Extraction algorithms for cortical control of arm prosthetics

Andrew B Schwartz*†, Dawn M Taylor† and Stephen I Helms Tillery†

Now that recordings of multiple, individual action potentials are being made with chronic electrodes, it seems that previous work showing simple encoding of movement parameters in these spike trains can be used as a real-time control signal for prosthetic arms. Efficient extraction algorithms can compensate for the limited ensemble sample acquired with this emerging technology.

Addresses
*The Neuroscience Institute, 10640 John Jay Hopkins Drive, San Diego, CA 92121, USA; e-mail: aschwartz@nsi.edu
†Department of Bioengineering, Arizona State University, Tempe, AZ 85287-9709, USA
Correspondence: Andrew B Schwartz

Current Opinion in Neurobiology 2001, 11:701–707

0959-4388/01/$ – see front matter

Abbreviations

2D	two-dimensional
3D	three-dimensional
ANN	artificial neural network
ML	maximum likelihood
PDF	probability-density function
PVA	population vector algorithm
SOFMs	self-organizing feature maps

Introduction

The will to move and how this desire is carried out have been major philosophical and scientific issues throughout written history. During the past few decades, our ability to examine cortical activity directly has enabled the covert processes that precede volitional movement to be investigated scientifically. Now, as our understanding of these processes has advanced, we can apply our new knowledge to individuals who have the desire to move but cannot. The field of neuroprosthetics embodies a migration from basic to applied science.

Of current interest are methods to intercept and interpret neural signals in an effort to generate control signals for arm actuators. Extraction algorithms for motor control operate on spike trains recorded from a population of cortical units with the purpose of predicting arm trajectories. These methods were developed mainly to model putative brain mechanisms and recognize represented movement parameters in intact primates. Accurate trajectories are produced using current extraction algorithms in intact subjects, with the idea that, if computationally efficient, they will successfully generate robust signals for prosthetic control in subjects who cannot move their arms.

To form the trajectory prediction, the desired movement parameter is derived from a sample of firing rates. If, for example, the chosen parameter is movement direction θ, we need to calculate an estimate of movement direction, $\hat{\theta}$, given a sample of firing rates, **a**, from different cells. Although the activity of a single cell does have some predictive power, the single-unit relation to movement is too coarse and populations of activity are needed to provide the kind of speed and accuracy for control of prosthetic devices. Here, we review different extraction algorithms that might be useful for extracting control signals from a population of simultaneously recorded single units.

Linear methods

Modulation in motor cortical neuron firing rate often has an almost linear relationship to movement kinematics. Therefore, linear equations are commonly used to describe expected cell behavior.

Population vector algorithm

Restating and generalizing the original directional findings of Georgopolous *et al.* [1,2] to three dimensions leads to the following linear equation for a single cell (Figure 1):

$$D - b_0 = b_x m_x + b_y m_y + b_z m_z \quad (1)$$

where D is the cell's discharge rate, b_0 is its mean discharge rate, m_x, m_y and m_z are the x, y and z components of a unit vector pointing in the direction of movement, and b_x, b_y and b_z are regression coefficients. This is equivalent to the dot product expression:

$$\mathbf{B}\cdot\mathbf{M} = b_x m_x + b_y m_y + b_z m_z \quad (2)$$

where **B** can be conceptualized as a vector pointing in the cell's preferred direction, that is, the direction where the cell fires maximally. The magnitude of **B** is equal to the cell's maximum increase in discharge rate above the mean. **M** is a unit vector in the movement direction (consisting of m_x, m_y and m_z). This linear relation can also be expressed in terms of a cosine-tuning equation:

$$|\mathbf{B}|\,|\mathbf{M}|\cos\theta \quad (3)$$

where θ is defined as the angle between the preferred and movement directions. This is a wide tuning function with one central peak [3]. An example of this function is shown in Figure 2a. Although these equations describe single-cell activity during movement, they are too ambiguous and coarse for accurate predictions of movement velocity.

