Biol. Cybern. (2003)
DOI 10.1007/s00422-002-0374-6

Biological Cybernetics

Robustness of neuroprosthetic decoding algorithms

Mijail Serruya[1], **Nicholas Hatsopoulos**[2], **Matthew Fellows**[1], **Liam Paninski**[3], **John Donoghue**[1]

[1] Department of Neuroscience, Brown University, Providence, RI 02912, USA
[2] Department of Organismal Biology and Anatomy, University of Chicago, Chicago, IL 60637, USA
[3] Center for Neural Science, New York University, New York, NY 10003, USA

Received: 10 June 2001 / Accepted in revised form: 31 October 2002 / Published online: 21 February 2003

Abstract. We assessed the ability of two algorithms to predict hand kinematics from neural activity as a function of the amount of data used to determine the algorithm parameters. Using chronically implanted intracortical arrays, single- and multineuron discharge was recorded during trained step tracking and slow continuous tracking tasks in macaque monkeys. The effect of increasing the amount of data used to build a neural decoding model on the ability of that model to predict hand kinematics accurately was examined. We evaluated how well a maximum-likelihood model classified discrete reaching directions and how well a linear filter model reconstructed continuous hand positions over time within and across days. For each of these two models we asked two questions: (1) How does classification performance change as the amount of data the model is built upon increases? (2) How does varying the time interval between the data used to build the model and the data used to test the model affect reconstruction? Less than 1 min of data for the discrete task (8 to 13 neurons) and less than 3 min (8 to 18 neurons) for the continuous task were required to build optimal models. Optimal performance was defined by a cost function we derived that reflects both the ability of the model to predict kinematics accurately and the cost of taking more time to build such models. For both the maximum-likelihood classifier and the linear filter model, increasing the duration between the time of building and testing the model within a day did not cause any significant trend of degradation or improvement in performance. Linear filters built on one day and tested on neural data on a subsequent day generated error-measure distributions that were not significantly different from those generated when the linear filters were tested on neural data from the initial day ($p < 0.05$, Kolmogorov-Smirnov test). These data show that only a small amount of data from a limited number of cortical neurons appears to be necessary to construct robust models to predict kinematic parameters for the subsequent hours. Motor-control signals derived from neurons in motor cortex can be reliably acquired for use in neural prosthetic devices. Adequate decoding models can be built rapidly from small numbers of cells and maintained with daily calibration sessions.

Correspondence to: M. Serruya
(e-mail: Mijail_Serruya@brown.edu)

1 Introduction

Over 600,000 patients in the United States suffer from severe motor impairment due to spinal-cord and brainstem trauma, amyotrophic lateral sclerosis, severe muscular dystrophy, and pontine stroke. The acute and chronic treatment of all these conditions costs billions of dollars each year. For the majority of these individuals, the cerebrum is intact, indicating that the central control for fine motor behavior remains healthy despite being cut off from its effectors. Evidence for this includes three recent fMRI studies (Shoham et al. 2001; Humphrey et al. 2000; Foltys et al. 2000) in which patients who had lived with quadriplegia for several years activated primary motor cortex when imagining movement of paralyzed limbs.

While brain-computer interfaces based on EEG activity have been studied intensively for over 40 years (Robinson 2000), the long time constant and global nature of the EEG signal hampered its use for rapid and fine effector control such as a computer cursor or patients' own muscles via implanted muscle stimulators (Lauer et al. 1999, 2000).

By contrast, intracortically recorded neural activity potentially comprises a valuable multidimensional signal to control human neural prosthetic devices. Although long-term recording of multiple neurons presents formidable technical challenges, there have been a number of significant recent successes in advancing this technology, making possible its adoption for use in neural prosthetics (Maynard et al. 1997, 2000; Williams et al. 1999). Recordings from multiple electrodes implanted

intracortically in nonhuman primates can be used to reconstruct hand location (Maynard et al. 1999; Paninski et al. 1999; Wessberg et al. 2000; Helms Tillery et al. 2000; Serruya et al. 2001; Hatsopoulos et al. 2001). Further, Kennedy et al. (2000) have implanted pairs of electrodes in the primary motor cortex of several patients with locked-in syndrome. Using only the activity of two to five isolated neurons as input, patients have learned to control a computer cursor to communicate (Kennedy et al. 2000). However, there is presently no established optimal decoding method. A further unknown concerning the use of single unit activity as control signals is the day-to-day stability of decoding. Thus, an accurate decoding algorithm on one day may be grossly inaccurate some time later if the recorded population changes either through electrode motion or neural plasticity. Determining the optimal algorithm requires understanding both the amount of data needed to provide a useful signal and the necessity for recalibration.

Decoding algorithms used to create control signals that can drive motor prosthetic devices can be organized into two broad classes: discrete and continuous. Discrete algorithms treat the decoding problem as one of classifying a sample of neural activity into one of several distinct action categories. Maynard et al. (1999) were able to correctly predict direction (out of a possible eight directions) in up to 90% of tested trials using the firing rates of 16 cells in a 600-ms window centered around movement onset. Hatsopoulos et al. (2001) were able to correctly predict direction (out of two possible directions) in 100% of tested trials using the firing rates of eight cells in a 200-ms window right before movement onset. In both cases these are average results using different combinations of cells. One of the important features of motor cortex is that decoding does not depend on particular cells with exceptional tuning; rather many, if not most, cells in motor cortex participate in the representation. Classification is inherently more complex if it is ultimately to return information about continuous trajectory. The two decoding algorithms examined here provide either continuous, absolute hand 2D position or discrete movement direction.

Recently, two groups demonstrated that macaques could learn to use neurally controlled cursors in a behaviorally useful way (Serruya et al. 2002; Taylor et al. 2002). While both groups report that control could be achieved when recording from only a small group of neurons (both used less than 20), they did not document how the amount of time to build the model affected reconstruction and behavioral utility. The study of Taylor et al. (2002) did examine activity across days; however, a multitude of different decoding approaches with iterative features were used, precluding the possibility of assessing decoder stability across days. Hence while the two studies confirm, using different calibration and decoding techniques, that nonhuman primates can use neurally derived signals in a behaviorally useful way, a rigorous analysis of discrete and continuous decoders with offline data can prepare the way for improved online closed-loop applications.

