ELSEVIER

Journal of Neuroscience Methods 113 (2002) 187–197

JOURNAL OF NEUROSCIENCE METHODS

www.elsevier.com/locate/jneumeth

Method for assessing directional characteristics of non-uniformly sampled neural activity

Paul L. Gribble *, Stephen H. Scott

CIHR Group in Sensory-Motor Systems, Department of Anatomy and Cell Biology, Centre for Neuroscience Studies, Queen's University, Kingston, Ont., Canada

Received 27 July 2001; received in revised form 5 October 2001; accepted 5 October 2001

Abstract

We present a new analytical method for characterizing the directional tuning of neural data. The method is based on computing parameters associated with the geometric properties of solids, and provides an estimate of preferred direction in the context of non-uniform sampling of directions. Unlike optimization methods based on fitting tuning functions, the plate method is computationally fast, and does not require the assumption of an underlying tuning function (e.g. cosine or von Mises functions). In addition to estimating the preferred direction of a dataset, the plate method provides other parameters to fully characterize the directional properties of neural data. The method is presented in the context of a two-dimensional coordinate system but may in principle be extended to higher dimensional spaces as well.

Keywords: Directional tuning; Neural discharge; Tuning functions; Plate method; von Mises; Cosine; Vector method

1. Introduction

Research in many areas of neuroscience has focused on characterizing the spatial or directional properties of neurons. For example, individual neurons in visual sensory areas are tuned to the orientation, or direction of motion, of a stimulus presented within a neuron's receptive field (Hubel and Wiesel, 1969). Similarly, neurons in motor cortical areas are tuned to the direction of limb motion during a reaching task (Georgeopoulos et al., 1982; Scott and Kalaska, 1997). A number of properties of the directional tuning of neurons may be defined, including the broadness of tuning, the magnitude of neural activity and the 'preferred direction' of a neuron (e.g. Amirikian and Georgeopoulos, 2000). A number of different computational methods have been used to characterize properties such as these, and they fall into two general categories: analytical methods, and optimization procedures.

In the vector method, an analytical technique, the preferred direction of a neuron is characterized by summing together individual vectors, whose directions represent the stimulus (or response) directions, and whose magnitudes represent the associated magnitudes of neural responses in each direction. The direction of the mean vector then represents the preferred direction of the neuron (Batschelet, 1981; Georgeopoulos et al., 1982; Scott and Kalaska, 1997). A major advantage of the vector method is its simplicity and computational speed. However, the vector method depends on the assumption that directions are sampled equally around the unit circle. Deviations from this result in biased estimates of preferred direction that are skewed away from directions that are undersampled and towards oversampled directions.

In optimization methods, the parameters of an underlying tuning function are varied in order to fit the neural data. Typically this is achieved using an iterative multidimensional optimization numerical technique. This approach has been used in the past to characterize directional data using cosine, gaussian and von Mises tuning functions (e.g. Schwartz, 1992; Ajemian et al., 2000; Amirikian and Georgeopoulos, 2000; Todorov,

* Corresponding author. Present address: Department of Psychology, The University of Western Ontario, London, Ont., Canada N6A 5C2. Tel.: +1-519-661-2111x86185; fax: +1-519-661-3961.
E-mail address: pgribble@uwo.ca (P.L. Gribble).

0165-0270/02/$ - see front matter
PII: S0165-0270(01)00492-7

2000). Optimization methods based on fitting tuning functions can handle non-uniform sampling of directions, but tend to be computationally slow due to their iterative nature.

Here we present a new analytical method, the 'plate method', that addresses these problems. The method makes use of parameters associated with the geometry of objects to define a number of useful parameters for characterizing the directional tuning of neurons. We demonstrate that the plate method provides an estimate of preferred direction in the context of non-uniform directional sampling that is as good as is provided by methods that use tuning functions. The plate method does not require the assumption of an underlying tuning function, but is based solely on the data itself. We compare the performance of the plate method to other widespread methods including the vector method, cosine and von Mises fitting. Since the plate method is analytical, the computational time required is an order of magnitude shorter than that required to run comparable fitting procedures using cosine or von Mises tuning functions.

2. Methods

The plate method involves expressing neural activity as a solid area with uniform density in the coordinate space in which it is sampled. For example, in an experiment in which neural activity is recorded from motor cortex cells during horizontal plane limb movements in different directions (e.g. Scott et al., 2001), the directional characteristics of a cell's activity may be expressed as an area in Cartesian space. As an example, Fig. 1(A) shows hypothetical cell activity as a function of spatial direction for eight different movement directions (vertical lines). The method involves linear interpolation to compute cell activity for directions that are not sampled (solid lines). As a result, the data are divided into eight sections. Fig. 1(B) shows the result of plotting the magnitude of discharge as a function of the direction in Cartesian space. Note that because the neural activity was linearly interpolated relative to movement direction (Fig. 1A), the outer perimeters of the interpolated sections appear curved in Fig. 1(B).

The centroid of the area may be computed and used to define a cell's preferred direction. The centroid $(\hat{x}, \hat{y})$ of the area as a whole may be expressed as the sum of the centroids of each element, weighted by the areas of each element:

$$\hat{x} = \frac{\sum_{i=1}^{n} \hat{x}_i A_i}{\sum_{i=1}^{n} A_i} \quad (1)$$

$$\hat{y} = \frac{\sum_{i=1}^{n} \hat{y}_i A_i}{\sum_{i=1}^{n} A_i} \quad (2)$$

where $(\hat{x}_i, \hat{y}_i)$ and A_i are the centroids and areas, respectively, of each of the n individual elements that comprise the total area ($n = 8$ in the case of the data shown in Fig. 1). The centroid of the individual elements may be expressed as:

$$\hat{x}_i = \frac{\int_{\theta_i}^{\theta_{i+1}} r_i^3/3 \cos\theta \, \mathrm{d}\theta}{A_i} \quad (3)$$

$$\hat{y}_i = \frac{\int_{\theta_i}^{\theta_{i+1}} r_i^3/3 \sin\theta \, \mathrm{d}\theta}{A_i} \quad (4)$$

where A_i, the area of an element, is:

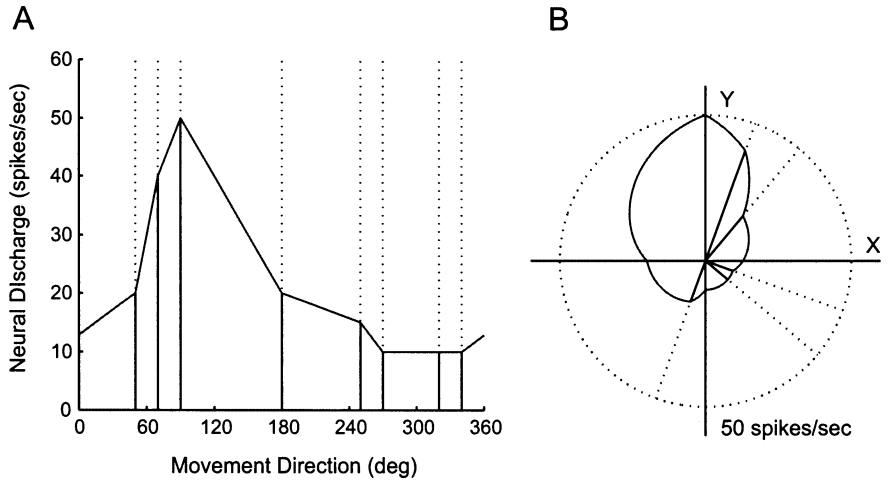

Fig. 1. (A) Neural activity is shown for a hypothetical cell during planar arm movements in different directions, sampled non-uniformly around the unit circle. The data are linearly interpolated over directions. (B) Representation of neural data as an area in Cartesian space.