The cosine tuning function is wide, encompassing all movement directions. Recent, more detailed, experiments [4] show, however, that these tuning functions may be more narrow than the cosine function. Nonetheless, the peak of the tuning function is used to catgorize each cell's contribution to the ensemble's movement representation in the population vector algorithm (PVA). The i^{th} contribution, $\mathbf{C}_i$, to the population output is represented as a unit vector pointing in its preferred direction, and weighted by some function of its firing rate, $w_i = f(D)$. Several different

weighting formulae have been explored [5]. The weighted cell vectors are then summed across all cells to form the population vector, **P**, which points in the predicted direction of movement.

$$\mathbf{P} = \sum_i C_i . Wi \quad (4)$$

If the cells are truly cosine-tuned and their preferred directions are uniformly distributed, then performance of the PVA is equivalent to a maximum likelihood (ML) estimation ([6]; and see below) under uniform variance conditions. As long as the cells have radially symmetric tuning functions, the expected value of the PVA will still be accurate given a set of uniform preferred directions [5]. But the uniformity constraint is seldom met, and the quality of tuning can vary greatly with a small sample of unit recordings.

Under these conditions, several measures can be taken to improve the PVA further. For example, each unit's contribution to **P** can be scaled by some measure of its quality of tuning (such as using **B** from Equation 2 above instead of the unit vector **C**). A non-uniform distribution of tuning directions can be addressed by normalizing the magnitude of each component of **P**. If the cells are truly cosine-tuned, this is easily done by dividing the p_x, p_y and p_z values respectively by the magnitudes of c_x, c_y and c_z (or of b_x, b_y and b_z) summed across all cells. Compensation for non-uniform preferred directions, and for the PVA dependence of symmetrical tuning functions is a benefit of the Optimal Linear Estimator developed by Salinas and Abbott [7]. Instead of defining a cell's preferred direction as the peak of a cosine function, the tuning function's center of mass is used. Each cell's contribution to the population vector is weighted by its lack of correlation with the tuning functions of the other contributors, effectively correcting for non-uniformities.

Optimal linear filters

The PVA has been used to predict two-dimensional (2D) and three dimensional (3D) reach direction by applying it to the mean activity of the cell during each movement. It has also been used to generate trajectories during reaching and drawing [8–11], by dividing the cell activity into bins and calculating a population vector for each bin. A weakness of PVA is that it does not take full advantage of the temporal characteristics of motor cortical discharge. Optimal linear filters make use of these characteristics to predict specific trajectories from parallel time series of cell activity ([12•]; and L Paninski, M Fellows, NG Hatsopoulos, JP Donoghue, personal communication). An optimal linear filter has the general form:

$$P(t) = \sum_{i=0}^{T_{pre}/dt} \sum_{j=1}^{C} a_{ij} \cdot N(t + i \cdot dt, j) \quad (5)$$

where P is the predicted parameter at time t, dt is a negative interval, i iterates back in time to T_{pre}, a is a fitted coefficient and N is the firing rate for cell j at time $t + i \cdot dt$. This method is especially useful for fitting data in which there is a significant lag between the discharge rate and P. It is also clear that there is information, useful for making the prediction, present in the spike train for a significant interval before time t in the movement. A separate coefficient is regressed for each lag and cell. In the procedure used by Wessberg *et al.* [12•], position along a 3D trajectory was predicted by finding a separate coefficient for each dimension, lag and cell.

Other methods

One limitation of linear filter methods is that they rely on an *a priori* model of the movement-related neuronal responses. Deviations from the model will occur if unaccounted sources of spike modulation are significant. These extra sources of variance may be consistently related to the movement and thereby carry predictive information. The remaining methods make no prior assumptions about discharge-movement relations beyond the selection of an output parameter space.