2 Methods

2.1 Behavioral tasks

Three male macaque monkeys MR, ME, and MT (one *Macaca fascicularis*, two *M. mulatta*) and one female monkey MC (*M. mulatta*) were operantly conditioned to perform two different arm-reaching tasks. For both tasks, the monkey moved a manipulandum at the end of a two-joint arm in the horizontal plane that controlled the position of a cursor on a vertically oriented monitor. The continuous tracking task required visually guided, planar, slow tracking hand movement (Paninski et al. 2003), and the discrete movement task involved step-tracking movements from a central holding position to a radially located target position (Georgopoulos et al. 1986; Hatsopoulos et al. 1998a, 1998b).

For the discrete task, the monkey moved the cursor from a center "hold" target to one of several peripherally positioned targets. The distance from the center target to the peripheral target was 6 cm on the tablet and 3 cm on the screen. A trial was composed of three epochs: a hold period during which time the monkey had to maintain the cursor at the hold position for 0.5 s, a random 1-s to 1.5-s "instructed delay" period during which the target for the forthcoming action appeared but movement was withheld, and a "go" period initiated by target blinking (mean reaction time, ~365 ms), which was followed by movement to the appropriate target within a 600-ms interval. The position of the manipulandum was digitized at 167 Hz, with an accuracy of 0.25 mm (tablet and pen system: Wacom Technology Corp., Vancouver, WA) and recorded to disk. Three versions of the task were used with two, four, or eight possible final targets. Monkeys successfully completed 50 to 150 trials per direction during a recording session.

For the continuous tracking task, the monkey was required to manually follow a target that moved smoothly and randomly on the computer screen. Instantaneous visual position feedback, indicating hand position on the tablet, was provided by a black dot on the screen (the feedback cursor; 0.5-cm diameter corresponding to 0.3° of visual angle). At the beginning of each trial, a larger (1.5-cm radius, 0.8°) red circle appeared on the screen in a random position drawn from a two-dimensional, zero-covariance Gaussian (up to the cutoff imposed by the edge of the screen) distribution. The initial position of the monkey's hand was unconstrained. If the monkey did not move his hand such that the feedback cursor fell within the target circle within 1.5 s (4 s for one of the animals) after the appearance of the target, the trial aborted and the target appeared at a new (independently, identically distributed) position to begin the next trial. If the target was acquired within the allotted time, the target remained stationary for 1 s, as a stabilization period. If this hold phase was completed satisfactorily, the target began to move in a smooth, but random, fashion. If the monkey was able to maintain the feedback cursor within the interior of the target circle for 6 to 10 s (the length of the trials was increased and the radius of the target decreased with training), the animal

received a juice reward and the trial ended. The trial aborted if the feedback cursor left the interior of the target during this time. By this design, the horizontal and vertical components of stimuli guiding hand position during this period were generated by a Fourier synthesis of a finite number of sinusoids of different frequencies with random phases and magnitudes (Paninski et al. 2003). The path taken by the stimulus was randomly generated and presented only once such that a given trajectory was never displayed twice in an experiment.

2.2 Neural recording

Details of the basic recording device and protocols are available elsewhere (Maynard et al. 1997, 1999), with only a few differences to note here. When the monkeys were trained up to a success rate of 80% on the visuomotor tracking task, a microelectrode array consisting of 100, 1.0 mm or 1.5-mm-long platinized-tip silicon probes (Bionic Technologies, Inc.; impedances between 200 and 500 kΩ using a 1-nA, 1-kHz sine wave), regularly arranged in a square grid (400-μm interelectrode separation), was implanted in the arm representation of MI. [See Donoghue et al. (1998) for additional details on surgical procedures and animal care.] All procedures were in accordance with protocols approved by Brown University Institutional Animal Care and Use Committee and with the Guide for the Care and Use of Laboratory Animals (National Institute of Health publication no. 85–23, revised 1985). During a recording session, signals were amplified and sampled at either 20, 30, or 40 kHz per channel using a commercially available amplification and signal-processing system (Datawave Technologies, Longmont, CO; Bionic Technologies Inc., Salt Lake City, UT; Plexon, Inc., Dallas, TX). All waveforms that crossed an experimenter-set threshold were stored (from 0.4 ms before to 1.5 ms after the threshold crossing). Single-unit isolation was achieved by a two-part process. First, offline spike sorting was performed using one of three different spike-sorting algorithms: (1) clustering using features of the waveform (time to first peak and trough, peak and trough magnitude), (2) clustering on the first two principal components of the waveform to generate prototype waveforms, and (3) applying voltage-time boxes through which the waveform had to pass. Second, the interspike-interval histogram and autocorrelation histograms were examined to ensure that a unit did not spike within 2 ms after the previous spike (absolute refractory period). Only single units that were isolated by these criteria and had signal-to-noise ratios greater than 2.5 were analyzed further. For the continuous task, only data recorded during the tracking phase from 1 s after the target began moving to 1 s before the trial ended were used in the reconstruction analysis.

2.3 Mathematical analyses

For both the discrete and continuous kinematic reconstruction models, we determined the effect of varying the amount of data (number of trials) used to build the model or the time between a fixed amount of data used to build the model and a fixed amount of data to be classified. A third analysis comprised building the linear filter model on the entire data set for one day and testing it on the data recorded on another day. All analyses were executed in Matlab (MathWorks, Natick, MA).

2.3.1 Maximum-likelihood model. For the discrete-direction task, the spike count for each neuron in the 500-ms epoch following the go cue was used to estimate the mean and standard deviation (SD) of direction-conditioned Gaussian probability distributions(P(spike count|direction$_i$)). The go cue time was used as a reference instead of the start of movement time because it is more analogous to the situation of a paralyzed patient who would have no "start of movement" but who could use a device involving visual go cues. Our assumption of Gaussian directional tuning of the neurons pertains less to an accurate description of the neurons' encoding than to the ability of the classifier to perform accurately in a cross-validated manner.

To classify trials, the probability of moving in each direction, given the observed spike count, was calculated as follows. The spike count for each cell in the 500 ms after the go cue was compared with each of our previously generated conditionals to find the probability of that particular count for each given direction. The direction that maximized the likelihood of the data was considered the predicted direction (direction i for which P(direction$_i$ |spike count) is maximum). The performance measure comprised the number of correctly predicted directions divided by the total number of trials to be classified.

