Decoding M1 Neurons During Multiple Finger Movements

S. Ben Hamed,[1] M. H. Schieber,[1,2] and A. Pouget[1]

[1]*Department of Brain and Cognitive Science and the Center for Visual Science, University of Rochester; and* [2]*Departments of Neurology and Neurobiology, University of Rochester, School of Medicine and Dentistry, Rochester, New York*

Submitted 21 July 2006; accepted in final form 5 April 2007

Hamed S Ben, Schieber MH, Pouget A. Decoding M1 neurons during multiple finger movements. *J Neurophysiol* 98: 327–333, 2007. First published April 11, 2007; doi:10.1152/jn.00760.2006. We tested several techniques for decoding the activity of primary motor cortex (M1) neurons during movements of single fingers or pairs of fingers. We report that single finger movements can be decoded with >99% accuracy using as few as 30 neurons randomly selected from populations of task-related neurons recorded from the M1 hand representation. This number was reduced to 20 neurons or less when the neurons were not picked randomly but selected on the basis of their information content. We extended techniques for decoding single finger movements to the problem of decoding the simultaneous movement of two fingers. Movements of pairs of fingers were decoded with 90.9% accuracy from 100 neurons. The techniques we used to obtain these results can be applied, not only to movements of single fingers and pairs of fingers as reported here, but also to movements of arbitrary combinations of fingers. The remarkably small number of neurons needed to decode a relatively large repertoire of movements involving either one or two effectors is encouraging for the development of neural prosthetics that will control hand movements.

INTRODUCTION

Decoding population neuronal activities has become an important topic in neuroscience. From a theoretical perspective, the ability to read out neural patterns of activity can help us understand the structure of the neuronal code, how information is distributed over several thousand neurons, and how it flows from one area to the next (Dayan and Abbott 2001; Pouget et al. 2003). From a more clinical perspective, efficient decoding of neuronal population activity will be crucial for development of neuroprosthetics that are controlled on-line by neural activities.

Several decoders have been used in previous studies, such as the population vector, the optimal linear estimator, the maximum likelihood estimator, or the Bayesian estimator (Georgopoulos et al. 1986; Pouget et al. 2003; Salinas and Abbott 1994; Seung and Sompolinsky 1993). All of these methods use a population pattern of activity as input and return an estimate (or a probability distribution in the case of the Bayesian approach) of the encoded variable. In all cases, the encoded variable is assumed to take only one value at any given time. This is indeed a reasonable assumption in most cases. For instance, in decoding a reaching movement, the hand can be moving in only one direction at any given time. Likewise, when decoding the position of a rat in a maze, the rat can be in only one place at any time.

The situation becomes more complex, however, in decoding the concurrent motion of multiple effectors. Finger movements provide an important example because one finger might move in isolation or multiple fingers might move together. One solution to the problem of decoding these movements might be to develop a separate estimator for each finger, but this approach rests on the assumption that each finger is controlled independently of the others. In most natural behaviors, however, multiple fingers move together (for review, see Schieber and Santello 2004), suggesting that the pattern of neural activity needed to move the index finger, for example, may differ depending on whether the index moves in combination with the middle finger, in combination with the ring finger, or in isolation. It is therefore unclear whether training multiple decoders in parallel—one per finger—can lead to a high level of classification performance using a reasonably small number of M1 neurons.

We therefore examined several different methods of decoding finger movements. Surprisingly, we found that 100 neurons are sufficient to decode movements of pairs of fingers with an accuracy of 90.9%. Moreover, the same estimator can be used to decode single finger movements with near-perfect (>99%) accuracy using as few as 30 neurons.

METHODS

This study describes analysis of data collected from two monkeys that have been the subject of previous reports (Georgopoulos et al. 1999; Schieber 1991; Schieber and Hibbard 1993). Methods used for behavioral training and data collection have been described in these reports and are summarized here as needed. All care and use of these monkeys complied with the U.S.P.H.S. Policy on Humane Care and Use of Laboratory Animals and was approved by the institutional animal care and use committee.

Behavioral task

Each monkey was trained to perform visually cued individuated flexion and extension movements of the right hand fingers and/or wrist (Schieber 1991). As the monkey sat in a primate chair, the right elbow was restrained in a molded cast, and the right hand was placed in a pistol-grip manipulandum that separated each finger into a different slot. At the end of each slot, the fingertip lay between two microswitches. By flexing or extending the digit a few millimeters, the monkey closed the ventral or dorsal switch, respectively. The manipulandum, in turn, was mounted on an axis that permitted flexion and extension wrist movements. Each monkey viewed a display on which each digit (and the wrist) was represented by a row of five light-emitting diodes (LEDs). When the monkey flexed or extended a digit,

Address for reprint requests and other correspondence: A. Pouget, Brain and Cognitive Science Dept., Meliora Hall, Univ. of Rochester, Rochester, NY 14627 (E-mail: alex@bcs.rochester.edu).