Maximum likelihood estimation

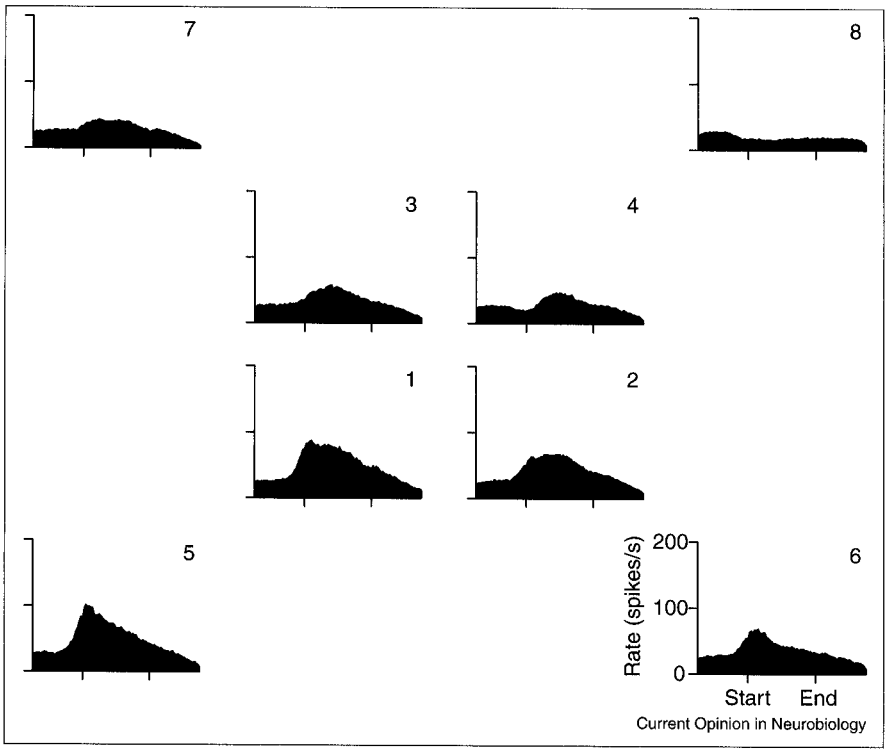
Using a chronic recording technique, we recorded a motor cortical unit as a monkey moved to eight targets from the middle of a cube to its eight corners [13]. The movement-aligned histograms are shown in Figure 1, and these firing rates when plotted against movement direction could be fitted to a cosine function. Statistically, the tuning relation can be considered as a likelihood function. This function gives the probability of discharge rate, a, for each movement direction, θ; $f(a|\theta)$. The ML estimate is the θ that maximizes the probability of observing the rate a. The tuning function shown in Figure 2b is a 2D projection of the more general 3D condition shown in Figure 2a and provides a straightforward means of explaining the ML method. Taking a single firing rate, a, on the ordinate, we can go across the figure, counting directions corresponding to each sample of this discharge rate. We then divide each of these observations by the total number of samples in the observed direction to get $P(a|\theta)$.

For illustration, this was performed in three activity bands — low, medium and high rates — instead of using individual firing rates. The three panels of Figure 2c show that movement directions 180° away from the preferred direction are likely to occur when the firing rate is low, a wide range of directions within 135° of the preferred direction are likely for moderate rates, and movements near the cell's preferred direction are most probable for high rates.

What we really want for prediction is $P(\theta|a)$. This can be generated either indirectly by the 'forward' process used in Figure 2c or directly by calculating the percentage of movement directions that occur for each firing rate. This was done, preserving the 3D information embedded in the data, in Figure 3. Firing rates were again divided into three ranges and a probability-density function (PDF) surface constructed in polar coordinates. The directions of each target are shown by the numbered circles.

Figure 1

Discharge profiles of a unit recorded during a 3D center-out task. The workspace consisted of a cube with targets at each of the eight corners. Each movement began in the center of the cube and ended at one of the corner targets. Histograms are arranged by target location. The middle four histograms correspond to the distal, far face of the cube, and the outer four to the proximal, near face. The unit was recorded with a chronic technique and the data consisted of 490 repetitions to each of the eight targets. The cell's preferred direction points between targets 1 and 5.



For low rates it can be seen that movements toward targets 6, 7 and 8 are the 'winners,' with target 8 (near, up and to the right) being the most likely. This is the target that is most nearly opposite the preferred direction and corresponds to the histogram with the lowest rate in Figure 1. For middle firing rates, targets 3, 4, 6 and 7 are the winners. Movements near targets 3 and 7 (up and to the left) are the more likely. Finally, the surface shows peaks near targets 1, 2, 3, 5 and 6 for the highest rates, with the most likely direction between targets 1 and 5 (down and to the left). This corresponds to the preferred direction.

The forward method shown in Figure 2 has the advantage that the PDF can be fitted with a basis function that is usually cosine-like [14], allowing estimates of poorly sampled directions to be made. This is shown, for instance, by the data gap around the preferred direction in Figure 2b. Technically, either conditional probability can be derived from the other using Bayes Theorem, if $P(\theta)$ and $P(a)$ are known.

For accurate prediction, the PDFs of individual neurons, such as those illustrated for the above unit, must be combined with those of other units recorded simultaneously to form a joint probability distribution, $P(\theta|\mathbf{a})$. If the firing rates were independent, the PDFs for each unit could be multiplied together to get the joint probability [15]. Ideally, this method would work by calculating $P(\theta|a)$ for every activity level for each cell to form a 'dictionary'. Then when a novel **a** is sampled, the individual firing rates would be used as an index to each PDF in the dictionary. The selected PDFs would then be multiplied and the maximum of the resulting joint probability taken as the most likely movement direction for the population.