2.3.2 Linear filter model. For the continuous tracking task, neural spike count and kinematic data $(x(t); y(t))$ were placed into 50-ms bins. Linear filters were constructed by building a response matrix containing the firing-rate history of each neuron for the last 1.5 s (30 bins) and regressing this matrix onto the two columns of kinematic absolute positions using a pseudo-inverse technique. The closed-form solution of the least-squares formulation can be written as:

$$\mathbf{u} = \mathbf{R} \cdot \mathbf{f} = \mathbf{R}(\mathbf{R}^{\mathbf{T}}\mathbf{R})^{-1}\mathbf{R}^{\mathbf{T}}\mathbf{k}$$

where $\mathbf{R}$ is the response matrix, $\mathbf{f}$ the linear filter, $\mathbf{k}$ the kinematic values (absolute position), and $\mathbf{u}$ the reconstruction. The response matrix was built in the format outlined by Warland et al. (1997) and adapted for the motor system by Paninski et al (2003). The analysis was restricted to "causal" (predictive) filters such that for a given kinematic point at time t, the 30th bin (for each neuron) contained the rate at time t, with the previous bins containing the firing rates earlier in time. Thirty bins (corresponding to 1.5-s filters) were chosen because Paninski et al. (2003) found that shorter filters did not perform as well and that longer filters did not provide much additional information. Hence we generated two filters (one each for $x(t)$ and $y(t)$), each with 30

coefficients per cell and an additional offset coefficient. Unlike the maximum-likelihood model (which assumes a Gaussian encoding model probability distribution), and unlike several other models of how motor cortical neurons encode kinematic parameters that describe the relationship between firing rate and kinematics as a cosine tuning function (Georgopoulos et al. 1986; Helms Tillery et al. 2000), the linear filter model makes no assumptions about underlying neural encoding distributions of neural representation of a kinematic variable; it simply solves for the least-squared-error linear solution. While the construction of confidence intervals, computing mutual information, and other theories built around linear regression assume Gaussian noise, the least-mean-squares equation does not depend on normality anywhere. The mathematical details of the linear filter analysis in the time domain are presented in Paninski et al. (2003).

Each row in the response matrix **R** embodies the firing-rate history of all recorded neurons leading up to the corresponding point in the vector **k** in a manner such that they can be taken as self-contained data samples. This allowed us to concatenate all the data and parse them into 1-s windows (or 20 rows, as linear filter neural and kinematic data were binned in 50-ms intervals). Because we ensure that no individual sample includes data from more than one trial, the loss of distinction of trial stop and start times in parsing into 1-s samples does not affect subsequent analyses.

Three types of error measures were used to evaluate the performance of the linear filters: (1) the average Euclidean distance (in cm) between the actual and reconstructed positions,

$$\sum_{i=1}^{n} \sqrt{(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2} \ , \tag{1}$$

the r^2 value of the actual position accounted for by that of the reconstruction,

$$1 - \frac{\sum_{i=1}^{n} (x_i - \hat{x}_i)^2}{\sum_{i=1}^{n} (x_i - \bar{x})^2} \ , \tag{2}$$

and the correlation coefficient (ρ) between the actual and reconstructed positions

$$\frac{\sum_{i=1}^{n} (x_i - \overline{x_i}) \cdot \left(\hat{x}_i - \overline{\hat{x}_i}\right)}{\sqrt{\left(\sum_{i=1}^{n} (x_i - \overline{x_i})^2\right) \cdot \left(\sum_{i=1}^{n} \left(\hat{x}_i - \overline{\hat{x}_i}\right)^2\right)}} \ . \tag{3}$$

Separate correlation coefficients and r^2 values were computed for both the absolute horizontal (x) and vertical (y) positions of the manipulandum.

2.3.3 Amount of data. To address the question of how much data was necessary to build a robust decoding model, we systematically varied the amount of data used to build a model and observed the prediction accuracy of kinematics from neural activity in a fixed set of mutually exclusive trials. For both the discrete and continuous data sets, we used time as the independent variable encompassing amount of data. For the discrete task, data samples comprised 500-ms windows after the go cue of a single trial. The continuous data were then parsed into 10-s windows (data having been previously concatenated into 1-s windows), each of which was considered a possible sample to be used in the data-accrual process. Table 1 provides a summary of the nine data sets analyzed.

The procedure conducted to evaluate the effect of the amount of data on a model's ability to predict movement accurately consisted of two steps. In the first step, the data were split into a test subset, which would later be classified and was never used in the construction of models, and a build subset, a pool of data from which a third accrual set could be built up. In both the discrete and continuous data, the test subset comprised the last 10% of total samples available. The second step was to iteratively build up the size of the accrual set by randomly selecting data samples from the build set and using them to build a model and compute classification performance of the test subset trials. The procedure was to start with one randomly selected sample in the build subset and put it into an accrual set. This accrual set was used to build the model (Gaussian parameters for

Table 1. Summary of nine data sets examined in the analysis of the effect of amount of data on classification performance. For the discrete data sets, two numbers are provided to describe amount of data to achieve optimal performance. Optimal performance comprises the peak value of the cost function defined in Sect. 2.3.3. For the continuous tracking data, the cost is subtracted from Eq. 1 for the Euclidean distance performance measure. For the discrete tasks, the amount of data needed to classify at a rate whose p value < 0.05 by the binomial test is shown in brackets

Data set	Task type	Number of cells	Amountof data (s) to reach optimal performance [to reach classification rate for binomial test $p < 0.05$]
t1070496	Discrete, two-direction	8	19 [4.0]
r1000606	Discrete, two-direction	7	18 [3.5]
r1010309	Discrete, four-direction	13	43 [8.0]
c1000201	Discrete, eight-direction	7	22 [12.5]
c1990702	Continuous tracking	18	170
c1990715	Continuous tracking	13	160
c1990716	Continuous tracking	17	180
e1990324	Continuous tracking	5	100
e1990421	Continuous tracking	4	130

discrete task, linear filters for continuous task), test it on all of the samples in the test subset, and record the average performance or error measure resulting from these tests. The next step was to add another randomly selected sample from the build subset to the previously established accrual set, build the models, and again test them. This process was repeated multiple times in order to increase the amount of data used to build the models.

By virtue of accruing build trials randomly, we avoided the possible confound of temporal proximity of build and test trials that would have been introduced had we accrued the trials sequentially. However, there remained another possible confound. For a given number of build trials, the resulting error measures could be related to both the effect of that amount of data and to the characteristics of the particular trials selected. To control for the effect of particular trials we proceeded to repeat the entire accrual process 100 times (this can be considered a type of bootstrap, Casella and Berger 2002). Each sample in the test set was predicted independently and the end results averaged together to provide a single measure of prediction accuracy on the entire test set.