The costs of publication of this article were defrayed in part by the payment of page charges. The article must therefore be hereby marked "*advertisement*" in accordance with 18 U.S.C. Section 1734 solely to indicate this fact.

closing a microswitch, the central yellow LED went out and a green LED to the left or right, respectively, came on, cueing the monkey as to which switch(es) had been closed. Red LEDs to the far left or right were illuminated one at a time, under microprocessor control, instructing the monkey to close that one switch (or move the wrist). If the monkey closed the instructed switch within the 700 ms allowed after illumination of the red instruction LED and held it closed for a 500-ms final hold period without closing any other switches, the monkey received a water reward. After each rewarded trial, the movement to be instructed for the next trial was rotated in a pseudorandom order. We abbreviate each instructed movement with the number of the instructed digit (1 = thumb through 5 = little finger, W = wrist), and the first letter of the instructed direction (f – flexion; e – extension); for example, "4f" indicates instructed flexion of the ring finger.

Monkeys K and S each performed all 12 possible movements involving flexion or extension of a single digit or of the wrist (1f, 1e, 2f, 2e, 3f, 3e, 4f, 4e, 5f, 5e, Wf, and We). In addition, each monkey also performed movements involving concurrent flexion or extension of two digits. Concurrent flexion of the thumb and index finger, for example, which we denote 1 + 2f, was instructed to the monkey by simultaneous illumination of the red LEDs used to instruct thumb flexion and index finger flexion. To obtain a reward, the monkey was required to close either the thumb or index finger flexion switch within 700 ms, to close the remaining switch within 100 ms of closing the first, and to hold them both closed for 500 ms without closing any other switches. Monkey K performed six multiple finger movements: 1 + 2f, 1 + 2e, 2 + 3f, 2 + 3e, 4 + 5f, 4 + 5e. Monkey S performed two: 1 + 2f, 1 + 2e. For each monkey, these multiple finger movements were presented intermingled with the 12 single finger movements in a pseudorandomized block design.

Data collection

After training, aseptic surgery under general anesthesia was used to perform a craniotomy over the left central sulcus at the level of the hand representation to implant a rectangular Lucite recording chamber over the craniotomy and to implant head-holding posts. Once the monkey had recovered from this procedure and had become accustomed to performing the finger movement task with the head held stationary, single neuron activity was recorded with conventional techniques as the monkey performed the finger movement task each day. Data were collected from 115 task-related neurons in the primary motor cortex (M1) of monkey K and from 61 M1 neurons in monkey S. In both cases, neurons were recorded from area 4 in the anterior bank and lip of the central sulcus.

Decoding methods

To include a neuron in this analysis, we required that the neuron had been recorded for a minimum of nine trials of each type of finger movement performed by the monkey. Data sufficient for these analyses were available from a total of 140 neurons: 100 from monkey K and 40 from monkey S. For a few neurons, as many as 19 trials of each movement were available. For each trial and for each neuron, spikes were counted in the 100-ms interval directly preceding the end of the movement (switch closure), a period when most M1 neurons reach peak firing rates. Using these data from M1 neurons in monkeys K and S, we examined the performance of the following five methods of decoding finger movements.

REGULAR POPULATION VECTOR. The population vector (PV) is typically used to estimate a periodic variable, like the direction of arm movement, from a set of neuronal responses (Georgopoulos et al. 1986). Georgopoulos and colleagues have shown how to extend this approach to discrete classes, and this approach has been used previously to analyze movements of single fingers (Georgopoulos et al. 1999). In this study, each finger movement is assigned a direction in three-dimensional (3D) space, with all 12 directions equally spaced in 3D. Then, on each trial k, the PV, $\mathbf{p}^{\mathrm{k}}$, is computed according to

$$\mathbf{p}^{\mathrm{k}} = \sum_{i=1}^{\mathrm{N}} \mathbf{c}_i r_i^k \tag{1}$$

where $\mathrm{r}_{\mathrm{i}}^{\mathrm{k}}$ is the activity of neuron i on trial k (i.e., the spike count in the 100-ms interval preceding the end of the movement), and $\mathbf{c}_{\mathrm{i}}$ is the preferred direction for neuron i.

For all cells, the preferred directions $\{\mathbf{c}_{\mathrm{i}}\}$ were calculated using the data from all available trials but two, and the PV was estimated using the data from the remaining two trials. This was repeated on all possible pairs out of the nine minimum trials retained for all cells, and the resulting PVs were averaged to produce a mean PV estimate.

In the Results, we report the performance of the PV in terms of the average angular difference between the vector assigned to the actual movement on trial k and the decoded population vector $\mathbf{p}^k$. We also report the percentage of correct responses taken as the percentage of conditions in which the PV was closest to the vector assigned to the actual movement.

OPTIMAL PV. The optimal population vector (OPV) is similar to the regular PV estimator in the sense that, on every trial k, we first estimate a vector $\mathbf{p}^{\mathrm{k}}$ by taking a linear combination of the neural activities

$$\mathbf{p}^k = \sum_{i=1}^{\mathrm{N}} \mathbf{w}_i r_i^k$$

The main difference from the PV is that for the OPV the weights, $\{\mathbf{w}_{\mathrm{i}}\}_{\mathrm{i}=1\ldots\mathrm{N}}$, are not set a priori to the preferred directions, $\{\mathbf{c}_{\mathrm{i}}\}_{\mathrm{i}=1\ldots\mathrm{N}}$. Instead, we seek the weights that minimize the cost function

$$\mathrm{E} = \sum_{k=1}^{\mathrm{K}} \|\mathbf{p}^{\mathrm{k}} - \mathbf{p}^{*\mathrm{k}}\|^2$$