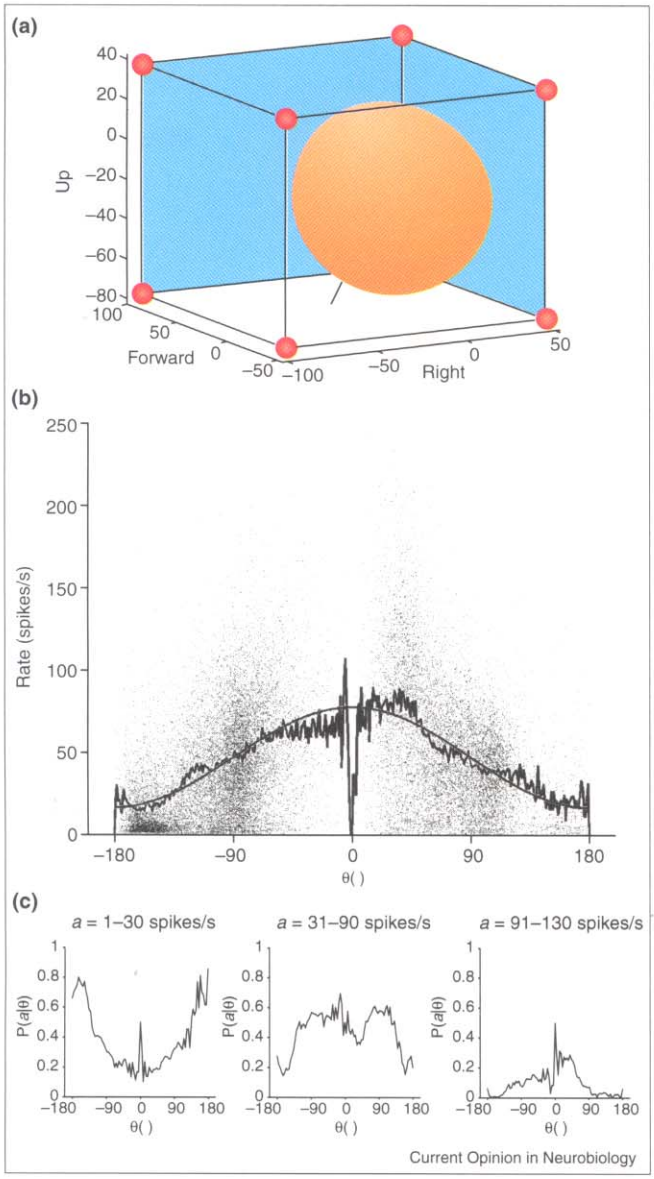
Theory suggests that the ML method might be optimal in terms of extracted information [16,17]. One factor determining the utility of this method is whether PDFs can be combined together to form the joint probability distribution for the population. If, for example, the firing rates were independent, then the individual PDFs could be multiplied together. However, correlations between spike trains from different neurons may occur [18,19], and these can change the movement-related information. This should be taken into account when forming the ensemble estimate.

Pattern recognition

A snapshot of firing rates across a population can be considered a pattern. If the variation in this pattern is distinct with changes in movement parameters, pattern recognition algorithms can be used for parameter extraction. For practical applications, this requires an algorithm that is efficient or that can express pattern variation with a minimal number of factors. By taking advantage of co-variation between pattern elements, principal components analysis produces factors representing maximum and independent proportions of variance in a data set [20,21] to reduce the complexity of the pattern.

As an extension of the snapshot analogy, information about each parameter can be found in the activity that takes

Figure 2



Direction–activity relationship. **(a)** The 3D tuning function for the unit shown in Figure 1 is plotted in Cartesian space. This was calculated using Equation 1 in a multiple regression: $b_0 = 47.2$, $b_{forward} = 15.6$, $b_{up} = -22.6$, $b_{right} = -13.0$. The axes are in units of spikes/s. The distance from the origin to the edge of the volume gives the predicted discharge rate in the chosen movement direction. The preferred direction is indicated by the line through the tuning volume pointing downward and to the left (as the monkey would view the workspace). The preferred direction is between targets 1 and 5. **(b)** Plot of discharge rate against the angle between the movement and preferred direction, θ. Movement direction and discharge rate were calculated about every 48 ms for all data, resulting in 39,191 samples shown here as dots for the unit illustrated in the previous figures. The smooth line is the rate predicted by the cosine function in Equation 3. The jagged line is the mean activity for each value of θ, $E(a|\theta)$. Few movements were made within 20° of the preferred direction resulting in a data gap for these directions. **(c)** The firing rates were divided into three ranges (1–30, 31–90 and 91–130 spikes/s and the probability of having a rate in this range, given θ calculated by counting the instances of that firing rate in each direction and dividing by the total number of movements in that direction.

place before the parameter changes; a sequence of pattern snapshots carries more information than a single snapshot. This is true of single-cell activity patterns where, for example, the shape as well as the height of the peri-event histogram is related to target direction (Figure 1).

