen on which objects appeared to be in three dimensions. One of its arms was restrained while the other was free. It had been trained previously to use its arm to control the movement of a cursor on the screen. It was rewarded with water whenever it successfully moved the cursor from a center position to one of eight targets located at the corners of an imaginary cube in space. The targets were presented in random order. The entire experiment lasted about nine minutes. The dataset associated with this experiment includes the firing times of the 65 neurons,

with accuracy to within 0.000025 seconds. In addition the dataset includes the position of the cursor, recorded approximately every 30 milliseconds. One can use the cursor position to extract what the direction of hand movement was in each 30 millisecond increment, as well as the hand speed, acceleration, and higher derivatives of the speed if desired. Finally, the dataset also includes the position of the target at any given time.

## 2.2 Brain Control

In the second experiment, both of the monkey’s arms were restrained. The monkey was presented with the same task to perform, although in this experiment the monkey needed to learn to control the movement of the cursor by thinking about it, not by actually making the movements with its own arm. By using an algorithm that iteratively refined estimates of cell tuning properties as the monkey attempted to move the cursor, Taylor et al. were able to predict relatively accurate movement trajectories for the cursor based on the observed firing rates of the 65 neurons. The data produced in this experiment includes the same variables as that for the hand control experiment, with the exception that the cursor position does not correspond to arm position but to the predictions given by Taylor’s algorithm.

## 2.3 Brain Control with Constant Parameters

In the third experiment, the monkey’s arms were still restrained and it continued to move the cursor by modulating the firing rates of the 65 recorded neurons. However, in this experiment the parameters used by Taylor’s algorithm were fixed, taken from a period in the previous experiment in which the algorithm was making the most accurate predictions. This experiment also introduced six new targets, in addition to the eight used in the previous two experiments. These targets corresponded to center of each of the six faces of a cube. This was to determine whether the parameters that had been chosen in the previous experiment would allow the monkey to make movements to novel targets.

## 3 Statistical Methods

Suppose that for each small increment of time (say 30 ms), we model the relationship between how many times a particular neuron fires and the intended movement for that time increment using the likelihood  $p(R_t^{(i)}|M_t)$ , where  $R_t^{(i)}$  represents the firing counts of neuron  $i$  in interval  $t$  and  $M_t$  is a vector of variables related to the intended movement at time  $t$ , such as speed and direction. In reality this relationship may not be instantaneous, and we will discuss later how these methods can be extended to incorporate any time delays in the system. Given that we observe the firing rates of a population of neurons, predicting the intended movement the prosthetic can be done in several ways. One way is to use maximum likelihood estimation for the components of  $M_t$ . Another

is to specify a prior distribution for  $M_t$  and then base our estimates on the resulting posterior distribution. In either case, it is critical that we model the likelihood well. Finding a good model and estimating the associated parameters is described in the next section. Then we return to the question of movement prediction and give more details about the two methods above.

### 3.1 Model selection and parameter estimation

Since the number of times each neuron fires in a particular interval of time is a count variable, one way to model the firing activity for a given neuron is as a Poisson process. However, because the rate of firing in a given interval may depend on other covariates including variables related to intended movement, we take this Poisson process to be inhomogeneous and model the rate of the process as a function of other variables. We can then use Poisson regression to estimate any model parameters. The main question becomes what variables should be included in the model and what form the mean function should take.

#### 3.1.1 Intended Direction

The relationship between the direction of real arm movements and neuronal firing rates has been studied more than for any other movement variable. Researchers have found a strong relationship between the direction of arm movement and the firing rates of neurons in the primary motor cortex [1, 2, 6, 7, 10, 12, 13]. Most of these studies have found that the mean firing rates for a given neuron are often well fit by some function of the cosine of the angle between the direction the arm actually moved and the neuron’s “preferred direction,” which is the name given to the direction in which the neuron fires the most.

For controlling a neuroprosthetic, the important question is not how firing rates are affected by actual movement directions, but by intended movement directions. In order to fit such a model, one needs data on the intended movement, which is greatly facilitated by using an experimental design such as those described above. In these experiments, the monkey was rewarded for moving the cursor into the target, so a good guess of intended direction at each time point is the direction needed to move toward the target. When the neuroprosthetic is being used by a human, such a task could also be used as a calibration method for the algorithm, and we could rely also on stated intent instead of relying solely on the reward structure of the experiment.