where $\mathbf{p}^{*\mathrm{k}}$ is the vector corresponding to the actual movement on trial k. The optimal weights are given by

$$\mathbf{W}^{\mathrm{T}} = \Sigma^{-1}\mathbf{C}_{\mathrm{rp}*}$$

where $\mathbf{W}$ is the weight matrix with column $\{\mathbf{w}_{\mathrm{i}}\}_{\mathrm{i}=1\ldots\mathrm{N}}$, $\boldsymbol{\Sigma}$ is the covariance matrix of the neuronal responses, and $\mathbf{C}_{\mathbf{rp}*}$ is the covariance matrix between the response and the vector $\mathbf{p}*$ corresponding to the actual movements.

Here again, for all cells, the optimal weights $\{\mathbf{w}_{\mathrm{i}}\}_{\mathrm{i}=1\ldots\mathrm{N}}$ were calculated on all available trials but two, and the PV was estimated on the remaining two trials. This was repeated on all possible pairs out of the nine minimum trials retained for all cells, and the resulting PVs were averaged to produce a mean PV estimate.

LOGISTIC REGRESSION. Logistic regression (LR) is equivalent to training a nonlinear two layer neural network. The input layer consisted of N units corresponding to N M1 neurons. N was varied as described in the final section of METHODS. These N input units projected onto 12 output units, 1 output unit for each possible movement (Fig. 1*A*). The activity of output unit i on trial k, y_i^k, was determined according to

$$y_i^k = \frac{1}{1 + \exp\left(-\beta \sum_{j=1}^{N} w_{ij} r_j^k + b_i\right)}$$

where $\{w_{ij}\}_{i=1\ldots6,\, j=1\ldots\mathrm{N}}$ are the weights, b_{i} is the bias of output unit i, $\{r_j^k\}_{j=1\ldots\mathrm{N}}$ are the activities of the M1 neurons on trial k, and β is a constant (β = 0.0002). The weights were optimized through gradient

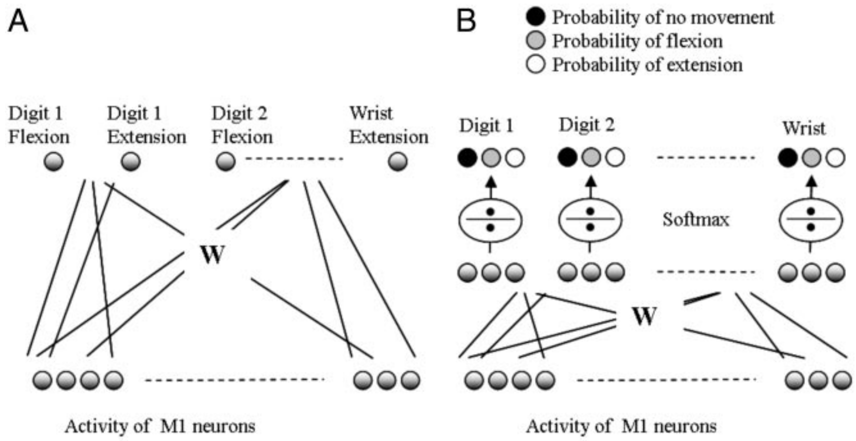

FIG. 1. Network representations of the logistic regression (LR) and softmax (SM) estimators. *A*: logistic regression. Input layer encodes the activity of *N* M1 neurons before a finger movement (*N* varies from 5 to 140, depending on simulation). Output layer contains 1 unit per movement for a total of 12 units (5 fingers plus wrist and 2 movements per finger or wrist). Output units are trained to encode probability of movement they encode given current input activities. Training involves optimizing weights *W* through gradient descent with cross-validation. *B*: SM. Input layer is as in *A*. Output layer is organized in 6 triplets of units (5 fingers plus wrist). Each triplets encodes probability distributions over 3 mutually exclusive states of corresponding finger (or wrist): no movement, flexion, and extension. SM normalization is used within each triplet to ensure that activity of the 3 units always sum up to 1, as should be the case for a probability distribution. Weights *W* are trained through gradient descent with cross-validation.

descent with cross-validation as described below. With this particular activation function, the value of each output unit is constrained to lie within the [0,1] interval. This enabled us to interpret the activity of output unit *i* as encoding the probability that movement *i* was produced on trial *k*.

This approach was applied to both single finger movements and movements of pairs of fingers, but with one major caveat: the number of fingers moving on a given trial was known for each trial. If only one finger moved, for that trial, we picked the one finger with the highest probability. If two fingers moved, we picked the two fingers with the highest probability.

