Behavioral/Systems/Cognitive

Decoding Trajectories from Posterior Parietal Cortex Ensembles

Grant H. Mulliken,[1] Sam Musallam,[2] and Richard A. Andersen[1,2]

[1]Computation and Neural Systems and [2]Division of Biology, California Institute of Technology, Pasadena, California 91125

High-level cognitive signals in the posterior parietal cortex (PPC) have previously been used to decode the intended endpoint of a reach, providing the first evidence that PPC can be used for direct control of a neural prosthesis (Musallam et al., 2004). Here we expand on this work by showing that PPC neural activity can be harnessed to estimate not only the endpoint but also to continuously control the trajectory of an end effector. Specifically, we trained two monkeys to use a joystick to guide a cursor on a computer screen to peripheral target locations while maintaining central ocular fixation. We found that we could accurately reconstruct the trajectory of the cursor using a relatively small ensemble of simultaneously recorded PPC neurons. Using a goal-based Kalman filter that incorporates target information into the state-space, we showed that the decoded estimate of cursor position could be significantly improved. Finally, we tested whether we could decode trajectories during closed-loop brain control sessions, in which the real-time position of the cursor was determined solely by a monkey's neural activity in PPC. The monkey learned to perform brain control trajectories at 80% success rate (for 8 targets) after just 4–5 sessions. This improvement in behavioral performance was accompanied by a corresponding enhancement in neural tuning properties (i.e., increased tuning depth and coverage of encoding parameter space) as well as an increase in off-line decoding performance of the PPC ensemble.

Key words: brain–machine interface; trajectory decoding; neural prosthetics; sensorimotor control; posterior parietal cortex; neurophysiology

Introduction

Scientific and clinical advances toward the development of a cortical neural prosthetic to assist paralyzed patients have targeted multiple brain areas and signal types (Kennedy et al., 2000; Wessberg et al., 2000; Serruya et al., 2002; Taylor et al., 2002; Carmena et al., 2003; Shenoy et al., 2003; Musallam et al., 2004; Patil et al., 2004; Wolpaw and McFarland, 2004; Hochberg et al., 2006; Santhanam et al., 2006). The first generation of motor prostheses focused primarily on extracting continuous movement information (trajectories) and emphasized the premotor (PMd) and primary motor cortices (M1) in monkeys (Serruya et al., 2002; Taylor et al., 2002; Carmena et al., 2003; Santhanam et al., 2006). Studies performed in the posterior parietal cortex (PPC) and in PMd have emphasized decoding goal information, such as the intended endpoint of a reach (Musallam et al., 2004; Santhanam et al., 2006). Musallam and colleagues also showed that cognitive variables (e.g., expected value of a reach) could be exploited to boost the amount of goal information decoded from PPC. However, less emphasis has been placed on decoding trajectories from PPC. Earlier off-line analyses showed that area 5 neurons are correlated with various motor parameters during reaching movements (Ashe and Georgopoulos, 1994; Averbeck et al., 2005). Recently, we reported that PPC encodes the instantaneous movement direction of a joystick-controlled cursor (i.e., with approximately zero lag time), suggesting that these dynamic tuning properties reflect the output of an internal forward model (Mulliken et al., 2008). Nonetheless, these PPC studies did not address whether trajectories could be reconstructed from ensemble neural activity. Carmena and colleagues decoded trajectories off-line and during closed-loop brain control trials from ensembles in multiple brain areas, including PPC, M1, PMd, and primary somatosensory cortex (S1) (Wessberg et al., 2000; Carmena et al., 2003). However, they reported relatively poor off-line reconstruction performance for their PPC sample compared with other brain areas they recorded from (e.g., M1). Therefore, it is unlikely that their closed-loop brain control performance relied strongly upon PPC activity relative to signals from other areas (e.g., M1).

Therefore, the extent to which PPC can be used to decode trajectories, off-line and during online brain control remains unclear. Here we show that trajectories can be reliably reconstructed off-line using relatively small numbers of simultaneously recorded PPC neurons. We also show that goal-based, state-space decoding methods as well as adjustable-depth multielectrode array (AMEA) recording techniques can be advantageous for a

Received April 4, 2008; revised Sept. 13, 2008; accepted Oct. 21, 2008.