For a given neuron  $i$ , we may use a Poisson regression model to relate the firing rate in time interval  $t$ ,  $\lambda_t^{(i)}$ , to intended direction. By way of analogy to the cosine models used to relate mean firing rates to observed direction of real arm movements, we write the firing rate in time interval  $t$  as

$$\lambda_t^{(i)} = \delta * \exp \left\{ \beta_0^{(i)} + \beta_x^{(i)} Dx_{t+\tau} + \beta_y^{(i)} Dy_{t+\tau} + \beta_z^{(i)} Dz_{t+\tau} \right\}, \quad (1)$$

where  $\delta$  is the length of the time interval. The covariates  $Dx$ ,  $Dy$ , and  $Dz$  represent intended direction in terms of the associated point on the unit sphere.

Because the cosine of the angle between two unit vectors is equal to their dot product, this linear model is equivalent to a model in which the firing rate is a function of the cosine between the intended direction and the neuron’s “preferred direction.” The time difference parameter  $\tau$  can be either positive or negative and must be estimated along with the other parameters.

To estimate  $\tau$ , one can use a measure of how much the fitted model given a particular value of  $\tau$  reduces the model deviance relative to a model including only the intercept. For instance, choose  $\tau$  to maximize

$$\mathcal{D} = \frac{\text{null deviance} - \text{residual deviance}}{\text{null deviance}} \quad (2)$$

where the residual deviance is under the model for a given  $\tau$ , estimating the  $\beta$  coefficients using Poisson regression. The reason for using this ratio instead of simply choosing  $\tau$  to minimize residual deviance is that changing  $\tau$  may also change the number of missing values present in the data used to estimate the model, so the null deviance may change as well. One could also smooth the values of  $\mathcal{D}$  and maximize over the smoothed values. An example of these smoothed values of  $\mathcal{D}$  obtained for one neuron are shown in Figure 1.

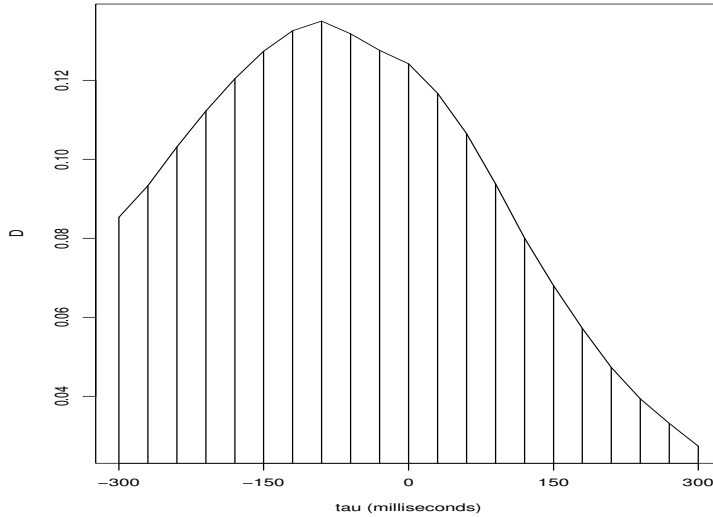


Figure 1: Values of the statistic  $\mathcal{D}$  corresponding to possible values of  $\tau$  between -300 ms and 300 ms. There is a clear maximum at  $\tau = -90$  ms, so this would be chosen as an estimate of  $\tau$ .

The results shown in Figure 1 are typical among the neurons in the dataset whose mean firing rates varied the most with intended direction. Previous research on real arm movements has also accounted for these time lag parameters

[10, 16]. However, the interpretation of  $\tau$  is different when the model includes intended movements and not real arm movements. Specifically, when modeling real arm movements the estimated parameters tend to be positive, indicating that a change in firing rate usually precedes a change in movement, which is not surprising. Occasionally in studies of real arm movements, neurons also have lags which are estimated to be negative. This is usually interpreted as being due to some feedback mechanism, whereby the movements the arm makes are relayed back to the brain.

When we model intended movement, the interpretation changes. The intended movement will be formed using visual feedback. For instance, in the case of the center-out task, if the monkey sees the target to the left of the cursor on the screen, we assume that its intended direction of movement is to the left. But if we calculate the intended direction based on what currently appears on the screen, we will ignore any time that this visual feedback takes. We fit models for intended direction for each of the neurons in the dataset described in section 2.3, where the movement of the cursor was controlled by the neuronal firing rates and Taylor’s algorithm using fixed parameter values. We found that the estimates of  $\tau$  were more often negative, indicating that what was on the screen preceded a change in the firing rates. Using a model that includes a negative value for  $\tau$  raises some challenges for prediction that are discussed in section 3.2.