A cost function was created to quantify the compromise between using more data to build a model and the resulting model's predictive accuracy. In terms of possible practical application of the algorithms, there are two costs: the computational requirements of building the models as the data set grows and the patient's ability and willingness to use the decoder as calibration time increases. As we here consider time scales of seconds to minutes, the processing time with current processor speeds of 500 to 1000 MHz to solve for Bayesian parameters or linear filters is negligible. There exists no standardized measure of cost for patient cooperation for this type of task. We propose here that human attention falls off exponentially (to the power of 2) with time for a given calibration task. Our cost function is necessarily arbitrary but provides an intuitive sense of time required to build a model that behaves reliably and accurately.

For the discrete task the cost function consisted of the number of samples used raised to the power of 2 and normalized and subtracted from the performance measure (fraction of the test set classified correctly):

$$\frac{\sum_{i=1}^{n} k_i}{n} - \frac{j^2}{m^2} \tag{4}$$

where there are n samples in the test set, and $k = 1$ if the direction (d) of the sample is predicted correctly ($\hat{d} = d$) and 0 otherwise ($\hat{d} \neq d$), j is the number of samples (500 ms each) used to build the model, and m is the maximum number of samples that could be used to build a model.

For the continuous-task cost function, analogous formulae were used, namely, by inserting the performance measure equations discussed above:

$$\{\text{Equation } \mathbf{1} +\} \text{ or } \{\text{Equations } \mathbf{2} \text{ or } \mathbf{3} -\} \frac{j^2}{m^2} \tag{5}$$

where j is the number of samples (10 s each) used to build the model, and m is the maximum number of samples that could be used to build a model.

2.3.4 Intervening time. We fixed the amount of data to build the models and varied the duration between the data used to build the model and the data that was classified in order to evaluate the effect of intervening time on reconstruction accuracy. We selected data from a subset comprising a fixed number of trials and shifted the starting point of this subset trial by trial across the entire experimental session. For each starting position, we used that particular set of trials to reconstruct the last 100 trials of the experiment. Because fatigue or other nonstationarities might introduce a systematic change in the relationship between neural activity and kinematics, we also used each trial set to reconstruct the first 100 trials of the experiment, as any trends common in both reconstructing the last and first 100 trials would be independent of fatigue. Based on preliminary findings regarding the effect of the amount of data on classification performance, the number of trials to build the discrete model was set at 20, while for the linear filters it was set at 100.

2.3.5 Across days. Because we were able to record from the same channels across subsequent days in the animal trained to perform the continuous tracking task, we performed a third analysis to determine how well filters built on one day could reconstruct position on another day. We assigned one neuron to each channel. If a channel displayed multiple units on waveform analysis, we summed all classified spike trains together. In our previous same-day analysis, we could use waveform classification to distinguish separate units on a given channel. We summed already classified spikes for the across-days analysis because of our inability to positively identify a given waveform across days on a channel that carries multiple waveforms from multiple units. It should be noted that even though the spikes are summed, the classification process does ensure that noise is filtered out.

For the across-days analysis, samples were defined as trials (as opposed to arbitrarily defined 10-s windows). To establish a baseline of the performance of the filters on a given day, we set aside one trial and used the remaining trials to build linear filters. We then used these filters to reconstruct position in the trial set aside and noted the error measures. We repeated this cross-validation procedure for each trial in the experiment on a given day to generate a distribution of error measures across all trials reconstructed.

To determine the ability of the linear filters to reconstruct across days, we built filters by using all available trials on one day and then used these filters to classify each trial one at a time on a data set from a subsequent or previous day. We then built histograms of the error measures for each trial generated on one day (cross validated) and another day. Because there were different numbers of trials each day, we plotted the histograms in terms of percentage of total number of

trials. We tested the two distributions for equality using a parameter-free Kolmogorov-Smirnov test.

3 Results

Four discrete-task data sets were examined (two two-direction, one four-direction, and one eight-direction). Increasing the amount of data used to derive the Gaussian distribution parameters caused a rapid improvement in model performance after which performance appeared to plateau (Fig. 1).

Adding trials improved performance from a baseline of 50% (chance level for two directions) to a maximum of 100% (Fig. 1a). As shown in Fig. 1b, performance improved up to a maximum of 95% in the four-direction data set, whereas Fig. 1c shows performance up to a maximum of 34% for the eight-direction data set. With the discrete-task cost function we chose, optimal performance occurred with 18 and 19 s in the two two-direction data sets, 43 s for four-directions, and 22 s for eight-direction data.

To evaluate the effect of varying the amount of time between the construction and application of the models, subsequent sets of 20 trials were used to construct models to classify both the last and first 100 trials in each data set. No overall trend related to absolute experimental time was observed. This procedure was tested for all the discrete-task data sets, and one representative example can be seen in Fig. 2.

For the continuous tracking task, eight data sets were used: five for the analysis of the amount of data and an additional three for the across-days analysis. Only trials in which the monkey tracked the target successfully >3 s were used for the construction of linear filters and the subsequent testing of the filters. Building filters with 1 to 2 min of data caused rapid improvements in reconstruction accuracy as measured by all performance measures employed as can be seen in Fig. 3. A slower trend of improvement followed.

Using the cost function based on the Euclidean distance, optimal performance was achieved, on average, with 160 s of data. The cost functions based on the other error measures peaked with approximately the same amount of data (ρ_x 90 s, ρ_y 100 s, r_x^2 120 s, r_y^2 130 s).

When the amount of data to build the models was fixed and only the time interval between the trials being used to build the model and the trials being reconstructed (within the same day) was varied, no consistent pattern emerged. The null hypothesis states that varying this interval has no effect on reconstruction error measures. As no overall trend was observed in reconstruction performance when the temporal duration between construction and application of the linear filters was varied, we cannot reject the null hypothesis.