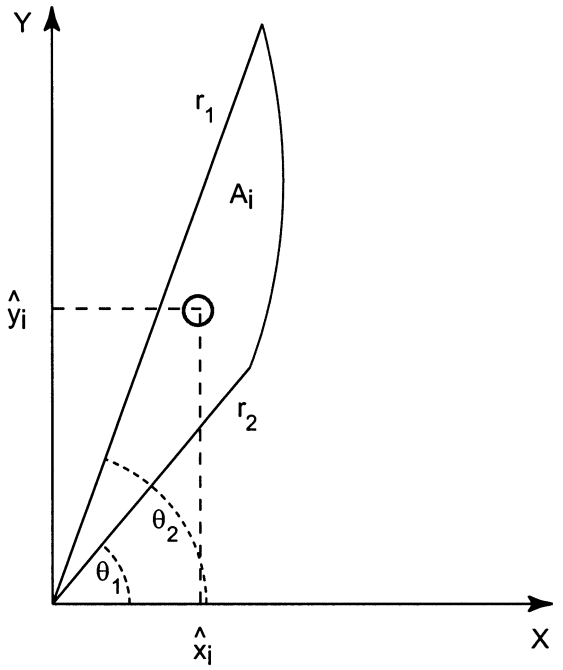

Fig. 2. Geometric properties of a single element of the total area (see Fig. 1). Based on the observed discharge levels r_1 and r_2 for directions θ_1 and θ_2, respectively, the centroid of the element $(\hat{x}_i, \hat{y}_i)$ may be computed (see text). The centroid of the element is indicated in the figure by o.

$$A_i = \int_{\theta_i}^{\theta_{i+1}} \frac{r_i^2}{2}\,\mathrm{d}\theta \tag{5}$$

and r_i, the function relating cell activity to direction θ is:

$$r_i = \frac{r_{i+1} - r_i}{\theta_{i+1} - \theta_i}(\theta - \theta_i) \tag{6}$$

Fig. 2 shows a schematic illustrating the parameters associated with one element of the area shown in Fig. 1(B).

Once the centroids of individual elements are computed, the preferred direction of the neural data as a whole, PD, may be calculated as the direction associated with the centroid $(\hat{x}, \hat{y})$ (indicated in Fig. 3 by X):

$$\mathrm{PD} = \tan^{-1}\frac{\hat{y}}{\hat{x}} \tag{7}$$

A measure of the average activity, M of the cell over all directions may be estimated by considering the total area A:

$$M = \sqrt{\frac{A}{\pi}} \tag{8}$$

where A is the sum of the areas of the individual elements:

$$A = \sum_{i=1}^{n} A_i \tag{9}$$

In addition, moments of inertia may be computed, which represent the overall level of activity about a given set of axes within the coordinate frame. For example I_x, I_y and I_{xy}, the moments of inertia about the x-axis and y-axis and the product of inertias, respectively, may be expressed as:

$$I_x = \sum_{i=1}^{n} I_{x_i} \tag{10}$$

$$I_y = \sum_{i=1}^{n} I_{y_i} \tag{11}$$

$$I_{xy} = \sum_{i=1}^{n} I_{xy_i} \tag{12}$$

where the individual moments for each of the n elements of the total area are given by:

$$I_{x_i} = \int_{\theta_i}^{\theta_{i+1}} \frac{2}{9} r_i^4 \sin^2\theta\,\mathrm{d}\theta \tag{13}$$

$$I_{y_i} = \int_{\theta_i}^{\theta_{i+1}} \frac{2}{9} r_i^4 \cos^2\theta\,\mathrm{d}\theta \tag{14}$$

$$I_{xy_i} = \int_{\theta_i}^{\theta_{i+1}} \frac{2}{9} r_i^4 \sin 2\theta\,\mathrm{d}\theta \tag{15}$$

Note that computational formulae corresponding to the solutions to the integrals presented here are given in Appendix A.

As a measure of the sharpness of tuning, one may compute the following ratio:

$$I_\mathrm{r} = \frac{I_1}{I_2} \tag{16}$$

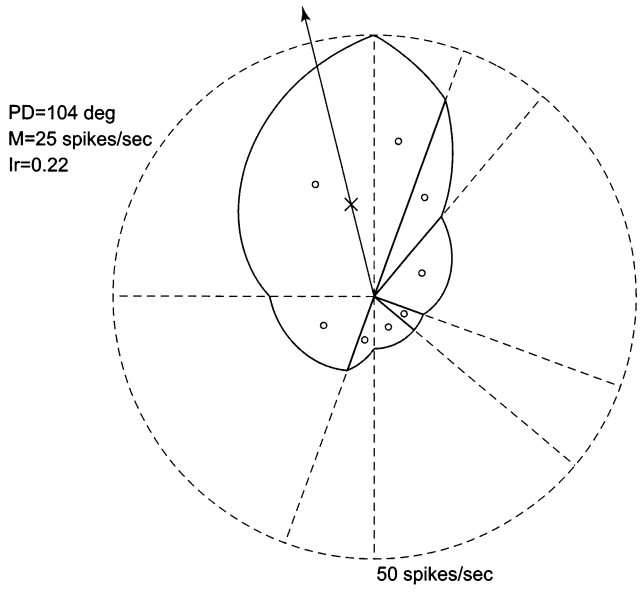

Fig. 3. Plate method parameters computed for the data shown in Fig. 1. The centroids of each element of the total area are indicated by o; the centroid of the total area is indicated by X; the preferred direction is indicated by an arrow.

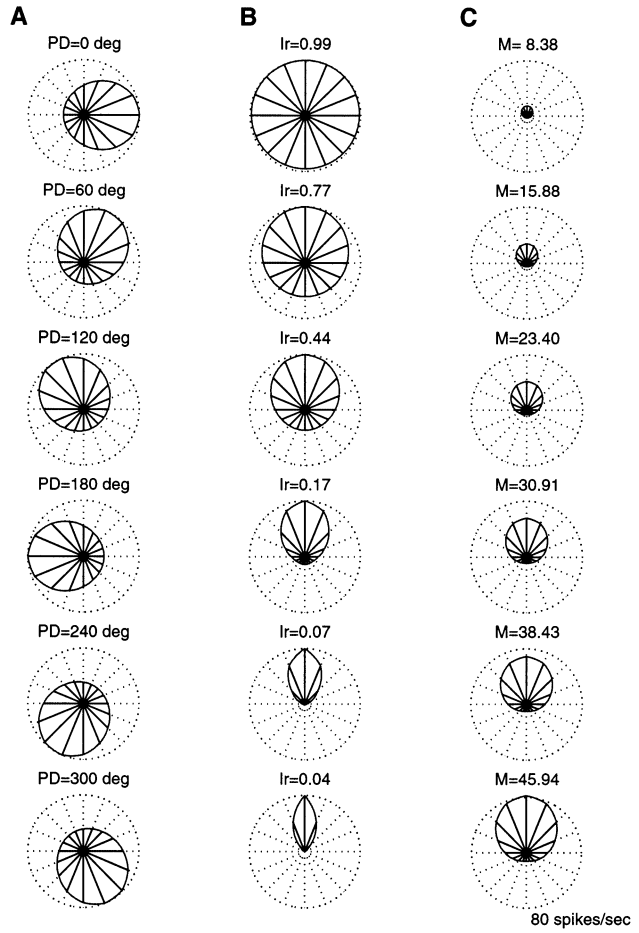

Fig. 4. Relationship of three plate method parameters to hypothetical neural data. (A) Each row represents a different preferred direction PD. (B) Shows the relationship between sharpness of tuning and the parameter I_r. (C) Shows how the parameter M can be used to characterize the mean level of activity over directions. In all cases shown in the figure, von Mises functions were used to generate the hypothetical data.