Given a particular value of  $\tau$ , the estimated coefficients in the linear model give an estimate of each cell’s preferred direction. In fact, the preferred direction for cell  $i$  lies through the point  $(\beta_x^{(i)}, \beta_y^{(i)}, \beta_x^{(i)})$ . It has been argued that the quality of predictive algorithms increases when the neurons in the recorded population have widely distributed preferred directions. It is clearly also beneficial to have neurons which are strongly tuned in the sense that the average firing rate changes a lot depending on the intended movement. One measure of tuning strength is the ratio of the firing rate in the preferred direction to the firing rate in the opposite direction. The symmetry of the cosine function implies that this is the direction in which the neuron tends to fire the least. Figure 2 shows the distribution of preferred directions during the constant parameter task described in section 2.3, with relative tuning strength indicated by the size of the points.

Another interesting finding was that the neurons with the highest tuning strengths also had the least variable estimates of  $\tau$  whenever  $\tau$  was estimated using distinct blocks of the dataset, each block consisting of all targets in the experiment being presented once.

### 3.1.2 Position of the cursor

Other variables not related directly to the monkey’s intended movement may also be represented in the firing rates of the neurons. For instance, previous research has shown that the firing rates of motor cortex movements may be affected by hand position. Although the monkey’s arms are strapped down in the brain control experiments, we can explore whether the position of the cursor

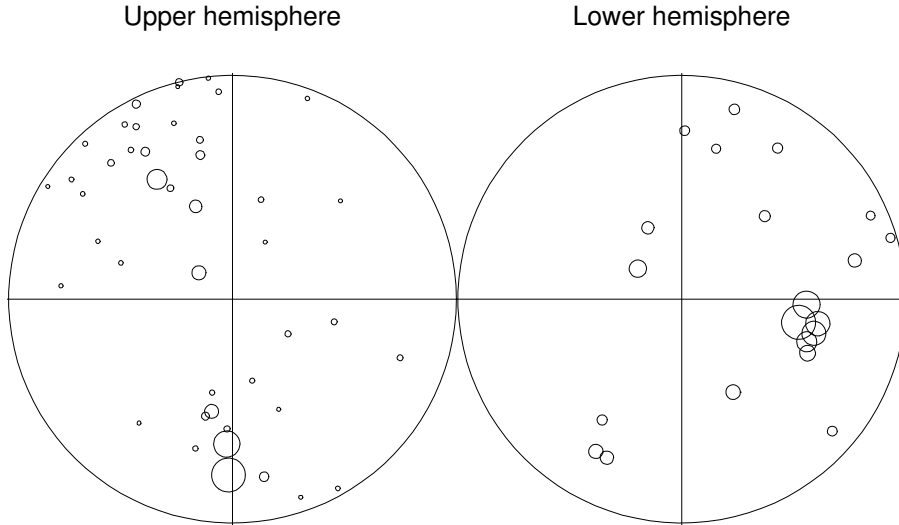


Figure 2: Preferred directions of each of 65 neurons recorded during the constant parameter experiment. Each direction is shown by the corresponding point on the unit sphere, and the sphere is shown in two dimensions using an equal area projection. The size of the points is proportional to tuning strength. Note that the distribution of preferred directions is close to uniform.

the monkey observes is represented somehow in the firing rates.

To compare the degree to which cursor position was represented, in comparison to intended direction, we fit models for each neuron that either included position or intended direction. The models for direction used the parametric form given in (1), a cosine of each neuron’s preferred direction. The models for position were allowed to take a more general form. Specifically, for each neuron we fit a generalized additive model [8], using a smoothing spline to fit each component of position. That is, the firing rate for neuron  $i$  at time  $t$  was modeled as

$$\lambda_t^{(i)} = \delta * \exp \left\{ \alpha_0^{(i)} + s_x^{(i)}(x_{t+\tau}) + s_y^{(i)}(y_{t+\tau}) + s_z^{(i)}(z_{t+\tau}) \right\}, \quad (3)$$

where  $(x_{t+\tau}, y_{t+\tau}, z_{t+\tau})$  is the position of the cursor at time  $t + \tau$  and each function  $s$  represents a different smoothing spline.