Two pairs of continuous-task data sets were used to evaluate the ability of linear filters built one day to reconstruct position on a subsequent or previous day. One pair had one intervening day; the other pair spanned three intervening days. For both data sets, the distributions of Euclidean distance errors across days were not statistically different ($p < 0.05$, Kolmogorov-Smirnov test, Fig. 4). By this test the linear filters performed just as well on a subsequent day as on the day they were created. In addition, the Euclidean distance distributions from filters generated and tested on the later day

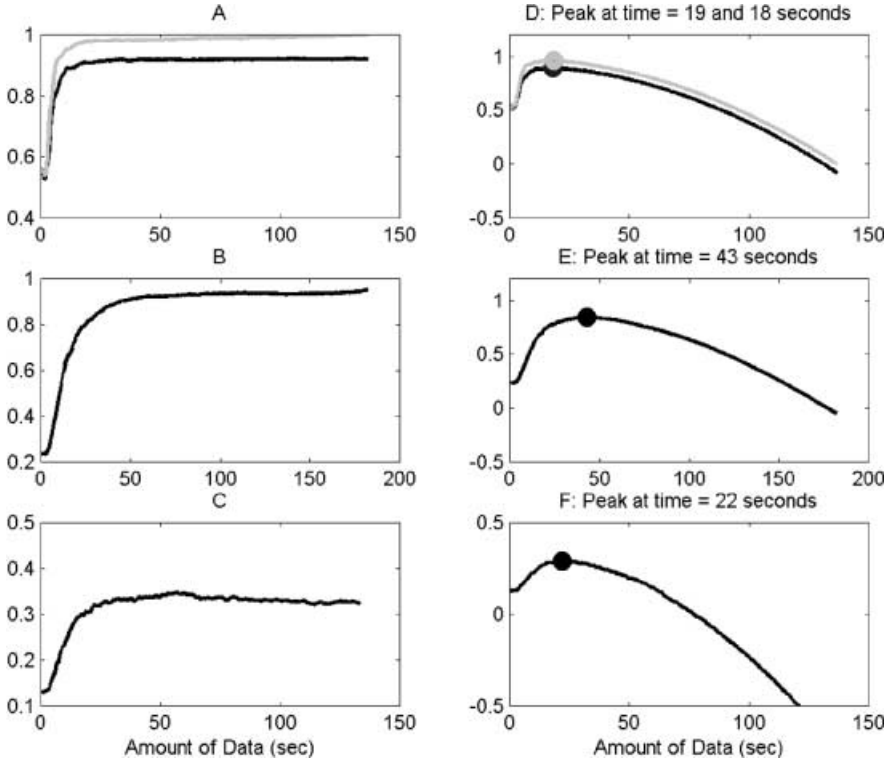

Fig. 1. Performance of discrete classifier with increasing amount of data (each trial = 500 ms) added to model. Ordinate axis represents fraction correct of last 10% of data set classified. In Fig. 1a, b, and c the *bold lines* show the mean performance of the classifier on 100 runs of available samples, for two-, four- and eight-direction data, respectively. Figure 1d, e, and f shows the result of the cost function and is titled with the amount of time needed for peak performance as defined by the cost function. The results of both two-direction data sets are shown in Fig. 1a and d

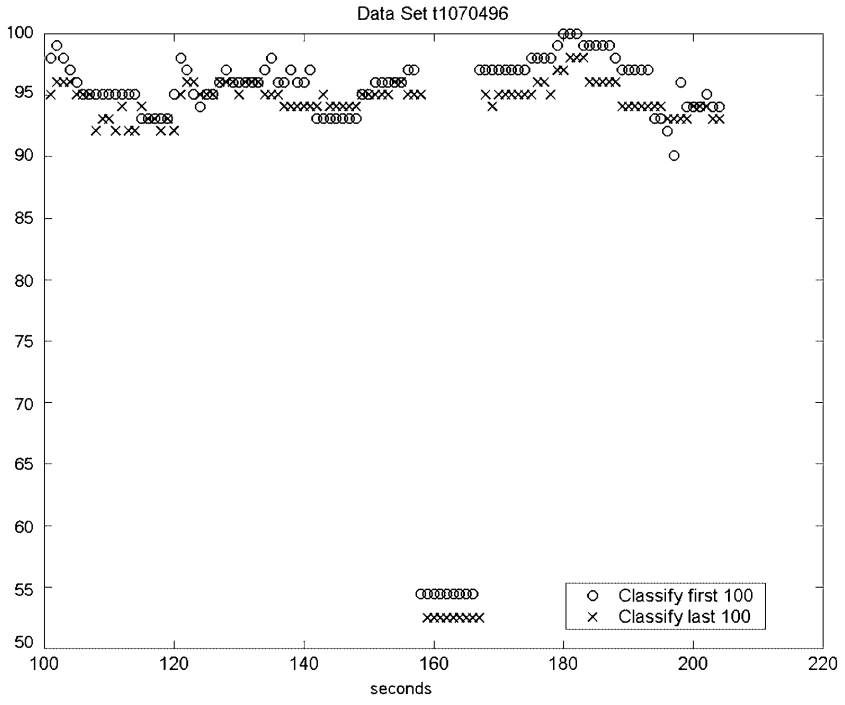

Fig. 2. Performance of discrete classifier as the time between model building and testing was varied. *Ordinate axis* represents percent correct of first or last 100 trials classified. Results from one two-direction data set are shown

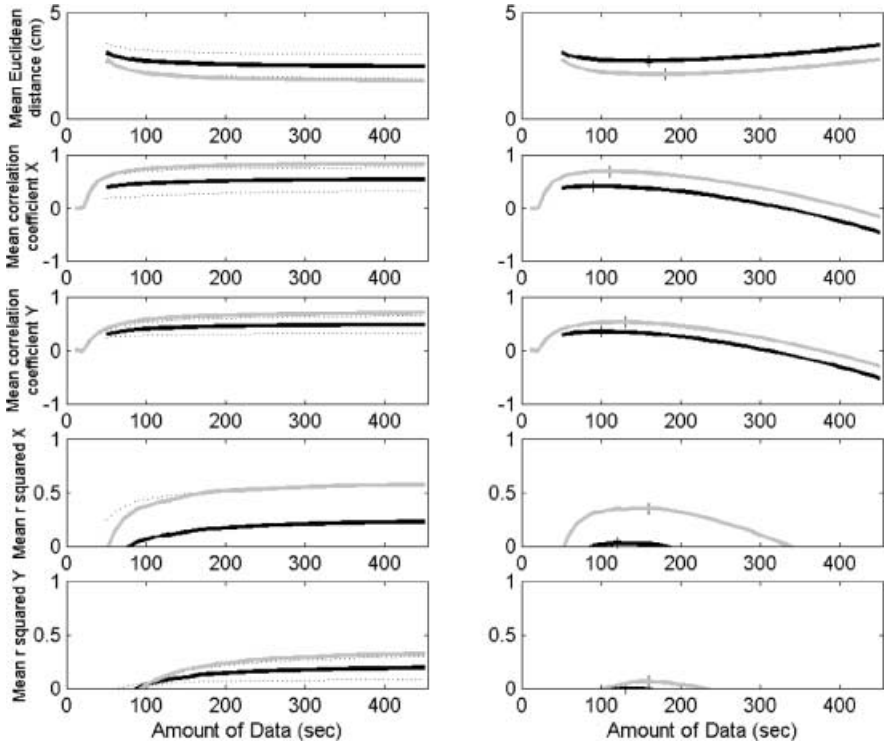

Fig. 3. Performance of linear filter with increasing amounts of data added to model. The left-hand column of graphs shows the mean raw performance measure across five data sets, whereas the right-hand column of graphs shows the cost function output for those performance measures. The *dark black line* represents the mean across the five data sets; the *dotted lines* show one standard deviation above and below the mean. The *gray line* displays the best-performing filter results for each measure and cost function. The *tick marks* on the left-hand graphs represent the peaks of the cost-function output (or trough in the case of Euclidean distance). Filters built upon just the first few trials produce performance measures that are undefined and hence not plotted; in the case of r^2 values for x and y, negative values are not plotted as they do not provide additional information beyond the fact the filters do not perform well in that regime

resulted in similar distributions when tested on the earlier day ($p < 0.05$, Kolmogorov-Smirnov test).