where I_1 and I_2 are the moments of inertia about the preferred direction PD and the axis perpendicular to the PD, respectively:

$$I_1 = \frac{I_x + I_y}{2} + \frac{I_x - I_y}{2} \cos 2\mathrm{PD} - I_{xy} \sin 2\mathrm{PD} \quad (17)$$

$$I_2 = \frac{I_x + I_y}{2} - \frac{I_x - I_y}{2} \cos 2\mathrm{PD} + I_{xy} \sin 2\mathrm{PD} \quad (18)$$

It should be noted that these axes intersect at the origin of the coordinate system, not at the area centroid $(\hat{x}_i, \hat{y}_i)$ described below.

Fig. 3 gives values for M, PD and I_r corresponding to the hypothetical neural data shown in Fig. 1. The centroids $(\hat{x}_i, \hat{y}_i)$ of each element are shown using circles, and the centroid of the total area $(\hat{x}_i, \hat{y}_i)$ is indicated by an X. The preferred direction of the neuron is shown using an arrow.

In order to visualize the relationship between parameters computed using the plate method, and directional data, hypothetical neural data were constructed using von Mises tuning functions. Fig. 4 illustrates the relationship between directionality (Fig. 4A), sharpness of tuning (Fig. 4B) and magnitude of response (Fig. 4C) to the plate method parameters PD, I_r and M. The PD parameter accurately represents the direction in which the magnitude of the response is greatest. As the sharpness of tuning decreases, the value of the I_r parameter increases, to a maximum value of 1.00 for a response not modulated by direction. It may be noted that a cosine function is associated with a value of 0.44 for the I_r parameter. The magnitude parameter M increases as the overall magnitude of the hypothetical neural response increases, and may be considered to represent an average response magnitude over all directions.

3. Comparison with other methods

Here we compare the performance of the plate method to three other widespread methods of characterizing the directional tuning of neurons—the vector method, cosine and von Mises fitting (see Appendix B). These methods are compared in terms of their accuracy in estimating preferred direction, the stability of these estimates in the face of random noise, and computational time requirements. A least-squares optimization routine provided in MATLAB software (Mathworks) was used to implement von Mises and cosine fitting. For the analyses presented here, the parameters of the optimization algorithm were set to the default values provided by MATLAB: the limit on the number of iterations was set to 1000, the termination tolerance on the tuning function value was 1e−6 and the termination tolerance on the tuning function parameter values was 1e−6.

Fig. 5 displays the results of applying the four methods to hypothetical neural data, constructed using a von Mises tuning function. Fig. 5(A) illustrates the directional tuning of this hypothetical neuron; in this case uniform directional sampling was assumed. The preferred direction (PD) of the neuron was computed using each method, and repeated 1000 times in the presence of random gaussian noise (mean = 0, S.D. = 25% of maximum discharge). The mean PDs are indicated in Fig. 5(A) using arrows for the plate, vector, cosine and von Mises methods, respectively, from left to right. In Fig. 5(B), the means and standard deviations (S.D.) are shown. For all four methods, the mean PD was not significantly different than the true PD (90°) (t-test, $P > 0.05$). The variability of the PD estimates for the plate method was slightly lower than for the other three methods.

The procedure was repeated using the same hypothetical neural data, but with a non-uniform sampling

of directions (Fig. 5C). In this case the mean PDs are all skewed towards quadrant I, which is over-sampled, and away from quadrant II, which is under-sampled. The vector method provides a highly biased estimate of the PD; the mean PD is significantly different than the true PD (t-test, $P < 0.01$). The other three methods provide a far better (though still slightly biased) estimate of the true PD. The mean PD provided by the plate method is not significantly different than the PDs computed using cosine and von Mises fitting (t-test, $P > 0.01$).

In the non-uniform sampling shown in Fig. 5(C), quadrants I and IV are oversampled relative to quadrants II and III, which are undersampled. We thus assessed the sensitivity of estimates of PD using the plate method to the preferred direction. A von Mises tuning function with a PD ranging from 0 to 360° was sampled using the non-uniform sampling shown in Fig. 5(C), and the plate method was used to estimate the preferred direction. In addition the procedure was repeated using tuning functions of different widths to assess the sensitivity of PD estimates using the plate method to the breadth of tuning. Fig. 6(A) displays the results. The error in estimating the true preferred direction is plotted against the true PD. Dashed lines indicate zero error. With increasing sharpness of tuning, error in predicting PD increases to a maximum of about $\pm 10°$ for the sharpest tuning function. Not