This work was supported by the National Eye Institute, the James G. Boswell Foundation, the Defense Advanced Research Projects Agency, and a National Institutes of Health training grant fellowship to G.H.M. We thank J. Burdick, E. Hwang, and M. Hauschild for comments on this manuscript, K. Pejsa and N. Sammons for animal care, and V. Shcherbatyuk and T. Yao for technical and administrative assistance.

Correspondence should be addressed to Grant H. Mulliken at his present address: McGovern Institute for Brain Research, Massachusetts Institute of Technology, Building 46, Room 6165, 77 Massachusetts Avenue, Cambridge, MA 02139. E-mail: grantm@mit.edu.

S. Musallam's present address: Department of Electrical and Computer Engineering, McGill University, Montreal, Quebec, Canada H3A 2A7.

DOI:10.1523/JNEUROSCI.1463-08.2008

prosthesis targeting PPC. Finally, we show for the first time that PPC ensembles alone are sufficient for real-time continuous control of a cursor. Interestingly, we observed strong learning effects in PPC during brain control, which emerged in parallel with an increase in behavioral performance over a period of several days.

Materials and Methods

Animal preparation. Two male rhesus monkeys (*Macaca mulatta*; 6–9 kg) were used in this study. All experiments were performed in compliance with the guidelines of the Caltech Institutional Animal Care and Use Committee and the National Institutes of Health *Guide for the Care and Use of Laboratory Animals*.

Neurophysiological recording. We recorded multichannel neural activity from two monkeys in the medial bank of the intraparietal sulcus (IPS) and area 5. We implanted two 32-channel microwire arrays (64 electrodes) in one monkey (first array 6–8 mm deep in IPS, second array 1–2 mm deep in area 5) (Microprobe). Chronically implanted array placements were planned using magnetic resonance imaging (MRI) and performed using image-guided surgical techniques to accurately target the desired anatomical location and to minimize the extent of the craniotomy (Omnisight Image-guided Surgery System, Radionics). In a second monkey we performed acute chamber recordings using a 6-channel microdrive (NAN Electrode Drive, Plexon). The placement of the recording chamber was verified using MRI, with the chamber centered at 5 mm posterior, 6 mm lateral in stereotaxic coordinates. For the remainder of this report, we will refer to a chronic microwire array as a fixed-depth multielectrode array (FMEA) and the microdrive as an adjustable-depth multielectrode array (AMEA). Spike sorting was performed online using a Plexon multichannel data acquisition system and later confirmed with off-line analysis using the Plexon Offline Sorter. Using the AMEA technique, we were able to maintain single-unit isolations of several neurons for ~1–2 h. Note that we did not specify that isolated neurons be tuned to a particular parameter measured in our task, but instead only required that each neuron have a minimum baseline firing rate (>2 Hz). Once single-unit isolations were established manually, isolations were typically maintained using an autonomous electrode positioning algorithm (SpikeTrack), which independently adjusted the depth of each electrode to continuously optimize extracellular isolation quality (Cham et al., 2005; Nenadic and Burdick, 2005; Branchaud et al., 2006). It should be noted that all neural units reported using the AMEA technique were well isolated single units, while neural units recorded using the FMEA technique consisted of a combination of single-unit and multiunit activity.

Experimental design. Monkeys were trained to perform a 2D center-out joystick (J50 2-axis joystick, ETI Systems) reaction task, in which they were required to guide a cursor on a computer screen to illuminated peripheral targets. 2D-axis joystick movements were comprised primarily of wrist movements, with maximal radial excursion of 5 cm (i.e., to peripheral target location). After several weeks of training, the monkeys were highly skilled at the task and were performing regularly above 90% success rate. Experimental sessions consisted of a joystick training segment for both monkeys, which was followed by, for one monkey, a closed-loop brain control segment.

The 2D center-out joystick task is illustrated in Figure 1*A*. The monkeys sat 45 cm in front of an LCD monitor. Eye position was monitored with an infrared oculometer. Monkeys initiated a trial by moving a white cursor (0.9°) into a central green target circle (4.4°) and then fixated a concentric, central red circle (1.6°). After 350 ms, the target jumped to 1 of 8 (or 12 for some sessions) random peripheral locations (11–14.7°). The monkeys then guided the cursor smoothly into the peripheral target zone while maintaining central fixation. Once the cursor was held within 2.2° of the target center for 350 ms, the monkeys were rewarded with a drop of juice. If fixation was broken during the movement, the trial was aborted. Monkeys were required to fixate centrally during the entire trajectory in order to maintain a constant visual reference frame. Earlier studies have shown that parietal reach region (PRR) encodes visual targets for reaching in eye coordinates and area 5 in both eye and hand coordinates (Batista et al., 1999; Buneo et al., 2002). In addition, this control was important to rule out the possibility that we were decoding activity related to eye position or saccades. [Note that this control was not instituted in a previous PPC decoding study (Carmena et al., 2003).] The duration of a typical trajectory, from a monkey's reaction time (i.e., when the monkey initiated movement of the cursor) to 80 ms after the cursor first entered the target zone was 510 ± 152 ms and 393 ± 152 ms, for monkeys 1 and 2 respectively (mean ± SD).