4 Discussion

This study addressed the following questions: (1) How much data is sufficient to build effective decoding algorithms for discrete and continuous classifiers? and (2) How stable is the performance of a decoding model over time within a day and across days? A large time requirement to create decoding models as well as the need for frequent, extensive recalibration would severely limit the practical application of neuroprosthetic devices.

Our results suggest that approximately 1 min of data is sufficient to build maximum-likelihood models, and 3 min of data are needed to build linear filters that reconstruct hand trajectory $(x(t), y(t))$ at the optimal level as judged by performance measures based on Euclidean distance, correlation coefficient, or the r^2 values. Since

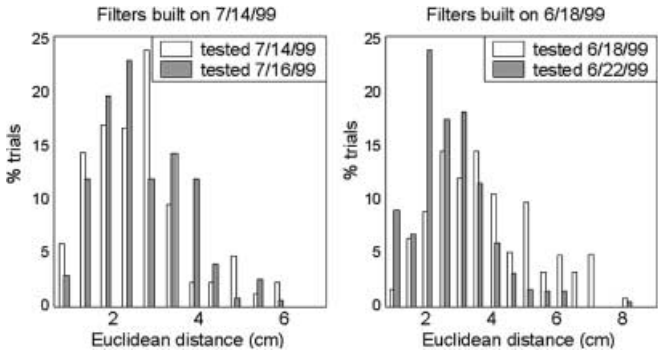

Fig. 4. Performance of linear filters across days. Filters were built on one day in a cross-validated form to generate the "same-day" distribution and were built on that entire day's data and tested on each trial of a subsequent day to generate an "across-days" distribution. As there were different numbers of trials completed each day, distributions were normalized to permit comparison. The error measures were placed into 0.5-cm bins; the *ordinate* on the top graphs displays the percent of the total trials on the tested day. The distributions across the two days sampled were not significant ($p < 0.05$, Kolmogorov-Smirnov test)

Euclidean distance combines horizontal and vertical information, we feel it may be the preferable error measure for comparing performance of the linear filters as construction data accrued.

We cannot conclude that discrete direction or continuous position is encoded in these neurons, only that we may recover these movement parameters from them. Because hand position covaries with joint angles, forces, and other aspects of arm movement, any one of these dimensions, or in fact multiple dimensions, may be encoded (Todorov 2000; Mussa-Ivaldi 1999; Reina et al. 2001) in the neural activity of primary motor cortex neurons. Neural activity may reflect or participate in the control of these variables sequentially or simultaneously.

Performance of the classifier was closely tied to the particular trials the model was built upon, as seen when the classification rate for each set on the last 100 and the first 100 trials are compared (Fig. 2). Two factors may account for this phenomenon: (1) certain 20-trial windows may happen not to contain many trials in a particular direction, thereby prohibiting proper construction of conditional probabilities in that direction, and (2) the neurons in particular 20-trial windows may fire in an aberrant manner relative to their average conditional firing rate, causing a kind of model overfitting to noncharacteristic trials.

In a practical neuroprosthetic application, it would be desirable to minimize the amount of time needed to recalibrate the control signal each day. Patients would probably tolerate a daily calibration session of 15 to 20 min better than hourly calibrations of 2 min. Our results have quantified how the classifier behaves within the 40 min examined and using from 7 to 18 neurons.

Though adding more trials of data to model construction does not improve performance after a certain point, adding more cells within that data may. While we used all neurons present in each data set for the current analysis, we hope that more would be available in a chronically implanted neuroprosthetic array. Wessberg et al. (2000) suggested that 600 to 900 neurons would provide optimal encoding. Though Paninski et al. (2003) showed that improvements occur as the number of cells is increased, they also showed that the data currently available cannot resolve this issue entirely because it is difficult to accurately extrapolate to very large numbers of cells adequately. Paninski et al. (2003) concluded that some subsets of cells are more informative than others. In addition to adding more neurons (to increase the likelihood of including informative ones), incorporating other aspects of the signal (e.g., fine temporal structure) or other types of signals (e.g., local field potentials) may also improve performance.

Increasing the amount of time intervening within a day between the data used to build the model and those used to test it did not affect performance of either the maximum-likelihood discrete direction classifier or the continuous position reconstruction by the linear filters. For the discrete classifier, performance was closely tied to the particular trials the model was built upon. The ability of the model to classify discrete states (such as direction of movement) well over time depends on a stable relationship of neural firing rates with their particular representation of that kinematic parameter. Increasing the number of trials may prevent overfitting by exposing the model to a greater range of firing rates per given direction. Just as certain trials that are used to build the model may be outliers, so too may certain trials that are being classified. Trials with neural firing patterns far enough from the mean activity tend to be poorly classified, independent of the amount of data the models are built upon. While our current analysis could not detect any systematic changes in how neurons encode kinematic parameters (discrete or continuous) over time, future studies may reveal more subtle transformations.

Linear filters built on one day were found to carry over their ability to reconstruct position on subsequent days. To our knowledge this is the first report of a small population of neurons found to represent a kinematic parameter in a stable fashion across several days. Moran et al. (2000) found that an individual neuron could represent kinematic parameters in a consistent manner across years. This report extends that finding to a small population of neurons, considered over a period of several days. Note that ours is a functional assay of population stability: rather than examining the waveforms or tuning properties of particular single units, here we observe the preservation of population-wide coding.