For each neuron and each of the two models (including either intended direction or cursor position), we calculated a statistic similar to that in (2). Now let

$$\mathcal{D}_{\text{pred}} = \frac{\text{null deviance} - \text{predictive deviance}}{\text{null deviance}}.$$

This statistic was calculated by first fitting each of the two models, for intended direction and cursor position, to part of the dataset. Then, the null

deviance was found for the rest of the dataset. (Note since the null deviance is for an intercept only model, it is the same whether no matter which of the two other models we are considering). The predictive deviance for each model was calculated as the residual deviance when the model fit on the previous part of the dataset was used to predict the firing rates in the rest. As such,  $\mathcal{D}_{\text{pred}}$  gives a measure of how much including a particular variable, such as intended direction or position, improves the model over an above an intercept only model. However, since we apply the model to a dataset other than the one used to fit the model, we avoid problems with overfitting. Unlike the original  $\mathcal{D}$  statistic,  $\mathcal{D}_{\text{pred}}$  can be either positive or negative. As shown in Figure 3, the values of  $\mathcal{D}_{\text{pred}}$  found for each neuron and each of the two models indicate that intended direction is more strongly represented in the firing rates than is cursor position. In addition, in general it is the same neurons which tend to encode both intended direction and position.

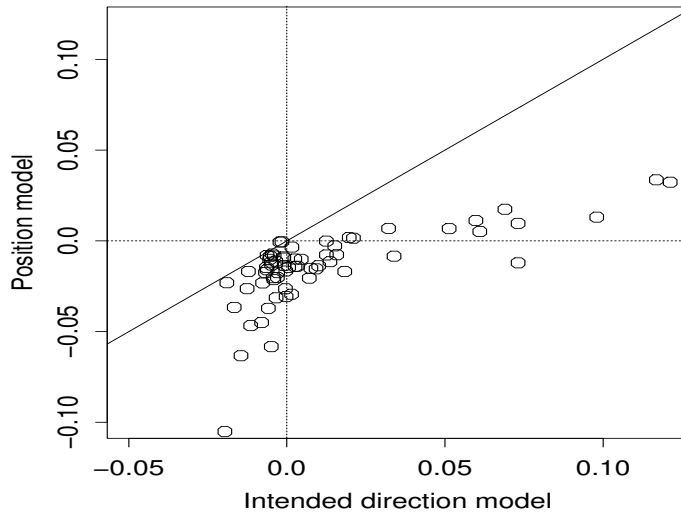


Figure 3: Each point on the plot represents the values of  $\mathcal{D}_{\text{pred}}$  found for a single neuron. Many of the recorded neurons do not seem to be encoding either intended direction or position. Among those that do, generally intended direction is encoded more strongly than position.

## 3.2 Movement prediction

Returning to the problem of predicting movement for time interval  $t$  given a model fitted on data prior to time  $t$ , we explore the frequentist and Bayesian methods mentioned at the beginning of section 3.

### 3.2.1 Maximum likelihood estimation

Suppose that the likelihood for the firing rates of the entire population of  $N$  neurons is the product of the Poisson densities for each neuron, each with mean  $\lambda_t^{(i)}$  a function of the intended direction as given in (1). Then we may predict the intended direction by maximizing the likelihood, subject to the constraint that the point  $(Dx, Dy, Dz)$  lie on the unit sphere. Although there is no closed form representation for the maximum likelihood estimates, they can be found numerically. One easy way to do this is to take a sample of points distributed uniformly on the sphere and evaluate the likelihood at each point.

### 3.2.2 Bayesian methods

However, because the movement at time  $t$  is closely related to the movement at time  $t - 1$ , it makes sense to consider a prior distribution for  $(Dx, Dy, Dz)_t$ . For instance, a prior distribution for movements assumed to be somewhat smooth might have mean direction  $(Dx, Dy, Dz)_{t-1}$  and be rotationally symmetric about its mean, saying that it is most likely that the movement will continue in the same direction, with small changes in direction slightly less likely, and large changes in direction even less likely. We can parameterize such a prior distribution using the spherical Fisher distribution. Reparameterizing the unit vector  $(Dx, Dy, Dz)_t$  in terms of the angles  $\theta$  and  $\phi$ , where  $\theta = \arccos(Dz)$  and  $\phi = \arctan(Dy/Dx)$ , the Fisher prior distribution has pdf

$$\pi(\theta, \phi) = \frac{\kappa}{4\pi \sinh \kappa} \exp \{ \kappa [ \sin \theta \sin \alpha \cos(\phi - \beta) + \cos \theta \cos \alpha ] \} \sin \theta$$

where  $\alpha$  and  $\beta$  are the angles describing the mean direction and  $\kappa$  is a scale parameter, with larger values of  $\kappa$  giving increased concentration about the mean.