SOFTMAX. The softmax (SM) estimator (Bishop 1995) was used to address the main limitation of the LR approach, the requirement that the number of fingers moving on each trial must be known a priori. For the SM network, the output layer was organized into six subgroups corresponding to the five fingers and the wrist. Each subgroup contained three units receiving inputs from all M1 neurons. These three output units were trained to encode the probabilities of no movement, of extension, and of flexion, respectively (Fig. 1*B*). (Again, weights were optimized through gradient descent with cross-validation, see METHODS). Because these three states of each finger are mutually exclusive (e.g., a given finger cannot flex and extend at the same time), the three probabilities were constrained to sum to unity through an SM normalization; thus the activity of unit *i* (with $i = 1.3$), in the *j*th subgroup ($j = 1.6$), for the *k*th trial is given by

$$o_{\mathrm{ij}}^{\mathrm{k}} = \frac{\exp\left(\sum_{i=1}^{N} w_{ij} l r_t^k\right)}{\sum_{j=1}^{3} \exp\left(\sum_{l=1}^{N} w_{ij} l r_l^k\right)}$$

Note that this equation ensures that the activity of the three units within a subgroup sums up to one, i.e., $\sum_{j=1}^{3} o_{ij}^{k}$ for any subgroup *i* and any example *k*.

SM-BEST VOTE-SELECTED SET (USING MUTUAL INFORMATION). Here we ranked the neurons from most informative to least informative. To do so, we computed the mutual information between single neurons and finger movements, using the standard definition (Dayan and Abbott 2001)

$$I(r_i;m)=H(r_i)-H(r_i|m)$$

where r_{i} and m are two random variables corresponding to the response of neuron *i* and the movement performed by the monkey. The probability distributions, $p(r_{\mathrm{i}})$ and $p(r_{\mathrm{i}}|m)$, which are needed to compute the entropies $H(r_{\mathrm{i}})$ and $H(r_{\mathrm{i}}|m)$, were estimated from the data by counting the frequency of each response using all the data available. This method is known to overestimate information, but we used *I* only to rank the neurons from most informative to least informative, enabling us to reduce the number of neurons needed for optimal decoding performance using the softmax estimator. This approach is labeled SM-best vote in Figs. 3 and 4.

Training and cross-validation

The parameters of all of the estimators described above were estimated with cross-validation. The dataset contained *N* neurons for which neuronal activity recorded in 9–19 trials was available (the number of neurons, *N*, was varied). For each of the 18 movements (12 single finger movements and 6 multiple finger movements), two trials of data from each neuron were set aside for testing the network. For training the network, single remaining trials of data from each neuron were selected randomly to generate 10,000 patterns of population activity (~56 for each of the 18 movements). In this way, we generated population patterns of activity for each movement even though the neurons actually were recorded one at a time on different days. For testing the network, we generated 200 new population activity patterns (~1 for each of the 18 movements) using the two trials of data from each neuron during each movement set aside for this purpose.

In the case of LR and SM, we used stochastic gradient descent to find the weights minimizing the squared error over all movements and all trials. While optimizing the weights on the training set, we also monitored classification performance on the testing set. Training was stopped when performance started decreasing on the testing set, a procedure known as early stopping. This procedure prevents overfitting on the training set and provides a better estimate of generalization performance (Morgan and Bourlard 1990).

Varying the number of neurons being decoded

For all five decoding methods, we systematically varied the number of neurons, *N*, being decoded to determine the minimum number of

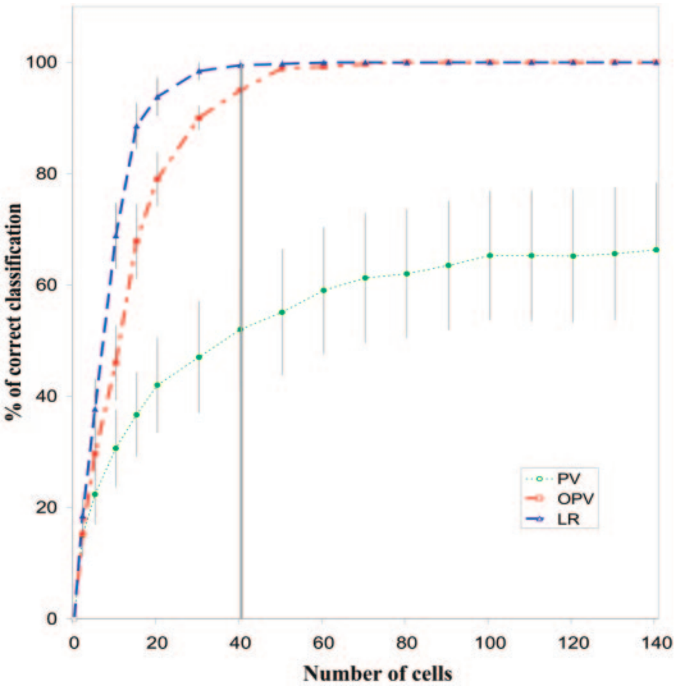

FIG. 2. Comparative performance of the regular population vector (PV), optimal population vector (OPV), and LR estimators on decoding single finger movements as a function of number of neurons being decoded. Gray bars represent SE. Long gray vertical line indicates number of cells for which LR reached a detection performance of 100%.

neurons needed for the best possible classification. N was set to 140, 100, 60, 40, 30, 20, 15, 10, and 5. For each value of N, the performance of the estimator was evaluated by randomly choosing N cells with replacement from the 140 cells available and optimizing and testing the estimator with those N cells. This process was repeated 10 times, and we report the average of these 10 repetitions.

RESULTS

Decoding of single finger movements with the regular PV and OPV estimators

We first decoded the 12 single finger and wrist movements from the discharge of M1 neurons using both the regular PV and OPV estimators (see METHODS). The average error, expressed in terms of the angle between the estimated direction and true direction, was 34.07° for the PV and 14.18° for the OPV, when using all 140 cells. These numbers are comparable with those reported by Georgopoulos et al. (1999), although this previous study did not use cross-validation.