Further study is needed to determine how different subsets of these populations can maintain their representation across various time scales, as the loss of a neuron over time (due to electrode shifting or neural plasticity) may occur in multielectrode arrays implanted in humans. Preliminary results from our lab indicate that linear filters generated on an earlier day are able to provide functionally useful reconstructions on a later day even if several neurons are no longer recordable on the later day. We believe this is a consequence of the stability of population-wide coding. In the analysis of this report, neurons in common across days were defined

only by being on the same channel. If the particular neurons being recorded across days had in fact changed, these results would suggest that the neurons clustered around a particular electrode tip share coding properties. Such a feature could help to bring stability to neuroprosthetic devices. From a practical standpoint, the ability to apply linear filters from previous days would entail less daily calibration time: a patient may be freed up from having to generate new filters each day.

The experiments carried out here describe the ability of two different types of decoding algorithms to be used to generate a control signal from multiple single-neuron activity. To ascertain under what circumstances which of the two signals would be most desirable, we must consider the strengths and limitations of each.

The length of the linear filters (i.e., how much of the immediate history of each neuron's firing rate is considered) strongly affects reconstruction. There is a tradeoff between performance and the delay in reconstruction. Paninski et al. (2003) found that increasing the temporal length of filters in the linear model improves reconstruction performance, though as more of the history of the firing rate of each neuron is considered, the amount of improvement in the reconstruction decreases. This tradeoff reveals how certain neurons are sensitive to movement information over a long time window. The discrete classifier suffers from an analogous problem: the more time required to count spikes after the go cue, the less the control signal can predict movements rapidly.

However, Hatsopoulos et al. (2001) found that despite the decrement in performance that occurs when counting in shorter time windows after the go cue, performance rates of up to 85% (in a two-direction task) were possible with only 10-ms windows. A discrete classifier may thus be preferable over linear filters for classifying movement at shorter time scales. The discrete classifier, while reconstructing kinematics in a much lower dimension than the linear filters, is more efficient in the short-time-window regime.

The decision of which model to employ can be made by either the patient or a supervisor algorithm that takes note of features of brain activity. Local field potentials (LFPs) change dramatically with the onset of a step-tracking movement (Donoghue et al. 1998). LFPs could thus be used as a supervisor signal informing the computer which model to apply to the incoming neural data. Multiple algorithms analyzing multiple input signals could be incorporated into one large framework to allow a patient the greatest possible range of independent activities (Fig. 5).

We propose that an optimal control signal could use both discrete and continuous forms of decoding simultaneously. With current processing power, these are not difficult to compute rapidly. New algorithms that compare the classification of the two could help to ensure that the desired outcome is achieved. Determining the optimal combination of discrete and continuous control will be a major challenge in the development of neuroprosthetic devices. Research with implanted muscle stimulators has already shown that patients apply control strategies in unanticipated ways: given 100 possible intermediate hand-grasp positions achieved by functional electrical stimulation and controlled by a shoulder mounted switch, they end up using only three; yet they use those three in a wide range of movement (Hart et al. 1998; Crago et al. 2000).

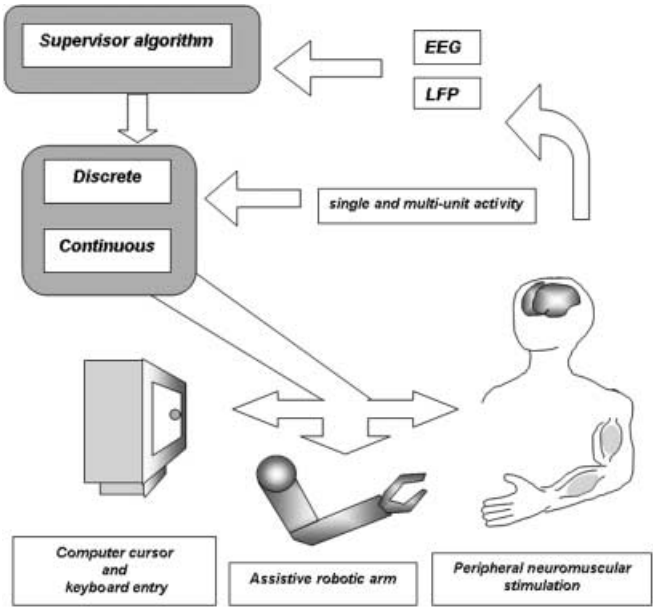

Fig. 5. Scheme by which multiple brain-derived signals can be integrated to control several assistive devices. Electroencephalographic and local field potential signals can be used both as elements of a supervisor algorithm to decode single-unit and multiunit activity and as voluntarily controlled toggle switches

These control signals could readily be applied to a number of real-world devices to restore function to paralyzed individuals. A robotic arm could be controlled by either having its endpoint continuously updated to the intended X, Y, and Z position, or a user could drive it with a series of discrete direction commands that would launch the arm inertially in the classified directions, much like using a joystick. We could envision the following implementation of a dual-signal strategy: to reach for and grasp a toothbrush, a robotic arm could be controlled by a series of ballistic discrete commands. Once in the right neighborhood, a patient could switch to continuous control for the actual brushing of the teeth.

For the discrete and continuous tasks discussed in this study, the requirement for gross movement could be substituted by instructions to attempt movement of a specific limb in a pattern displayed by a stimulus such as a cursor on a screen (Kennedy et al. 1998), a robotic arm end- position (Wessberg et al. 2000), or even their own arm via electrodes implanted into peripheral nerves and muscles (Lauer et al. 1999). As fMRI and PET studies have revealed different metabolic activity patterns across patients, depending on whether they were asked to "imagine" rather than actually attempt movement, the specific instructions and the patient's comprehension will be crucial during the calibration (Roland et al. 1980; Lotze et al. 1999). The details of the instruction may be crucial for future use of the device: if the conscious

percept of the device control is experienced to be very effortful, patients may be more reluctant to use the device and prefer standard alphabet boards and eye blinks to communicate.

One significant outcome of the present study is that, while daily calibration of a control signal would appear to be necessary if used for a human motor neural prosthetic, the calibration period is very short. It is reasonable to consider that a supervisor algorithm could be implemented to recognize that performance with the previous day's model is already very good and no further calibration is needed.

The experiments carried out here describe the possibility of using two different types of decoding algorithms to generate a control signal from multiple single-neuron activity. While the number of cells and experimental duration in our study are smaller than what we would hope would be developed in a device for human use, our findings establish calibration parameters and demonstrate the feasibility of using neurons as reliable control signals.

Acknowledgements. The authors wish to thank Elie Bienenstock, Michael Black, and Yun Gao for their helpful suggestions on the mathematical analysis. This work was supported by the U.S. National Institutes of Health (grants NIH #R01 NS25074 and NIH #N01-NS-9-2322 and NIMH K01 MH01671) and DARPA (grant MDA9720010026). The authors MS, NH and JP, wish to declare that they are co-founders and shareholders in Cyberkinetics, Inc.