This was the motivation for the technique developed by Isaacs *et al.* [22•]. Activity windows consisting of 200 ms of firing rates from each of N neurons in a simultaneously recorded population were placed in a data table. The window consisted of 10 bins and the firing rates for each bin of the window were placed down the column of the table. Simultaneous windows from each unit were concatenated so that each column comprised $N \times 10$ firing rates. Each subsequent column consisted of windows that had been shifted by one bin from those of the previous column. Rows of the table were ordered sequences of firing rates from a single cell. These rows were treated as vectors and placed in a correlation matrix from which eigenvectors and eigenvalues were calculated. The eigenvectors were multiplied by each column of the data matrix to give a dictionary of principal components.

This extraction algorithm worked by taking 200 ms of novel ensemble data, multiplying by the eigenvectors and then comparing these with the dictionary of principal components. The closest match identified a column in the data matrix corresponding to a particular window and movement vector associated with it. This analysis can be quite efficient; for example, eight principal components accounted for more than 90% of the variance in a data table consisting of 240 rows.

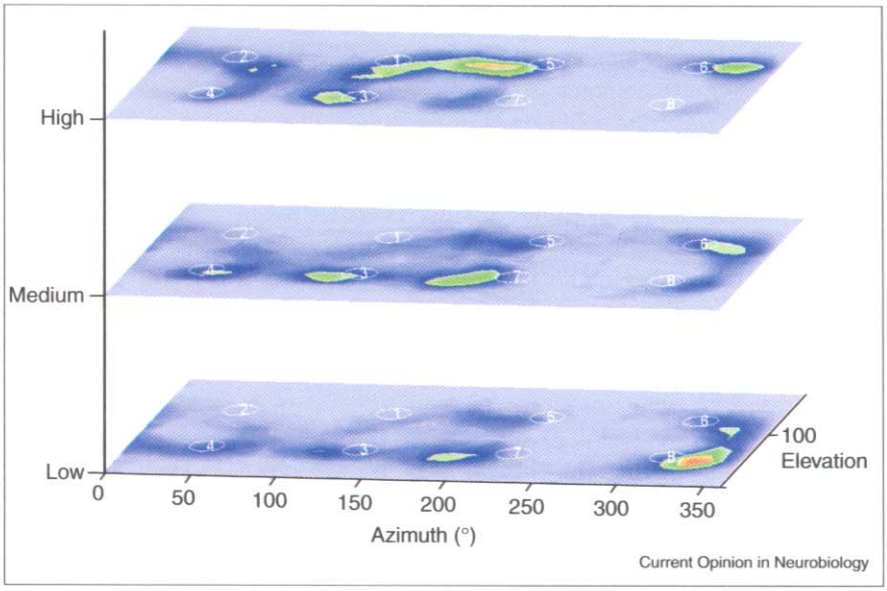
Neural networks

Artificial neural network (ANN) solutions can optimize each cell's contribution to the population prediction. Approaches ranging from self-organizing feature maps (SOFMs) to complex nonlinear recursive networks have been used as a means of mapping movement parameters to activity patterns.

SOFMs [23] are generally constructed as a single layer of nodes, each of which is initialized with a random vector of n weights, where n is the dimensionality of the input. The SOFM is tuned by comparing an input vector to the weight vector for each node. In a 'winner takes all' scheme, the node with the weight vector closest to the input gets its weights modified to be slightly closer to the input. Over training, separate input classes come to be selected by different nodes in the SOFM.