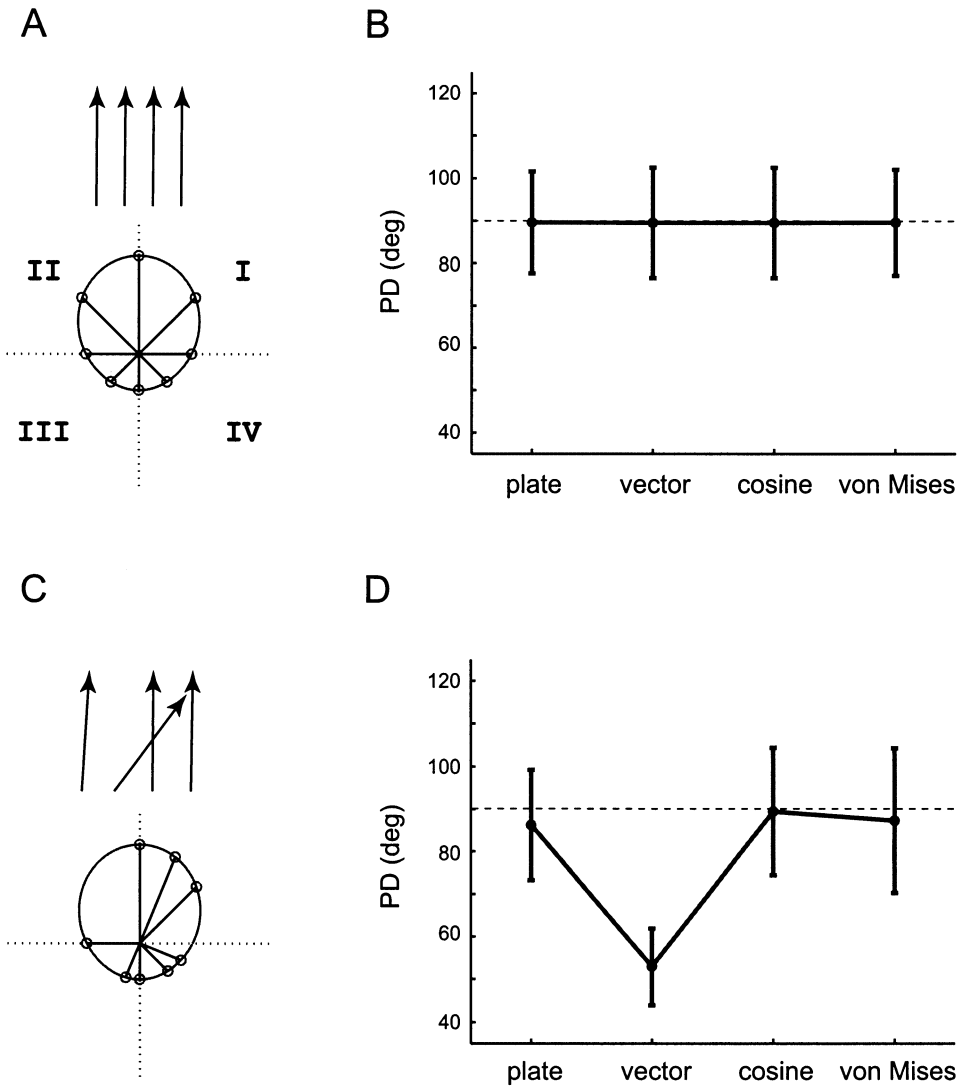

Fig. 5. Comparison of four different methods for assessing the preferred direction of hypothetical neural data. (A and B) The preferred direction of a hypothetical neuron sampled uniformly around the unit circle (A) was assessed using the plate method, the vector method, cosine and von Mises fits (see text). The procedures were repeated 1000 times in the presence of gaussian noise (mean = 0, S.D. = 25% of maximum discharge). Vertical arrows in A indicate the mean preferred direction computed by each method, respectively. B shows the mean and S.D. of preferred directions computed over 1000 trials for each method. Horizontal dashed line at 90° indicates the true preferred direction. (C and D) The procedure was repeated for a hypothetical neuron sampled non-uniformly around the unit circle.

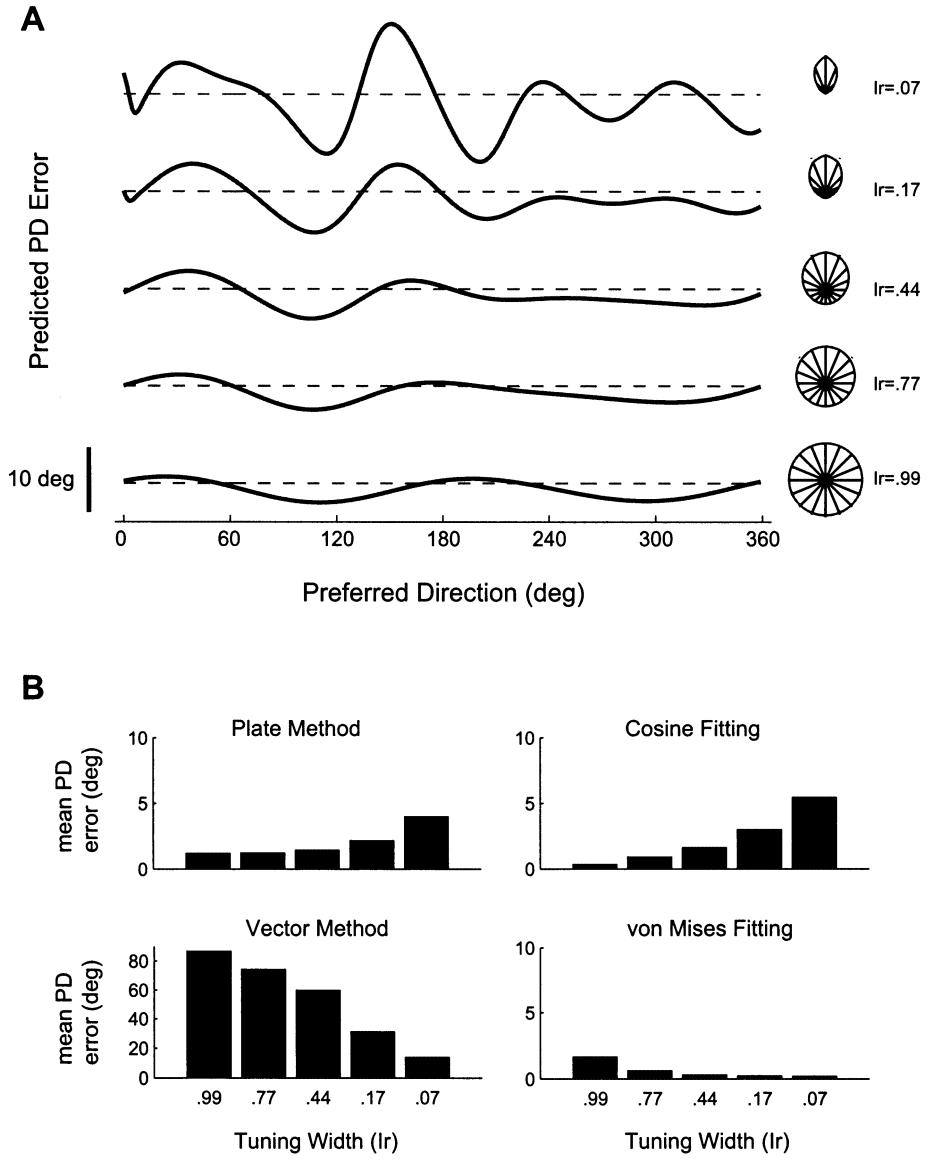

Fig. 6. Predicted error in preferred direction, PD for the plate method as a function of PD direction. The plate method was used to compute PD for von Mises tuning functions sampled with non-uniform directions (as in Fig. 5C). The true PD of the data was varied between 0 and 360°. The procedure was repeated using tuning functions of different widths. (A) For the plate method, the error in predicting PD is plotted against true PD. Dashed lines indicate zero error. (B) Mean absolute error in predicting PD over different directions is plotted as a function of tuning width, for the plate method, vector method, cosine and von Mises fitting.

surprisingly, maximum error in estimating PD occurs between 90 and 210°-regions corresponding to under-sampled directions. This procedure was also carried out for cosine, von Mises fitting and vector methods. In Fig. 6(B), the mean absolute error in predicting PD over different directions is plotted against the tuning widths tested, for the four methods tested. PD errors for cosine tuning are comparable to those for the plate method—errors range between 2 and 4.5°, and tend to increase as tuning width decreases. Errors for von Mises fitting are smaller, in the range of 0–2°, and there is tendency for errors to increase with broader tuning widths. Finally errors for the vector method are extremely high, not surprisingly (note the different vertical scale in Fig. 6B).