Figure 2 shows the results expressed in terms of percentage of correct classification as a function of the number of cells being decoded. The OPV outperformed the PV for all numbers of cells. In fact, the PV never approached 99% correct, even with 140 cells, whereas OPV reached 99.2% accuracy using as few as 60 randomly selected cells.

Decoding of single finger movements with LR and SM

LR is the optimal classifier when the data follows a multivariate Gaussian distribution with a fixed covariance matrix (Bishop 1995). Because this assumption holds to a first approximation with neural activities, LR was a natural estimator to try with our data set. Moreover, LR has the advantage that it estimates the probability distribution over all possible movements given the vector of neural activity, instead of generating a single answer as do the PV and OPV. Figure 2 shows the performance of the LR estimator compared with PV and OPV. LR outperformed both population vector estimators, decoding single finger movements with 99.48% accuracy using as few as 40 randomly selected cells.

To address the main limitation of the LR estimator, namely the need to know how many fingers are moving on each trial, we developed another estimator, which we call an SM estimator. This estimator returns six probability distributions: one for each finger and one for the wrist. Each probability distribution is defined over three possible states: flexion, extension, and no movement. The SM estimator can be read out on every trial without prior knowledge of the number of fingers moving: we simply look for the most probable state of each finger. A typical answer might look like the following: no movement for finger 1, flexion for finger 2, no movement for finger 3, flexion for finger 4, no movement for finger 5, and no movement for the wrist. This approach can be used to estimate single finger movements, two finger movements, and more generally, any arbitrary combination of flexion and extension over the five fingers and wrist, regardless of the number of fingers moving.

Figure 3 shows the performance of the SM estimator (SM exact match) compared with LR on decoding single finger movements. The SM estimator performed just as well as LR, even though it used no prior knowledge of the number of fingers moving on each trial. Remarkably, the SM estimator not only assigns the highest probability to the proper move-

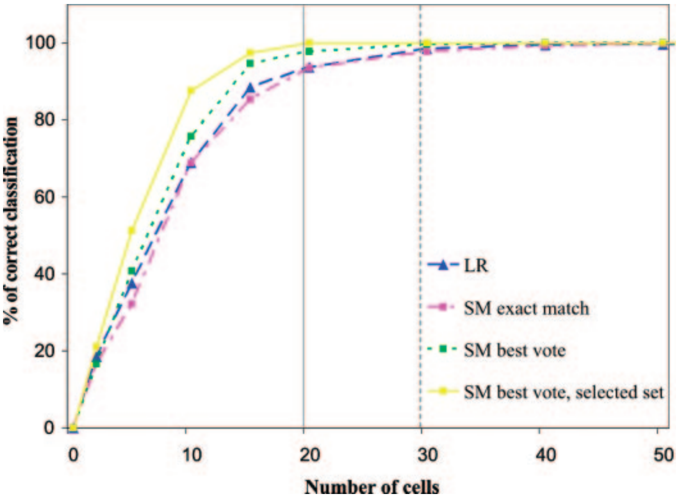

FIG. 3. Comparative performance of LR and SM estimators on decoding single finger movements as a function of dataset size. Two different classification criteria were applied to SM. In a most restrictive decoding scheme, SM estimator was considered to be correct only when it predicted correct movement for moving finger and no movement for all other fingers. In SM best vote, SM estimator was considered to be correct when it assigned highest probability to correct movement for moving finger, even if it predicted no movement for all other fingers. In other words, we assume that we know that only 1 finger moves on each trial, as was done for the LR estimator. Finally, for SM best vote, selected set, we applied the same criteria as for SM best vote, but we used the N cells with highest mutual information between responses and movement. Gray vertical dashed line indicates number of cells for which SM best vote reached a detection performance of 100%. Gray vertical continuous line indicates number of cells for which SM best vote, selected set reached a detection performance of 100%. SE is not plotted on this graph to avoid unnecessary clutter, but they are in the same range as in Fig. 2.

ment for the proper finger but also assigns the highest probability to no movement for all the other fingers.

For further comparison with the LR estimator, we also tested the SM estimator when assuming that we know that only one finger moved on each trial. This means that we read out the SM estimator in the same way as the LR estimator, searching only for the movement state (flexion or extension) with the highest probability, while ignoring all other answers. Therefore even if the SM estimator predicted that more than one finger moved, for this comparison, we nevertheless considered the prediction correct as long as the most likely movement was indeed the one that took place on that trial. As can be seen in Fig. 3 (SM best vote), this SM approach improves performance still further, decoding which finger moved at 99.6% accuracy using as few as 30 randomly selected neurons.

Finally, we asked whether the number of cells needed to decode at >99% accuracy could be reduced below 30 if we selected the neurons that provided the most information about which finger movement was performed. We therefore computed the mutual information between each neuron and the finger movements, and we trained the SM estimator using the *N* neurons with the highest information content. The number *N* was varied systematically to obtain the performance of the SM estimator as a function of the number of neurons. Using the most informative neurons, 20 selected cells were sufficient for 99.91% performance (Fig. 3, SM best vote-selected set). In fact, accuracy still was very high with as few as 15 cells (97.3% correct).