An SOFM has been used to construct arm trajectories from neural data [24]. In this analysis, the SOFM comprised 20×20 nodes. Input vectors were constructed from arrays of firing rates recorded across the ensemble. After training on a set of neural data, the SOFM was organized into an output topography, in which neighboring nodes corresponded to inputs of similar firing patterns. Trained with

Figure 3



Probability of movement in direction θ given a firing rate. Firing rate and 3D movement direction were calculated in intervals consisting of 24 ms (78,400 samples) as the arm moved during each trial. Movement direction was calculated as the difference between hand position at the start and end of movement. These were used to calculate PDFs for movement direction across each sample. The number of times a movement direction occurred in association with a given discharge rate was divided by the number of times that rate occurred, yielding $P(\theta|a)$, the probability of movement direction given a discharge rate. In this analysis, the firing rate ranged from 1 to 282 spikes/s. For display, the data were partitioned equally into groups of low, medium and high discharge rate. The 3D directions were converted into polar coordinates and are represented here as azimuth (0° to the right, positive counterclockwise viewed from above) on the horizontal axis and elevation (0° up) on the forward axis. The PDFs are shown here as three contour plots, arranged with the lowest firing rate category on the bottom and the highest group of rates on top. The colors correspond to the probability of a particular direction of movement occurring. Numbered circles are the target locations with the same labels as those used in Figure 1.

data from motor cortical cells (20 ms bins) that were recorded as a monkey performed center-out and spiral drawing, this algorithm produced clusters that corresponded to movement directions. Clusters with similar directions were next to each other, so that the output layer organization reflected the target locations in the center-out task and formed a spiral in the drawing task. This representation of the parameter space is referred to as the 'topology-preserving' property of the SOFM. In short, the SOFM classifies firing patterns according to their similarities, resulting in clusters that can then be examined for their relationship to specific behavioral variables.

A three-layer ANN based loosely on physiological connectivity has been used to generate isometric forces from motor cortical spike trains [25]. The output consisted of six muscles acting as simple arm actuators in a 2D shoulder–elbow model. Connection weights of the four-element middle layer were adjusted so that spike trains recorded as a monkey performed an isometric task produced a force in the model that matched that of the monkey.

Another ANN has been used by Wessberg *et al.* [12•] to control a robot arm using neural signals from a monkey's cerebral cortex. In this work, 'hand position' was the control variable assumed to be represented in the cortical activity. The ANN had a single hidden layer consisting of 15–20 units and used gradient-descent back-propagation for offline training. The three Cartesian dimensions were considered to be independent, and each had a separate output layer. Once trained, the network was used to predict hand position from ensemble discharge. This is an example of a classifier scheme used, in this case, to map neural input patterns to hand position.

A recurrent network using nonlinear excitation functions has been designed to act as an ML estimator and has produced a similar performance [6]. Inputs were ordered by preferred direction, and all 64 elements in the input vector had the same tuning function, but the tuning directions were separated by 6°. The input projected to an output layer that was also one-dimensional, with elements ordered by preferred direction. Finally, the output layer was fully interconnected. Connection weights were fixed and set so that nearest neighbors shared excitation. The excitatory weights decreased as the difference in preferred directions increased so that more distant weights became inhibitory.

The algorithm began with a transient input to the output layer. Activity in the output layer was then iterated five or six times until activity across the output vector formed a stable, unimodal hill with a distinct peak near a particular output element. The preferred direction of this element was taken as the most likely movement direction. This network is not adaptive because the weights are not adjusted with a learning rule — instead, the output layer filters the noisy input pattern. The nonlinear activation function of the output units effectively removes Gaussian noise from the input, fitting a hill to the population of activity from the input. On the basis of simulations, this network's performance approached that of ML estimation.

Conclusions

Extraction algorithms for motor prostheses are designed to recognize motor parameters embedded in neural activity. Optimal algorithms will extract all the available information about a specified parameter in the recorded population activity. For engineering applications, it is essential that the parameters specified are well represented in the neural structure from which the activity is being recorded. For example, several of the algorithms discussed here are based on the position of the moving limb but, in primary motor cortex at least, this parameter is more poorly represented than velocity during movement [24–26]. The idea of several sources of variance, which is used in statistics, is relevant to this issue. A neuron's modulation can be related to many parameters, with some parameters accounting for more of the modulation (variability) than others. If parameters are correlated, then they share a source of variability. Parameters that account for a larger percentage of the spike train variance can be more easily extracted with population algorithms than those with weaker representations. With most algorithms, the different sources of variability need to be specified explicitly because some sort of optimal function is being modeled to the cell response.