The tests for both uniform and non-uniform directional sampling were repeated to assess the sensitivity of the PD estimates to the amplitude of noise. Fig. 7 shows the results for five levels of random gaussian noise (mean = 0, S.D. from 0 to 50% of the maximum discharge). The preferred direction was 90° and the tuning width was $I_r = 0.44$ (e.g. Fig. 4). In Fig. 7(A), the mean PD error is plotted against the S.D. of the random gaussian noise that was added to the hypothetical neural data (Fig. 5), for each of the four methods, for a uniform sampling of directions. In all cases the

mean PD error increases approximately linearly with the amplitude of the noise. The slopes for the four methods were not significantly different ($P > 0.01$). Fig. 7(B) shows the results for a non-uniform sampling of directions. The mean PD errors for the vector method are significantly higher than for the other three methods (t-test, $P < 0.01$) except for the largest two noise levels, for which they are about the same. The slopes of the relationship between PD errors and noise level for the cosine and von Mises methods were not significantly different ($P > 0.01$) but the slope for the plate method was slightly lower ($P < 0.01$). For both uniform and in particular non-uniform directional sampling, the mean errors in predicting PD using the plate method were slightly lower than for the other three methods.

The computational time required to run each of the four methods was assessed. The tests were run using a dual-processor Intel Pentium-III 600MHz CPU running

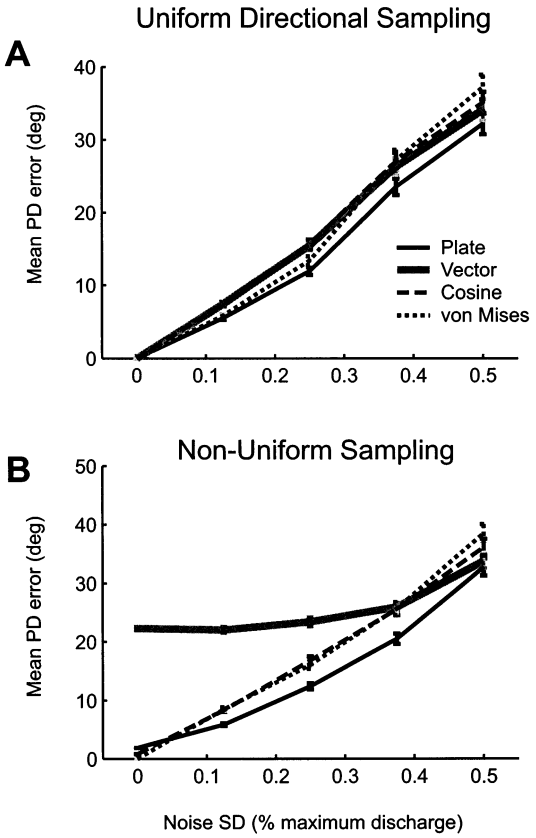

Fig. 7. Sensitivity of four methods of computing PD to level of random noise. Plate method, vector method, cosine and von-Mises fitting were repeated 1000 times in the presence of random (gaussian) noise (see Fig. 5A). The procedures were then repeated for five different noise levels ranging from 0 to 50% of the maximum amplitude of neural discharge. Mean errors in computed PD ($\pm$ one standard error (S.E.)) are plotted as a function of noise level for a uniform directional sampling (see Fig. 5A) and non-uniform sampling (see Fig. 5C).

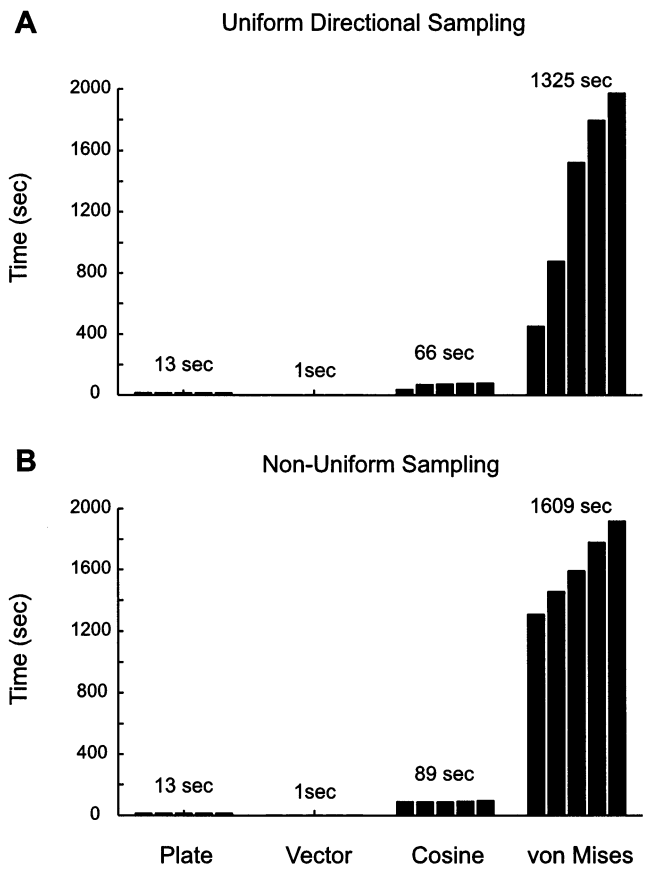

Fig. 8. Computational time measures. The average time (in seconds) taken to compute the PD of 1000 hypothetical neural signals (see Fig. 5) in the presence of random (gaussian) noise is plotted for plate and vector methods, and cosine and von Mises fits. The procedure was repeated for five different noise levels (see Fig. 7) and for uniform (A) and non-uniform (B) directional sampling. Text above each set of bars indicates the total time taken to compute 1000 PDs, averaged over five noise levels.

RED HAT LINUX v6.2 (kernel v2.2.19-6.2.1smp), using MATLAB v6.0.0.88 (R12). The time required to run each method 1000 times in the presence of gaussian noise was computed for the five levels of noise examined above (see Fig. 7). These times are shown in Fig. 8. For the von Mises fitting, and to a lesser extent cosine fitting, there was a dependence of the computational time on the level of noise—more time was required to fit noisier data. The average times across the five noise levels are shown above each set of bars. Computational times were similar for a uniform directional sampling (Fig. 8A) and a non-uniform sampling (Fig. 8B). The average time for 1000 runs was 13 s for the plate method, 1 s for the vector method, 66 s for the cosine fit (89 s for the non-uniform sampling) and 1325 s for the von Mises fit (1609 s for the non-uniform sampling).

4. Discussion

We have presented a new analytical method for characterizing the directional tuning of neural data. The plate method does not assume an underlying tuning function, it is computationally fast and it reduces bias in estimates of preferred direction in the context of non-uniform directional sampling.

While much research has focused on manipulating stimulus or response directions using uniform directional spacing, there are a number of reasons why directions may be sampled non-uniformly. For example, uniform sampling in one coordinate system may not correspond to uniform sampling in some other coordinate space of interest. For example in a recent study of motor cortex neurons during a reaching task, the investigators were interested in characterizing the relationship between cell discharge and the direction of motion in a shoulder–elbow joint torque space. Thus, it was desirable to specify movement targets uniformly sampled in joint torque space—resulting in non-uniformly spaced directions in Cartesian space (Scott et al., 2001).