References

Casella G, Berger RL (2002) Statistical inference, 2nd edn. Wadsworth Group, Pacific Grove, CA

Crago P, Kilgore K, Lauer R, Peckham H, Kirsch B (2000) Functional neuromuscular stimulation (FNS). Thirty-first neural prosthesis workshop. National Institutes of Health

Donoghue JP, Sanes JN, Hatsopoulos NG, Gaal G (1998) Neural discharge and local field potential oscillations in primate motor cortex during voluntary movements. J Neurophysiol 79: 159–173

Foltys H, Kemeny S, Krings T, Boroojerdi B, Sparing R, Thron A, Topper R (2000) The representation of the plegic hand in the motor cortex: a combined fMRI and TMS study. Neuroreport 17 11(1): 147–50

Georgopoulos AP, Schwartz AB, Kettner RE (1986) Neuronal population coding of movement direction. Science 233: 1416–1419

Hart RL, Kilgore KL, Peckham PH (1998) A comparison between control methods for implanted FES hand grasp systems. IEEE Trans Rehabil Eng 6(2): 208–218

Hatsopoulos NG, Ojakangas CL, Paninski L, Donoghue JP (1998a) Information about movement direction obtained from synchronous activity of motor cortical neurons. Proc Natl Acad Sci USA 95(26): 15706–15711

Hatsopoulos NG, Ojakangas CL, Maynard EM, Donoghue JP (1998b) Detection and identification of ensemble codes in motor cortex. In: Eichenbaum H, Davis J (eds) Neuronal ensembles: strategies for recording and decoding. Wiley, New York, pp 161–175

Hatsopoulos NG, Harrison MT, Donoghue JP (2001) Representations based on neuronal interactions in motor cortex, chap 15. In: Progress in brain research, vol 130. Elsevier Science BV, Amsterdam

Helms Tillery SJ, Taylor DM, Isaacs R, Schwartz AB (2000) Online control of a prosthetic arm from motor cortical signals. Abstract, Society for Neuroscience

Humphrey DR, Mao H, Schaeffer E (2000) Voluntary activation of ineffective cerebral motor areas in short- and long-term paraplegics. Abstract, Society for Neuroscience

Kennedy PR, Bakay RA (1998) Restoration of neural output from a paralyzed patient by a direct brain connection. Neuroreport 9(8): 1707–1711

Kennedy PR, Moore MM, King B (2000) Directionality coding in human cortical area 4: role of phase relationships of individual action potentials. Abstract, Society for Neuroscience

Lauer RT, Peckham PH, Kilgore KL (1999) EEG based control of a hand grasp neuroprosthesis. Neuroreport 10(8): 1767–1771

Lauer RT, Peckham PH, Kilgore KL, Heetderks WJ (2000) Applications of cortical signals to neuroprosthetic control: a critical review. IEEE Trans Rehabil Eng 8(2): 205–207

Lotze M, Montoya P, Erb M, Hulsmann E, Flor H, Klose U, Birbaumer N, Grodd W (1999) Activation of cortical and cerebellar motor areas during executed and imagined hand movements: an fMRI study. J Cogn Neurosci 11(5): 491–501

Maynard EM, Nordhausen CT, Normann RA (1997) The Utah intracortical electrode array: a recording structure for potential brain computer interfaces. Electroencephalogr Clin Neurophysiol 102: 228–239

Maynard EM, Hatsopoulos NG, Ojakangas CL, Acuna BD, Sanes JN, Normann RA, Donoghue JP (1999) Neuronal ineractions improve cortical population coding of movement direction. J Neurosci 19(18): 8083–8093

Maynard EM, Fernandez E, Normann RA (2000) A technique to prevent dural adhesions to chronically implanted microelectrod arrays. J Neurosci Meth 97(2): 93–101

Moran DW, Reina GA, Schwartz AB (2000) Long term analysis of a single motor cortical cell. Abstract, Society for Neuroscience

Mussa-Ivaldi FA (1999) Modular features of motor control and learning. Curr Opin Neurobiol 9: 713–717

Paninski L, Fellows MR, Hatsopoulos NG, Donoghue JP (1999) Coding dynamic variables in populations of motor cortex neurons. Abstract, Society for Neuroscience

Paninski L, Fellows MR, Hatsopoulos NG, Donoghue JP (2003) Temporal tuning properties of motor cortical neurons for hand movement. J Neurophys (in review)

Reina GA, Moran DW, Schwartz AB (2001) On the relationship between joint angular velocity and motor cortical discharge during reaching. J Neurophysiol 85(6): 2576–2589

Robinson CJ (2000) A commentary on brain computer interfacing and its impact on rehabilitation science and clinical applicability. IEEE Trans Rehabil Eng 8(2): 161–162

Roland PE, Larsen B, Lassen NA, Skinhoj E (1980) Supplementary motor areas and other corticl areas in organization of voluntary movements in man. J Neurophysiol 43: 118–136

Serruya MD, Shaikhouni AM, Hatsopoulos NG, Donoghue JP (2001) Algorithms for realtime control of a motor neuroprosthetic device. Abstract, Society for Neuroscience

Serruya MD, Hatsopoulos NG, Paninski L, Fellows MR, Donoghue JP (2002) Instant neural control of a movement signal. Nature 416(6877): 141–142

Shoham S, Halgren E, Maynard EM, Normann RA (2001) Motor-cortical activity in tetraplegics. Nature 413(6858): 793

Taylor DM, Tillery SI, Schwartz AB (2002) Direct cortical control of 3D neuroprosthetic devices. Science 296(5574): 1829–1832

Todorov E (2000) Direct cortical control of muscle activation in voluntary arm movements: a model. Nat Neurosci 3(4): 307–308

Warland DK, Reinagel P, Meister M (1997) Decoding visual information from a population of retinal ganglion cells. J Neurophysiol 78: 2336–2350

Wessberg J, Stambaugh CR, Kralik JD, Beck PD, Laubach M, Chapin JK, Jim J, Biggs SJ, Srinivasan MA, Nicolelis MA (2000) Real time prediction of hand trajectory by ensembles of cortical neurons in primates. Nature 408(16): 361–365

Williams JC, Rennaker RL, Kipke DR (1999) Long term neural recording characteristics of wire microelectrode arrays implanted in cerebral cortex. Brain Res Brain Res Protoc 4(3): 303–313