The SOFM is partially immune to this problem as it clusters its output nodes only on the basis of the similarity of its input firing patterns (but these still must be labeled after the network is tuned). Also, the PVA — perhaps because it relies on the actual firing rates of neurons — is robust in this respect as it can handle simultaneously at least two parameters, direction and speed [29], even though speed is weakly encoded in the activity of single cells. Two factors — a uniform parameter distribution and unimodal tuning functions — are almost universal in determining the success of all these algorithms. It might seem that the pattern recognizers would be spared from these limitations. Even in these approaches, however, non-uniform clustering in the parameter space makes random noise more damaging. Multi-modal tuning functions make it less likely that a clear winner can be found in ML methods. Of course, all extraction methods are limited by the range and type of variation in the input spike trains.

Correlation within and between parameters is a factor that has not been fully exploited as yet in decoding algorithms. For example, it is unlikely, within a time series of firing rates from a single cell during a movement, that rates adjacent in time are uncorrelated. Instead, these rates vary smoothly. The time-dependent coefficients in the optimal linear filters use this in an indirect way. It is well known that many movement parameters, such as intrinsic and extrinsic parameters of the arm, co-vary [13,30,31]. Knowledge of these relationships can make the extraction algorithms more efficient, by limiting possible outcomes using the same principles as fuzzy logic controllers [32].

The present set of extraction methods falls into three overlapping categories. 'Pattern recognition' can be done either explicitly, or more indirectly with ANNs and ML methods. The performance of these techniques is constrained by their training sets and may be limited, both in terms of extrapolation beyond and interpolation within the training set when novel data are applied. The success of the 'linear filters' is due to the underlying linearity of the relationship between firing rate and movement direction. Again, these filters are limited by the conditions used to fit their coefficients and may suffer from the same training constraints as ANNs. An advantage of the optimal filters is that they can account for time lags between spike and movement data, if these lags are stationary. The 'population vector algorithm' has the advantage of being independent of its exact basis function; it has been shown to be robust across tasks and can encode several parameters simultaneously. The PVA can be modified so that, like the other methods, single-cell contributions are weighted to give an optimal prediction. However, this method is probably the most sensitive to non-uniformities in the preferred direction distribution.

As neuroprosthetics advance, other performance factors will become important. For instance, now that animals are learning to use devices that provide feedback of their performance, the ability of the subject to modify neuronal activity patterns to fit better the constraints of the extraction algorithm, and the development of adaptive algorithms that modify these constraints can enhance the target accuracy in a reaching task [33]. Our ability to study these learning dynamics is an exciting development both in the science and in the engineering of neural systems.

Acknowledgements

Our work is supported by the Neurosciences Research Foundation and an NIH contract (N01-NS-9-2321).

References and recommended reading

Papers of particular interest, published within the annual period of review, have been highlighted as:

- • of special interest
- •• of outstanding interest

1. Georgopoulos AP, Kalaska JF, Caminiti R, Massey JT: **On the relations between the direction of two-dimensional arm movements and cell discharge in primate motor cortex.** *J Neurosci* 1982, **2**:1527-1537.

2. Georgopoulos AP, Caminiti R, Kalaska JF, Massey JT: **Spatial coding of movement: a hypothesis concerning the coding of movement direction by motor cortical populations.** *Exp Brain Res* 1983, **Suppl 7**:327-336.

3. Schwartz AB, Kettner RE, Georgopoulos AP: **Primate motor cortex and free arm movements to visual targets in three-dimensional space. I. Relations between single cell discharge and direction of movement.** *J Neurosci* 1988, **8**:2913-2927.

4. Amirikian B, Georgopoulos AP: **Directional tuning profiles of motor cortical cells.** *Neurosci Res* 2000, **36**:73-79.

5. Georgopoulos AP, Kettner RE, Schwartz AB: **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.** *J Neurosci* 1988, **8**:2928-2937.

6. Pouget A, Zhang K, Deneve S, Latham PE: **Statistically efficient estimation using population coding.** *Neural Comput* 1998, **10**:373-401.

7. Salinas E, Abbott LF: **Vector reconstruction from firing rates.** *J Comp Neurosci* 1994, **1**:89-107.

8. Schwartz AB: **Motor cortical activity during drawing movements: population response during sinusoid tracing.** *J Neurophysiol* 1993, **70**:28-36.

9. Schwartz AB: **Direct cortical representation of drawing.** *Science* 1994, **265**:540-542.

10. Moran DW, Schwartz AB: **Motor cortical activity during drawing movements: population representation during spiral tracing.** *J Neurophysiol* 1999, **82**:2693-2704.