The vector method is clearly inadequate for estimating preferred directions using non-uniform directional sampling. While methods based on fitting tuning functions do not require uniform sampling of directions, they are relatively slow because they typically use iterative numerical procedures to converge on a solution. This can have serious consequences on the time required to conduct sensitivity analyses or monte-carlo style 'bootstrap' tests, which often require thousands of runs on data for each neuron in a population (e.g. Scott and Kalaska, 1997). For example, to run a bootstrap test for directionality on a population of 500 neurons using 4000 iterations would require 894 h (37.3 days) using von Mises fitting (see Fig. 8). The plate method would require 7.2 h.

A related pitfall of optimization techniques centers around the problem of convergence. In the absence of a perfect (i.e. zero error) solution, iterative numerical optimization routines are generally halted once the error gradient reaches some minimum value, or once a pre-specified maximum number of iterations has been reached. In the former case the resulting set of parameters may correspond to the optimal values, but they may also represent a local (non-optimal) minimum. In many cases it is not possible to distinguish between these possibilities, and one must re-run the optimization routine with different initial parameter estimates to assess the stability of the solution. In the latter case, the algorithm may not have converged on an optimal solution once the maximum number of iterations has been reached, and so error may be introduced into the parameters of the fit (e.g. the estimate of preferred direction). This problem may be partially ameliorated, but not avoided in all cases by simply raising the maximum number of iterations. However, an appropriate stopping value is often not known a priori.

Another potential pitfall of techniques based on fitting tuning functions has to do with the tendency, in some cases, to overestimate the peak discharge of cells in cases where directions near the preferred direction are undersampled. In these cases, a fitting algorithm that attempts to maximize the fit between the data and a tuning function will be constrained to examining portions of the tuning function that are not near the preferred direction. As a result a fitting algorithm may be prone to overfitting the 'slopes' of the tuning function by modulating the sharpness of tuning. This may result in overestimates of the true peak discharge. This problem has been illustrated in Fig. 9. Data are sampled (shown using *o*) along equal directional intervals, but are undersampled near the preferred direction (90°). The data were generated by adding gaussian noise to an underlying von Mises tuning function (shown using dashed line) with a PD of 90°. A subsequent von Mises fit to the noisy data results in the tuning function shown using the solid line. The estimated peak discharge at the PD (150 spikes per s) is a gross overestimate of the true peak (40 spikes per s). In order to rule out this concern in the context of a particular dataset, a more systematic assessment of the conditions under which this problem may occur would be required.

Methods that do not rely on fitting an underlying tuning function, such as the vector and plate methods, provide a conservative estimate of peak discharge that is simply the maximum observed value. This avoids the

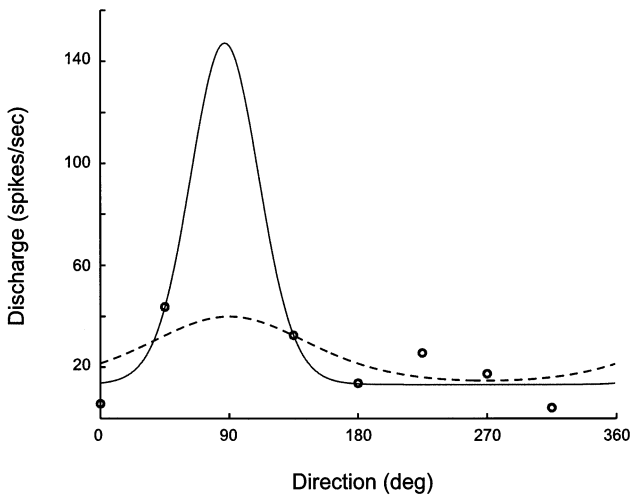

Fig. 9. Overestimating peak discharge using tuning functions for non-uniformly sampled data. A von Mises tuning function (solid line) was fit to hypothetical neural data that was undersampled around the true preferred direction (90°). The data (indicated using '*o*') were constructed based on a von Mises function with a preferred direction of 90° (dashed line). Random gaussian noise (mean = 0, S.D. = 15) was added to the signal.

problem of overestimating the peak discharge. In addition the cosine fitting procedure avoids this problem because the shape of the (cosine) tuning function is not varied. However, because of this constraint the cosine fitting procedure is unable to accurately characterize the directional tuning properties of cells that are significantly more or less sharply tuned than a cosine function. Recent data suggest that these cases represent a significant proportion of neurons in motor cortex. Amirikian and Georgeopoulos (2000) examined 73 primate motor cortex cells during a reaching task and found that more than half of the neurons that were well characterized using a von Mises tuning function had considerably narrower tuning than a cosine function.

In addition to providing a measure of the preferred direction PD of a dataset, the plate method provides a number of additional parameters that may be used to characterize the directional properties of neural data. The magnitude parameter M provides a measure related to the average strength of the signal over directions and the I_r parameter provides a measure of the broadness of tuning (it should be noted, however, that these parameters could be obtained by other means as well). Other parameters based on the concept of a moment of inertia about an axis may be of interest as well. For example, I_x and I_y provide a measure of the magnitude of a signal about the two coordinate axes, x and y, which may be useful for analyses involving differences in directionality, for example left versus right, or forward versus backward. In principle these inertial measures may be computed about any arbitrary axis of interest. The parameter I_{xy} (the product of inertias about the x and y axes) may also be of interest, which has a value of zero if the data are symmetric about one of the coordinate axes x and y.

In this paper, the plate method has been presented in the context of characterizing the directional properties of neurons. However, the method may be also be used to quantify directional properties of other signals as well, for example muscle activation patterns. A number of researchers have examined the directional tuning of muscle activation by recording electromyographic (EMG) signals from limb muscles during movements in different directions. Some have used cosine functions to characterize the directional tuning of EMG (Theeuwen et al., 1994) while others have utilized more complex tuning functions such as multiple-peaked cosine functions to characterize EMG signals with multiple directional peaks (Flanders and Soecthing, 1990; Herrmann and Flanders, 1988). The plate method may prove useful in the context of EMG recordings. In particular due to its analytical nature, and because the method is based only on observed data (and does not rely on assuming an underlying tuning function), the method may avoid many of the problems described above that are faced by optimization methods based on fitting tuning functions.

The plate method has been presented here in the context of a two-dimensional coordinate system, however, this is not a limitation of the method. In principle the plate method may be extended to the multidimensional case—for example, three-dimensional Cartesian space (Caminiti et al., 1990), or higher-dimensional spaces involving several limb joints. Another possible extension of the plate method is to include measures to characterize multi-lobed firing patterns, such as those described for limb muscle EMG patterns (Flanders and Soecthing, 1990). Additional measures which take into account boundary convexity/concavity may be considered (e.g. 'solid shape', Koenderink, 1990). This may be particularly useful in the context of higher dimensional applications such as multi-joint coordinate systems.

Acknowledgements

This research was supported by a CIHR Grant (MT-13462) and Scholarship to Stephen H. Scott and a CIHR Postdoctoral Fellowship to Paul L. Gribble.

Appendix A

In this section, we give the computational formulae associated with the equations presented in Section 2. A fully functioning MATLAB script has been made available and can be downloaded from the following website: http://spindle.ssc.uwo.ca/platemethod.html. The script takes as inputs a list of directions and discharge levels, and returns the plate method parameters PD, I_r, M, I_x, I_y, I_{xy}, $\hat{x}$ and $\hat{y}$.