If one wanted to include less information in the prior, one could also use a uniform distribution on the sphere, making all directions equally likely. In either case, one can calculate the posterior distribution for intended direction given observed firing rates and use the posterior mean or median at each time interval to predict the intended direction in that interval. This can be used to direct the movement of the prosthetic device.

### 3.2.3 Monte Carlo approximations to the posterior distributions

There is no closed form expression for the posterior distributions above, but importance sampling can be used to represent them using a sample of directions.

The sample mean can then be used as the estimate of the intended direction. Recall the basic importance sampling scheme for sampling from a posterior distribution given a likelihood  $\mathcal{L}(Dx, Dy, Dz)$  and prior distribution  $\pi(Dx, Dy, Dz)$ : Draw  $B$  samples from a distribution with pdf  $g(Dx, Dy, Dz)$ . Then evaluate the importance weights  $w(Dx, Dy, Dz) = \mathcal{L}(Dx, Dy, Dz)\pi(Dx, Dy, Dz)/g(Dx, Dy, Dz)$  for each  $(Dx, Dy, Dz)$  in the sample. This produces a weighted sample of size  $B$  from the posterior, or one can resample according to the weights to obtain an unweighted sample. These calculations are particularly straightforward if the distribution  $g(Dx, Dy, Dz)$  is the prior distribution  $\pi(Dx, Dy, Dz)$ , in which case the weights simplify to be proportional to the likelihood  $\mathcal{L}(Dx, Dy, Dz)$ . For instance, if a uniform prior distribution is used at each time step, it is easy to produce a sample which is uniformly distributed on the unit sphere, and the likelihood at each sampled point can be evaluated by multiplying the Poisson densities with separate means  $\lambda_t^{(i)}$  for each neuron.

However, how can one sample from the prior distribution when the direction for time interval is assumed to come from a Fisher distribution centered at the previous interval's direction, which is also unknown? A more powerful type of Monte Carlo approximation is needed in order to estimate the entire sequence of directions sequentially. This type of model falls into a class of models called generalized state-space models or dynamic models, if we make a few additional assumptions. We have a sequence of observed variables, which in the movement prediction problem is the sequence of firing rates for the whole population of neurons, and a sequence of unobserved variables, which is the sequence of intended directions. The generalized state-space model assumes that the unobserved sequence, the directions, form a Markov chain with some specified transition density. In analogy to the Fisher prior above, we can take the transition density to be according to a random walk with Fisher noise. An additional assumption of the model is that the current firing rates depend only on the current intended direction.

There is a type of sequential importance sampling scheme called particle filtering which relies on the structure of the state-space model [3]. Suppose we have a sample from the posterior distribution for time interval  $t$ , based on all the observed firing rates up to and including time  $t$ , which we will denote as  $R_{1,\dots,t}$ . In order to obtain a sample from the prior distribution at time  $t + 1$ , note that

$$\pi((Dx, Dy, Dz)_{t+1}|R_{1,\dots,t}) = \int p((Dx, Dy, Dz)_{t+1}|(Dx, Dy, Dz)_t) * p((Dx, Dy, Dz)_t|R_{1,\dots,t})d(Dx, Dy, Dz)_t$$

where the first term being integrated is the Markov transition density assumed in the model and the second term is the posterior density, for which we already have a sample. In order to obtain a sample from the prior, therefore, one can sample from  $p((Dx, Dy, Dz)_{t+1}|(Dx, Dy, Dz)_t)$  where the values of  $(Dx, Dy, Dz)_t$  come from the current posterior sample. For instance, when using the Fisher

transition density, this means that for each  $(Dx, Dy, Dz)_t$  in the posterior sample, we draw a new point  $(Dx, Dy, Dz)_{t+1}$  from a Fisher distribution centered about  $(Dx, Dy, Dz)_t$ . (See [4] for algorithms that can be used to generate pseudo-random samples from distributions defined on the sphere.) Choosing the concentration parameter  $\kappa$  for the Fisher distribution is discussed in section 3.3.