Of the 20 most informative cells, none discharged for one finger movement only. Instead, each of these neurons was intensely active during two or more finger movements and virtually inactive during others. Such distributed coding indeed is to be expected, given that a neuron that fired during only one finger movement would convey relatively little information (0.4 bits) compared, for instance, with a neuron that fired strongly for six movements and not at all for the other six (1 bit).

Our selection procedure was not necessarily optimal, however. We do not know that the 20 neurons with the highest mutual information constitute the best subset for classifying all movements. Therefore 20 neurons is an upper bound on the minimum number of neurons needed for near-perfect classification. However, better performance might be obtained with fewer neurons selected in a manner other than the one used here.

Decoding of multiple finger movements: training on single finger movements

Neither the PV nor the OPV is suitable for decoding simultaneous movements of multiple fingers because these estimators each return a single answer on any given trial. We therefore studied the LR and SM approaches for decoding trials in which two fingers moved at the same time. Here we used the population of 100 neurons from monkey K that had been recorded during at least nine trials of the 12 single finger and wrist movements plus the 6 movements in which the monkey moved two fingers at the same time.

First we explored whether an estimator trained on single finger movements can generalize to two-finger movements. Our results show that this is not the case. The LR estimator performed reasonably well on some two-finger combinations (e.g., 94% correct on 1 + 2e) but poorly on others (2% on 4 + 5e, which is below chance) for an average performance of 38% correct over all six two-finger movements. The SM estimator fared worse, failing to generalize from single finger movement to two-finger movements. The inability of the LR and SM methods to estimate two-finger movements after training on single finger movements indicates that *1*) the population activity patterns generated for different single-finger movements overlap (in that single cells discharge significantly during several single-finger movements (Poliakov and Schieber 1999; Schieber and Hibbard 1993) and *2*) the patterns of neural activity for two-finger movements are unlikely to be simple linear combinations of the patterns for single-finger movements.

To examine this idea further, we ran the following test. For all two-finger movements AB, where A and B refer to the single finger movements involved in this combination, we computed the angle between the population activity vector for AB and the vector obtained by taking the sum of the two population activity vectors for A and B. If the activity for the two finger movement is simply equal, or proportional to, the sum of the activities for the single finger movement, the angle should be close to zero. As can be seen on Fig. 4, the angle is close to 45° on average, indicating that some nonlinearity is involved.

Given the inability of the LR and SM approaches to generalize from single finger movements to two-finger movements, we tested both methods after training on all 18 single-finger and two-finger movements. Figure 5 shows the performance of several networks based on the activity of 100 neurons during the different movements. The LR network achieved nearly perfect performance on all combinations (overall average of 98.98%; Fig. 4, LR). However, this performance still relied on prior knowledge of whether the movement involved one finger or two fingers. The SM performance was nearly perfect on single-finger movements (99.25%) and several two-finger movements (Fig. 5, SM exact match), for an average performance of 90.9% over all 18 movements. Interestingly, and unlike what we observed for single-finger movements, performance of the SM estimator did not improve when tested with a priori knowledge of whether one or two fingers moved on each trial (Fig. 5, SM best vote).

DISCUSSION

Decoding of single finger movements and implications for neural prosthetics

Although many neural prosthetic approaches use recordings from large numbers of neurons, we were able to decode 12 single finger movements with >99% accuracy using as few as 30 neurons randomly selected from a population of 140 task-related neurons. If we assume that an individual can produce three finger movements per second, our performance corresponds to a bit rate of 10.75 bits/s [3 × log2(12)] for 40 neurons, which is significantly better than the current state of the art, which is ~6 bits/s for ~100 neurons (Santhanam et al. 2006; Taylor et al. 2003). This performance is also particularly remarkable given that it was obtained with relatively little data, because we had at most 19 trials per finger movement collected