Let c_1, c_2 and c_3 be:

$$c_1 = \frac{r_2 - r_1}{\theta_2 - \theta_1} \tag{19}$$

$$c_2 = \theta_1 c_1 \tag{20}$$

$$c_3 = r_1 - c_2 \tag{21}$$

where r_1, r_2, θ_1, θ_2 are as shown in Fig. 2. Then the area of an individual element A_i is:

$$A_i = \frac{1}{6}(c_1^2)((\theta_2^3) - (\theta_1^3)) + \frac{1}{2} c_3 c_1((\theta_2^2) - (\theta_1^2)) + \frac{1}{2}(c_3^2)(\theta_2 - \theta_1) \tag{22}$$

The centroid $(\hat{x}_i, \hat{y}_i)$ of an individual element is given by:

$$\begin{aligned}\hat{x}_i = {} & (1/3\sin\theta_2 - 1/3\sin\theta_1)c_3^3 \\ & + (\cos\theta_2 - \cos\theta_1 - \theta_1\sin\theta_1 + \theta_2\sin\theta_2)c_1c_3^2 \\ & + ((\theta_2^2-2)\sin\theta_2 + (-\theta_1^2+2)\sin\theta_1 + 2\theta_2\cos\theta_2 \\ & - 2\theta_1\cos\theta_1)c_1^2c_3 \\ & + ((\theta_2^2-2)\cos\theta_2 + (-\theta_1^2+2)\cos\theta_1 \\ & + (1/3\theta_2^3 - 2\theta_2)\sin\theta_2 \\ & + (-1/3\theta_1^3 + 2\theta_1)\sin\theta_1)c_1^3/(1/2\theta_2 - 1/2\theta_1)c_3^2 \\ & + 1/2c_3c_1(\theta_2^2-\theta_1^2) + (1/6\theta_2^3 - 1/6\theta_1^3)c_1^2 \end{aligned} \quad (23)$$

$$\begin{aligned}\hat{y}_i = {} & -1/3\cos\theta_2 + 1/3\cos\theta_1)c_3^3 + \sin\theta_2 - \sin\theta_1 \\ & + \theta_1\cos\theta_1 - \theta_2\cos\theta_2)c_1c_3^2 \\ & + ((-\theta_2^2+2)\cos\theta_2 + (\theta_1^2-2)\cos\theta_1 + 2\theta_2\sin\theta_2 \\ & - 2\theta_1\sin\theta_1)c_1^2c_3 \\ & + ((2\theta_2 - 1/3\theta_2^3)\cos\theta_2 + (1/3\theta_1^3 - 2\theta_1)\cos\theta_1 \\ & + (\theta_2^2-2)\sin\theta_2 \\ & + (-\theta_1^2 + 2(\sin\theta_1)c_1^3) \\ & /((1/2\theta_2 - 1/2\theta_1)c_3^2 + 1/2c_3c_1(\theta_2^2-\theta_1^2) \\ & + (1/6\theta_2^3 - 1/6\theta_1^3)c_1^2) \end{aligned} \quad (24)$$

The moments of inertia are given by:

$$\begin{aligned}I_x = {} & \left(\frac{1}{18}\sin 2\theta_1 - \frac{1}{9}\theta_1 + \frac{1}{9}\theta_2 - \frac{1}{18}\sin 2\theta_2\right)c_3^4 \\ & + \left(-\frac{2}{9}\theta_1^2 + \frac{1}{9}\cos 2\theta_1 - \frac{1}{9}\cos 2\theta_2 - \frac{2}{9}\theta_2\sin 2\theta_2\right. \\ & \left. + \frac{2}{9}\theta_1\sin 2\theta_1 + \frac{2}{9}\theta_2^2\right)c_1c_3^3 \\ & + \left(\left(-\frac{1}{6} + \frac{1}{3}\theta_1^2\right)\sin 2\theta_1 + \left(\frac{1}{6} - \frac{1}{3}\theta_2^2\right)\sin 2\theta_2\right. \\ & - \frac{1}{3}\theta_2\cos 2\theta_2 + \frac{1}{3}\theta_1\cos 2\theta_1 + \frac{2}{9}\theta_2^3 - \frac{2}{9}\theta_1^3\Big)c_1^2c_3^2 \\ & + \left(\left(\frac{1}{3}\theta_2 - \frac{2}{9}\theta_2^3\right)\sin 2\theta_2 + \left(\frac{2}{9}\theta_1^3 - \frac{1}{3}\theta_1\right)\sin 2\theta_1\right. \\ & + \left(\frac{1}{6} - \frac{1}{3}\theta_2^2\right)\cos 2\theta_2 + \left(-\frac{1}{6} + \frac{1}{3}\theta_1^2\right)\cos 2\theta_1 - \frac{1}{9}\theta_1^4 \\ & \left. + \frac{1}{9}\theta_2^4\right)c_1^3c_3 \\ & + \left(\left(\frac{1}{6}\theta_2 - \frac{1}{9}\theta_2^3\right)\cos 2\theta_2 + \left(\frac{1}{9}\theta_1^3 - \frac{1}{6}\theta_1\right)\cos 2\theta_1\right. \\ & + \left(\frac{1}{6}\theta_2^2 - \frac{1}{18}\theta_2^4 - \frac{1}{12}\right)\sin 2\theta_2 \\ & + \left(-\frac{1}{6}\theta_1^2 + \frac{1}{12} + \frac{1}{18}\theta_1^4\right)\sin 2\theta_1 + \frac{1}{45}\theta_2^5 \\ & \left. - \frac{1}{45}\theta_1^5\right)c_1^4 \end{aligned} \quad (25)$$

$$\begin{aligned}I_y = {} & \left(-\frac{1}{9}\theta_1 + \frac{1}{18}\sin 2\theta_2 + \frac{1}{9}\theta_2 - \frac{1}{18}\sin 2\theta_1\right)c_3^4 \\ & + \left(-\frac{1}{9}\cos 2\theta_1 - \frac{2}{9}\theta_1^2 + \frac{2}{9}\theta_2\sin 2\theta_2 + \frac{2}{9}\theta_2^2\right. \\ & \left. + \frac{1}{9}\cos 2\theta_2 - \frac{2}{9}\theta_1\sin 2\theta_1\right)c_1c_3^3 \\ & + \left(\left(-\frac{1}{6} + \frac{1}{3}\theta_2^2\right)\sin 2\theta_2 + \left(\frac{1}{6} - \frac{1}{3}\theta_1^2\right)\sin 2\theta_1\right. \\ & - \frac{1}{3}\theta_1\cos 2\theta_1 + \frac{2}{9}\theta_2^3 + \frac{1}{3}\theta_2\cos 2\theta_2 - \frac{2}{9}\theta_1^3\Big)c_1^2c_3^2 \\ & + \left(\left(-\frac{1}{6} + \frac{1}{3}\theta_2^2\right)\cos 2\theta_2 + \left(\frac{1}{6} - \frac{1}{3}\theta_1^2\right)\cos 2\theta_1\right. \\ & + \left(-\frac{1}{3}\theta_2 + \frac{2}{9}\theta_2^3\right)\sin 2\theta_2 + \left(\frac{1}{3}\theta_1 - \frac{2}{9}\theta_1^3\right)\sin 2\theta_1 \\ & \left. - \frac{1}{9}\theta_1^4 + \frac{1}{9}\theta_2^4\right)c_1^3c_3 \\ & + \left(\left(-\frac{1}{6}\theta_2 + \frac{1}{9}\theta_2^3\right)\cos 2\theta_2\right. \\ & + \left(-\frac{1}{9}\theta_1^3 + \frac{1}{6}\theta_1\right)\cos 2\theta_1 \\ & + \left(-\frac{1}{6}\theta_2^2 + \frac{1}{18}\theta_2^4 + \frac{1}{12}\right)\sin 2\theta_2 \\ & \left. + \left(\frac{1}{6}\theta_1^2 - \frac{1}{18}\theta_1^4 - \frac{1}{12}\right)\sin 2\theta_1 + \frac{1}{45}\theta_2^5 - \frac{1}{45}\theta_1^5\right)c_1^4 \end{aligned} \quad (26)$$