11. Schwartz AB, Moran DW: **Motor cortical activity during drawing movements: population representation during lemniscate tracing.** *J Neurophysiol* 1999, **82**:2705-2718.

12. • Wessberg J, Stambaugh CR, Kralik JD, Beck PD, Laubach M, Chapin JK, Kim J, Biggs SJ, Srinivasan MA, Nicolelis MA: **Real-time prediction of hand trajectory by ensembles of cortical neurons in primates.** *Nature* 2000, **408**:361-365.
This paper shows that, in single trials in real time, ensemble activity can encode arm trajectory. As the monkey subjects have no knowledge of their extracted signal, this is an example of open-loop control.

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

14. Rieke F, Warland D, de Ruyter van Steveninck R, Bialek W: *Spikes Exploring the Neural Code.* Cambridge, MA: MIT Press; 1997.

15. Brown EN, Frank LM, Tang D, Quirk MC, Wilson MA: **A statistical paradigm for neural spike train decoding applied to position prediction from ensemble firing patterns of rat hippocampal place cells.** *J Neurosci* 1998, **18**:7411-7425.

16. Paradiso MA: **A theory of the use of visual orientation information which exploits the columnar structure of striate cortex.** *Biol Cybern* 1988, **58**:35-49.

17. Seung HS, Sompolinsky H: **Simple models for reading neuronal population codes.** *Proc Natl Acad Sci USA* 1993, **90**:10749-10753.

18. Lee D, Port NL, Kruse W, Georgopoulos AP: **Variability and correlated noise in the discharge of neurons in motor and parietal areas of the primate cortex.** *J Neurosci* 1998, **18**:1161-1170.

19. Maynard EM, Hatsopoulos NG, Ojakangas CL, Acuna BD, Sanes JN, Normann RA, Donoghue JP: **Neural interactions improve cortical population coding of movement direction.** *J Neurosci* 1999, **19**:8083-8093.

20. Glaser EM, Ruchkin DS: *Principles of Neurobiological Signal Analysis.* London: Academic Press; 1976.

21. Gorsuch RL: *Factor Analysis*, edn 2. Hillsdale, New Jersey: Lawrence Erlbaum and Associates; 1983.

22. • Isaacs RE, Weber DJ, Schwartz AB: **Work toward real-time control of a cortical neural prosthesis.** *IEEE Trans Rehabil Eng* 2000, **8**:196-198.
The first demonstration of single-trial, open-loop arm trajectory extraction.

23. Kohonen T: **Cortical maps.** *Nature* 1990, **346**:24.

24. Lin S, Si J, Schwartz AB: **Self-organization of firing activities in monkey's motor cortex: trajectory computation from spike signals.** *Neural Comput* 1997, **9**:607-621.

25. Lukashin A, Amirikian BR, Georgopoulos AP: **A simulated actuator driven by motor cortical signals.** *Neuroreport* 1996, **7**:2597-2601.

26. Kettner RE, Schwartz AB, Georgopoulos AP: **Primate motor cortex and free arm movements to visual targets in three-dimensional space. III. Positional gradients and population coding of movement direction from various movement origins.** *J Neurosci* 1988, **8**:2938-2947.

27. Fu QG, Flament D, Coltz JD, Ebner TJ: **Temporal encoding of movement kinematics in the discharge of primate primary motor and premotor neurons.** *J Neurophysiol* 1995, **73**:836-854.

28. Ashe J, Georgopoulos AP: **Movement parameters and neural activity in motor cortex and area 5.** *Cereb Cortex* 1994, **6**:590-600.

29. Moran DW, Schwartz AB: **Motor cortical representation of speed and direction during reaching.** *J Neurophysiol* 1999, **82**:2676-2692.

30. Soechting JF, Flanders M: **Moving in three-dimensional space: frames of reference, vectors, and coordinate systems.** *Annu Rev Neurosci* 1992.

31. Helms Tillery SI, Ebner TJ, Soechting JF: **Task dependence of primate arm postures.** *Exp Brain Res* 1995, **104**:1-11.

32. Wang QJ, He J: **Fuzzy control of postural stability under large perturbations.** *International Federation of Automatic Control* 1999, 313-334.

33. Taylor DM, Schwartz AB: **Using virtual reality to test the feasibility of controlling an upper limb FES system directly from multiunit activity in the motor cortex.** In *Proceedings of the 6th Annual IFESS Conference: 2001 June 16–20; Cleveland.* 2001:132-134.