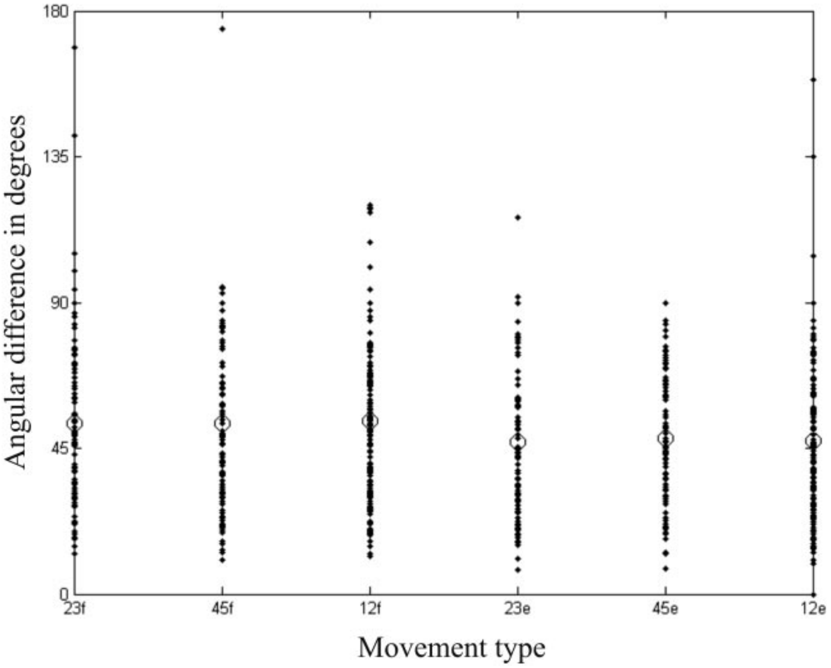

FIG. 4. Test on nonlinearity for 2 finger movements. For each 2-finger movement, we computed angle between population activity vector for this 2-finger movement and sum of population activity for the 2 single finger movements involved in this pair. Black dots correspond to single trials; circles correspond to average angles for each 2-finger movement. For all 2-finger movements, angle is close to 45° on average, indicating that patterns of neural activity for 2-finger movements are not simply sums of patterns for single-finger movements.

from any given neuron during recordings that lasted at most 30 min. Decoding performance using the LR and SM estimators should improve considerably with long-term recordings from implanted electrodes, when hundreds to thousands of trials per movement condition could be collected and used to train the decoding estimators.

These conclusions, however, should be tempered by the fact that we used independent neurons, whereas these other studies relied on neurons recorded simultaneously. As we discuss below, it is possible that our performance would decrease significantly with correlated neurons. Moreover, it is possible that the neurons we used were particularly well tuned compared with the neurons typically found with multiunit recordings using grids. Indeed, in single cell recordings, the experimentalist can select neurons with high firing rates, which are likely to be more informative. With grids recordings, this is not an option because the electrodes cannot be moved independently. Finally, the M1 neurons from which we recorded may have responded in part to proprioceptive feedback. In a paralyzed individual, such feedback would not be present, and the discharge of M1 neurons might differ significantly. This in turn could affect decoding performance. The precise impact of this factor cannot be assessed with our data set but deserves to be investigated in future studies.

Decoding of multiple finger movements

Decoding multiple finger movements failed when the LR and SM estimators were trained exclusively on single finger movements. This failure may arise in part from the fact that the neural and muscular activation required to move two fingers is not simply the linear combination of the activations required to move each finger in isolation (Schieber 1990). Because multiple adjacent fingers tend to move together, movement of a single finger in isolation requires contraction of both the muscles that move that finger and additional muscles that check unwanted motion in adjacent digits (Reilly and Schieber 2003; Schieber 1995; Schieber and Santello 2004). Hence generalization of a network trained on single finger movements to decode multiple finger movements is unlikely to succeed. When the LR and SM estimators were trained using data from

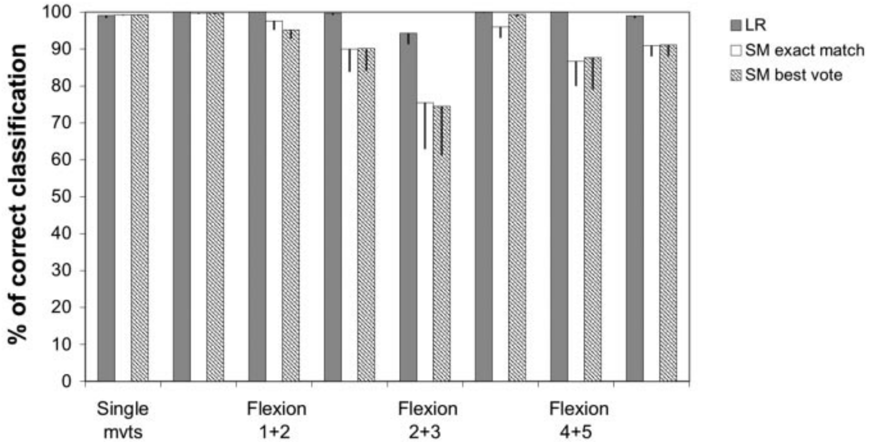

FIG. 5. Comparative performance of LR and SM estimators on decoding single and multiple finger movements. SM exact match and SM best vote are as in Fig. 3. Downward pointing bars represent SE.

all 12 single-finger and 6 two-finger movements, however, both methods provided accurate decoding using 100 neurons.

Nonetheless, with 140 cells, we were able to obtain >90% accuracy in decoding combination finger movements. We do not have enough data to know whether this performance would generalize to arbitrary combinations of movements. Assuming for purposes of discussion that a subject could produce three combinations of six possible movements per second (1 bit per digit or wrist), the bit rate potentially would reach 18 bits/s. In part, such a hypothetical bit rate, well above what has been reported by other groups, may reflect the simultaneous control of six effectors. These encouraging numbers are likely to be overestimates, however, given that all the caveats we have raised in decoding single movements would apply to combination movements as well, and in particular, the issue of correlations.