### 3.2.4 Time lags

Finally, we need to address the question of how the time delay parameters may be incorporated into prediction. If  $\tau$  is positive, this means we can make a prediction for the intended movement several time intervals ahead. However, if  $\tau$  is negative, then to make a prediction at time  $t$  we must wait until we see the data up to some time point strictly greater than  $t$ . For the work described here, we have estimated the model which includes  $\tau$ , but taken  $\tau$  to be zero when doing prediction so that predictions can be made in real time. At this point it is unclear how including  $\tau$  in the model, whether  $\tau$  is negative or positive, would affect the accuracy of the prediction, and much work could be done here. For instance, more complicated models could allow  $\tau$  to vary not only by neuron but to vary in time or with the type of movement. There is some evidence that this is true for real arm movements [10].

### 3.2.5 Presmoothing

One final refinement that may be made to the methods above is to look at longer time intervals. For instance, instead of using each 30 millisecond time interval separately, apply the same methods but take the data for the firing counts at time  $t$  to be the sum of the firing counts from  $t - 1$  to  $t + 1$ . The length of the interval  $\delta$  used in (1) will now be three times as large. All the results reported in the next section follow this convention, as it gave better performance in every case.

## 3.3 Prediction Results

In order to examine the accuracy of the predictions given by the maximum likelihood and Bayesian methods, we applied each to the dataset described in 2.3, in which the cursor movement was controlled by the firing rates and Taylor’s algorithm with fixed parameter values. Note that because we applied the methods to a static dataset and not while the prosthetic was in use, what the monkey saw on the screen at any given time was determined by Taylor’s algorithm, not the new algorithms we want to test. Therefore, we can’t predict the entire trajectory for this dataset, since the intended directions corresponded to what the monkey saw on the screen. Instead, at each time interval we predicted only the direction the cursor should go next, not the resulting position. To assess the accuracy at each time interval, we calculate the angle between the inferred intended direction (the direction needed to reach the target) and the direction

given by that particular method. We call this the angle of discrepancy. This angle will be zero if the prediction is perfect and  $\pi$  radians if the prediction is in exactly the wrong direction. Applying this measure of accuracy, we examine how well each method performed under various settings.

One question of interest when using a Monte Carlo method such as the particle filter is how many samples are needed. We calculated the angle of discrepancy using varying sizes of samples, from 10 to 200. In general, there was little improvement gained by using a sample size of over 50. The results shown below use 50 samples.

Overall, the method of prediction based on using a Fisher transition density worked the best. The angle of discrepancy was reduced substantially over Taylor’s algorithm, and very modest improvements were seen over the maximum likelihood method. Most of these improvements are probably due to the smoothness assumptions made by using the Fisher transition density. The observed distributions for the angle of discrepancy at each time point for each of four methods are shown in Figure 3.3. Predictions chosen at random are shown first, for reference.

## 4 Future work

### 4.1 Speed

In the models above, we took the monkey’s intended direction at any given time to be the direction needed to go towards the target. Intentions for other movement variables such as speed are much harder to infer given the structure of the experiment. We can use natural arm movements as a guide to what the intention might be, although in a virtual reality environment such as the one used in the experiments described above, there is nothing to constrain the movement to look like natural movements. In this environment, we might infer that the intended speed is the one needed to move to the center of the target in the smallest increment of time used by the algorithm. However, we are still exploring how to modify the model for direction given in 1 to include a term for speed. Initial results show that including a linear term for intended speed do not fit the data well and more complicated models may be needed.

### 4.2 Modifying modeling assumptions

Also implicit in the models above was the assumption that neurons act independently. This is clearly a tenuous assumption to make, since we know that electrical signals are transmitted from one neuron to the next. However, it is very difficult to model the covariance structure between multiple neurons when the firing counts are assumed to be, say, Poisson distributed and not normally distributed. However, if one could devise a method of incorporating information about the relationships between multiple neurons, there is a potential for improvement in predicting movement.