$$\begin{aligned}I_{xy} = {} & \left(-\frac{1}{18}\cos 2\theta_2 + \frac{1}{18}\cos 2\theta_1\right)c_3^4 \\ & + \left(\frac{2}{9}\theta_1\cos 2\theta_1 + \frac{1}{9}\sin 2\theta_2 - \frac{1}{9}\sin 2\theta_1\right. \\ & \left. - \frac{2}{9}\theta_2\cos 2\theta_2\right)c_1c_3^3 \\ & + \left(\left(\frac{1}{6} - \frac{1}{3}\theta_2^2\right)\cos 2\theta_2 + \left(\frac{1}{3}\theta_1^2 - \frac{1}{6}\right)\cos 2\theta_1\right. \\ & \left. - \frac{1}{3}\theta_1\sin 2\theta_1 + \frac{1}{3}\theta_2\sin 2\theta_2\right)c_1^2c_3^2 \\ & + \left(\left(\frac{1}{3}\theta_2 - \frac{2}{9}\theta_2^3\right)\cos 2\theta_2\right. \\ & + \left(-\frac{1}{3}\theta_1 + \frac{2}{9}\theta_1^3\right)\cos 2\theta_1 + \left(\frac{1}{3}\theta_2^2 - \frac{1}{6}\right)\sin 2\theta_2 \\ & \left. + \left(-\frac{1}{3}\theta_1^2 + \frac{1}{6}\right)\sin 2\theta_1\right)c_1^3c_3 \\ & + \left(\left(-\frac{1}{18}\theta_2^4 - \frac{1}{12} + \frac{1}{6}\theta_2^2\right)\cos 2\theta_2\right. \\ & \left. + \left(-\frac{1}{6}\theta_1^2 + \frac{1}{12} + \frac{1}{18}\theta_1^4\right)\cos 2\theta_1\right. \end{aligned}$$

$$+\left(-\frac{1}{6}\theta_2+\frac{1}{9}\theta_2^3\right)\sin 2\theta_2$$
$$+\left(\frac{1}{6}\theta_1-\frac{1}{9}\theta_1^3\right)\sin 2\theta_1\Big)c_1^4 \quad (27)$$

Appendix B.

Vector method

For each of the n sampled directions θ_i and associated neural discharge rates r_i, a vector V_i is constructed with direction θ_i and length r_i. These vectors are summed and normalized to compute a mean vector **V**:

$$\mathbf{V}=\frac{\Sigma\, V_i}{\Sigma\,\|V_i\|} \quad (28)$$

The direction of the mean vector **V** then defines the preferred direction of the cell. The length of the mean vector may be used as an index of the sharpness of tuning—the sharper the tuning, the longer the mean vector (for further details see Batschelet, 1981).

Optimization methods

The relation between stimulus (or response) direction, θ, and cell discharge rate, r, can be characterized by a tuning function, $r(\theta, x)$, where x is a set of parameters of the tuning function. Iterative optimization routines (such as those provided in MATLAB (Mathworks) software) may be used to find the optimal set of parameters x to fit the tuning function to a set of observed data—e.g. a set of observed discharge levels R_{obs} over n directions θ_{obs}. Typically the optimal parameters x are found by minimizing a cost function Q:

$$\min_x Q(\theta, x)=\frac{1}{2}\sum_{i=1}^{n}[R_{\text{obs}_i}-r(\theta_{\text{obs}_i}, x)]^2 \quad (29)$$

Cosine fitting

The tuning function may be specified using a cosine relation:

$$r(\theta)=x_3+x_2\cos(\theta-x_1) \quad (30)$$

In this paper the parameters x_{1-3} are estimated using a multidimensional non-linear optimization routine provided in MATLAB (Mathworks, Inc). The preferred direction of the cell is characterized by the parameter x_1. The parameter x_2 is related to the magnitude of discharge modulation, and x_3 relates to the minimum discharge.

von Mises fitting

The tuning function $r(\theta)$ may also be characterized by a von Mises function (Batschelet, 1981):

$$r(\theta)=x_4+x_3e^{x_2\cos(\theta-x_1)} \quad (31)$$

Again, the parameters x_{1-4} are estimated using an optimization routine, and the preferred direction of the cell is specified by x_1. Parameters x_2, x_3 and x_4 are related to the sharpness of tuning, the modulation of discharge and the minimum level of discharge, respectively.

References

Ajemian R, Bullock D, Grossberg S. Kinematic coordinates in which motor cortical cells encode movement direction. J Neurophysiol 2000;84:2191–203.

Amirikian B, Georgeopoulos AP. Directional tuning profiles of motor cortical cells. Neurosci Res 2000;36:73–9.

Batschelet E. Circular Statistics in Biology. London: Academic Press, 1981.

Caminiti R, Johnson PB, Urbano A. Making arm movements within different parts of space: dynamic aspects in the primate motor cortex. J Neurosci 1990;10:2039–58.

Flanders M, Soecthing JF. Arm muscle activation for static forces in three-dimensional space. J Neurophysiol 1990;64:1818–37.

Georgeopoulos 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;2:1527–37.

Herrmann U, Flanders M. Directional tuning of single motor units. J Neurosci 1988;18:8402–16.

Hubel DH, Wiesel TN. Anatomical demonstration of columns in the monkey striate cortex. Nature 1969;221:747–50.

Koenderink JJ. Solid Shape. Cambridge, MA: MIT Press, 1990.

Schwartz AB. Motor cortical activity during drawing movements: single-unit activity during sinusoid tracing. J Neurophysiol 1992;68:528–41.

Scott SH, Kalaska JF. Reaching movements with similar hand paths but different arm orientations. I. activity of individual cells in motor cortex. J Neurophysiol 1997;77(2):826–52.

Scott SH, Gribble PL, Graham KM, Cabel DW. Dissociation between hand motion and population vectors from neural activity in motor cortex, Nature 2001;413:161–5.

Theeuwen M, Gielen CC, Miller LE, Doorenbosch C. The relation between the direction dependence of electromyographic amplitude and motor unit recruitment thresholds during isometric contractions. Exp Brain Res 1994;98:488–500.

Todorov E. Direct cortical control of muscle activation in voluntary arm movements: a model. Nature Neurosci 2000;3:391–8.