Impact of correlations

Because each of the present population of M1 neurons was recorded at a different time, the activities of different neurons in our population estimates effectively were uncorrelated. Our results might have been different had we used neural data obtained from simultaneous recordings of multiple single neurons, because the activity of simultaneously recorded cortical neurons often can be shown to be correlated (Bair et al. 2001; Maynard et al. 1999; Zohary et al. 1994). Such correlations can either increase or decrease the quality of decoding depending on their exact pattern (Abbott and Dayan 1999; Yoon and Sompolinsky 1999), however, making it difficult to predict whether performance would have been better or worse using simultaneously recorded neurons. Studies that have compared truly simultaneous recordings to "shuffled" recordings (equivalent to single unit recordings because shuffling data from different trials effectively removes the correlations between neurons) suggest that the impact of correlations is insignificant, or very small, for population of tens of neurons (Averbeck and Lee 2003; Panzeri et al. 2003). Moreover, Serruya et al. (2002) have found recently that 7–30 neurons recorded simultaneously are sufficient to control the location of a computer cursor on a screen. Although this task is quite different from the present finger movement task, their findings support the notion that <40 neurons can be sufficient for decoding, even when the neurons are recorded simultaneously. Our finding that as few as 30–40 M1 neurons were sufficient for decoding single finger movements therefore may have been similar even with simultaneously recorded neurons.

For larger neuronal sets, such as the 140 neurons we used for single finger movements, assessing the impact of correlated neuronal activity is more difficult because little is known about correlations among such large neuronal populations in vivo. Extrapolation from pairwise recordings suggests that, if our neurons were all equally positively correlated (the worse case scenario from the standpoint of decoding information), correlations become a major concern only for 1,000 neurons or more (Zohary et al. 1994). Therefore our results with 140 neurons recorded one at a time are likely to be reasonably close to what would have been obtained with 140 simultaneously recorded neurons. Nonetheless, there is little doubt that the impact of correlations on decoding multiple movements should be thoroughly studied in future experiments.

GRANTS

S. B. Hamed and A. Pouget were supported by National Institute of Mental Health Grant MH-57823, research grants from the Office of Naval Research, and the McDonnell-Pew, Sloan, and Schmitt foundations. M. H. Schieber was supported by National Institute of Neurological Disorders and Stroke Grant NS-27686.

REFERENCES

Abbott LF, Dayan P. The effect of correlated variability on the accuracy of a population code. *Neural Comput* 11: 91–101, 1999.

Averbeck BB, Lee D. Neural noise and movement-related codes in the macaque supplementary motor area. *J Neurosci* 23: 7630–7641, 2003.

Bair W, Zohary E, Newsome WT. Correlated firing in macaque visual area MT: time scales and relationship to behavior. *J Neurosci* 21: 1676–1697, 2001.

Bishop CM. *Neural Networks for Pattern Recognition.* New York: Oxford, 1995.

Dayan P, Abbott LF. *Theoretical Neuroscience.* Cambridge, MA: MIT Press, 2001.

Georgopoulos AP, Pellizzer G, Poliakov AV, Schieber MH. Neural coding of finger and wrist movements. *J Comput Neurosci* 6: 279–288, 1999.

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

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

Morgan N, Bourlard HA. Generalization and parameter estimation in feedforward nets: some experiments. In: *Advances in Neural Information Processing Systems 2*, edited by Touretzky D. San Mateo, CA: Morgan Kaufmann, 1990, p. 630–637.

Panzeri S, Petroni F, Petersen RS, Diamond ME. Decoding neuronal population activity in rat somatosensory cortex: role of columnar organization. *Cereb Cortex* 13: 45–52, 2003.

Poliakov AV, Schieber M. Limited functional grouping of neurons in the motor cortex hand area during individuated finger movements: a cluster analysis. *J Neurophysiol* 82: 3488–3505, 1999.

Pouget A, Dayan P, Zemel RS. Inference and computation with population codes. *Annu Rev Neurosci* 26: 381–410, 2003.

Reilly KT, Schieber MH. Incomplete functional subdivision of the human multitendoned finger muscle flexor digitorum profundus: an electromyographic study. *J Neurophysiol* 90: 2560–2570, 2003.

Salinas E, Abbott LF. Vector reconstruction from firing rates. *J Comput Neurosci* 1: 89–107, 1994.

Santhanam G, Ryu SI, Yu BM, Afshar A, Shenoy KV. A high-performance brain-computer interface. *Nature* 442: 195–198, 2006.

Schieber MH. How might the motor cortex individuate movements? *Trends Neurosci* 13: 440–445, 1990.

Schieber MH. Individuated finger movements of rhesus monkeys: a means of quantifying the independence of the digits. *J Neurophysiol* 65: 1381–1391, 1991.

Schieber MH. Muscular production of individuated finger movements: the roles of extrinsic finger muscles. *J Neurosci* 15: 284–297, 1995.

Schieber MH, Hibbard LS. How somatotopic is the motor cortex hand area? *Science* 261: 489–492, 1993.

Schieber MH, Santello M. Hand function: peripheral and central constraints on performance. *J Appl Physiol* 96: 2293–2300, 2004.

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

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

Taylor DM, Tillery SIH, Schwartz AB. Information conveyed through brain-control: cursor vs. robot. *IEEE Trans Neural Syst Rehabil Eng* 11: 195–199, 2003.

Yoon H, Sompolinsky H. The effect of correlations on the Fisher information of population codes. In: *Advances in Neural Information Processing Systems*, edited by Kearns M, Solla S, Cohn D. Cambridge, MA: MIT Press, 1999, p. 167–173.

Zohary E, Shadlen MN, Newsome WT. Correlated neuronal discharge rate and its implications for psychophysical performance. *Nature* 370: 140–143, 1994.