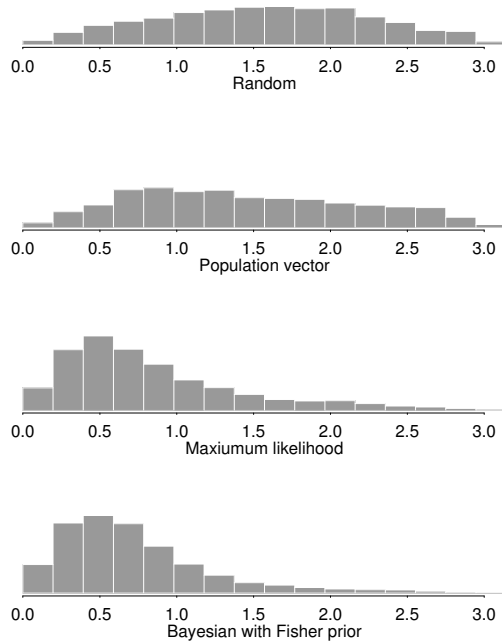


Figure 4: Distribution of angle of discrepancy for four methods of predicting intended direction. The results for the last method, Bayesian with Fisher prior, are when the parameter  $\kappa$  in the Fisher distribution is 10.

### 4.3 Real time implementation of statistical methods

We are currently working to implement the statistical methods described above in a future experiment with a rhesus monkey. In this experiment, the monkey will perform tasks similar to those described in this paper. However, the movement of the cursor will be determined by applying the statistical methods above to the observed firing rates. This will enable us to predict entire trajectories of movement, as well as to determine how quickly these methods allow the monkey to learn to control the cursor’s movement.

## References

- [1] B. Amirikian and A.P. Georgopoulos. Directional tuning profiles of motor cortical cells. *Neuroscience Research*, 36:73-79, 2000.

- [2] J. Ashe and A.P. Georgopoulos. Movement parameters and neural activity in motor cortex and area 5. *Cerebral Cortex*, 6:590:500, 1994.
- [3] A. Doucet, N. de Freitas, and N. Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Springer, New York, 2001.
- [4] N.I. Fisher, T. Lewis, and B.J.J. Embleton. *Statistical analysis of spherical data*. Cambridge University Press, Cambridge, 1987.
- [5] Y. Gao, M.J. Black, E. Bienenstock, S. Schoham, and J.P. Donoghue. Probabilistic inference of hand motion from neural activity in motor cortex. *Advances in Neural Information Processing Systems*, 14, 2002.
- [6] A.P. Georgopoulos, J.F. Kalaska, R. Caminiti, and J.T. Massey. On the relations between the direction of two-dimensional arm movements and cell discharge in primate motor cortex. *The Journal of Neuroscience*, 2:1527-1537, 1982.
- [7] A.P. Georgopoulos, R.E. Kettner, and A.B. Schwartz. Primate motor cortex and free arm movements to visual targets in three-dimensional space. II. Coding of the direction of movement by a neuronal population. *The Journal of Neuroscience*, 8:2928-2937.
- [8] T.J. Hastie and R.J. Tibshirani. *Generalized Additive Models*. Chapman & Hall, London, 1990.
- [9] R.E. Isaacs, D.J. Weber, and A.B. Schwartz. Work toward real-time control of a cortical neural prosthesis. *IEEE Transactions on Rehabilitation Engineering*, 8:196-198, 2000.
- [10] D.W. Moran and A.B. Schwartz. Motor cortical representation of speed and direction during reaching. *Journal of Neurophysiology*, 82:2676-92, 1999.
- [11] M.K. Pitt and N. Shephard. Auxiliary based particle filters. In *Sequential Monte Carlo Methods in Practice* Springer, New York, 2001.
- [12] A.B. Schwartz, R.E. Kettner, and A.P. Georgopoulos. Primate motor cortex and free arm movements to visual targets in three-dimensional space. I. Relations between single cell discharge and direction of movement. *The Journal of Neuroscience*, 8:2913-2927, 1988.
- [13] A.B. Schwartz and D.W. Moran. Arm trajectory and representation of movement processing in motor cortical activity. *European Journal of Neuroscience*, 12:1851-1856, 2000.
- [14] D.M. Taylor, S.I. Helms Tillery, and A.B. Schwartz. Direct cortical control of 3d neuroprosthetic devices. *Science*, 296:1829-1832, 2002.
- [15] I. Wickelgren. News focus: tapping the mind. *Science*, 299:496-499, 2003.

- [16] W. Wu, M.J. Black, Y.Gao, E. Bienenstock, M. Serruya, and J.P. Donoghue. Inferring hand motion from multi-cell recordings in motor cortex using a Kalman filter. SAB'02, Workshop on Motor Control in Humans and Robots: On the Interplay of Real Brains and Artificial Devices, Edinburgh, Scotland, 66-73, 2002.