During brain control sessions performed in one monkey, we disconnected the joystick from the cursor and attempted to decode the intended trajectory of the monkey using only his brain signals. The cursor was initially placed inside the central green target circle to start each brain control trial. The monkey was again required to look at a centrally located red circle to initiate the trial, but this time subsequent positions of the cursor were determined by a decoding model operating only on the neural signals. Cursor position was updated on the computer screen approximately every 100 ms until the cursor was held in the target circle (9°) for >100 ms. The trial was aborted if the monkey moved his eyes, or 10 s elapsed before successfully reaching the target.

Off-line algorithm construction. We sought to construct a decoding model that optimally estimated behavioral parameters from the firing rates of simultaneously recorded PPC neurons (e.g., minimized the mean squared reconstruction error, MSE). To further illustrate this situation, we have plotted simultaneously recorded spike trains from 5 neurons in the raster plots of Figure 1, *B* and *C*, which were collected during 10 trials made to both leftward and rightward targets. Neural activity is aligned to the reaction time and is plotted up to 80 ms after the cursor entered the target zone. Below each set of raster plots are the trial-averaged, standardized firing rate time courses associated with each of these two target directions. The instantaneous firing rate of a neuron was standardized by first subtracting the neuron's mean firing rate and then dividing by its SD. Using an ensemble of standardized firing rates along with concurrently recorded behavioral data from the joystick training segment, we constructed a mathematical decoding model to attempt to reconstruct the monkeys' trajectories off-line. To accomplish this, we tested two standard linear estimation algorithms: ridge regression and a modified Kalman filter (Kalman, 1960; Hoerl and Kennard, 1970). Fivefold cross-validation was used to assess performance and to perform model selection. Specifically, all trials from a joystick training segment were shuffled and then divided into five equal parts. Four of five parts were used to train the model and 1/5 parts was used to validate the model. Thus, the trained model was validated five times to obtain an average performance, with each of the five parts serving as the validation set one time.

Ridge regression. We first constructed a linear model of the instantaneous 2D cursor position (or velocity, acceleration, target position) at some time, $x(t)$, as a function of the standardized firing rates of N simultaneously recorded neural units. Firing rates were computed for non-overlapping 80 ms bins, and effectively represented the mean firing rate measured at the middle of each bin. Each sample of the behavioral state vector, $x(t)$, was modeled as a function of the vector of ensemble firing rates measured for three preceding time bins (i.e., lag times), centered at $\{r(t - 200 \text{ ms}), r(t - 120 \text{ ms}), r(t - 40 \text{ ms})\}$, which effectively represented the temporal evolution of the causal ensemble activity before each behavioral state measurement. To simplify our notation, we will refer to discretized time steps (k) for the remainder of this report, where $x(k)$ denotes the instantaneous behavioral measurement and $r(k)$ denotes the average binned firing rate of the ensemble 40 ms in the past, $r(k - 1)$ denotes the mean firing rate 120 ms in the past, etc. We tried a variety of bin sizes and number of lag time steps and found that these values provided the best performance over multiple sessions. During a given trial, spiking activity was sampled beginning from 240 ms before the monkeys' reaction time up to 80 ms after the cursor first entered the target zone. An estimate of the 2D cursor position (or velocity, etc.), $\hat{x}(k)$, was constructed as a linear combination of the ensemble of firing rates, r, sampled at multiple lag time steps according to

$$\hat{x}(k) = \beta_0 + \sum_{i=0}^{2} \sum_{j}^{N} \beta_{i,j} r(k - i, j) + \varepsilon(k - i), \qquad (1)$$

where ε represents the observational error. The MSE of the estimate, $\hat{x}(k)$, can generally be decomposed into two parts, a bias and a variance

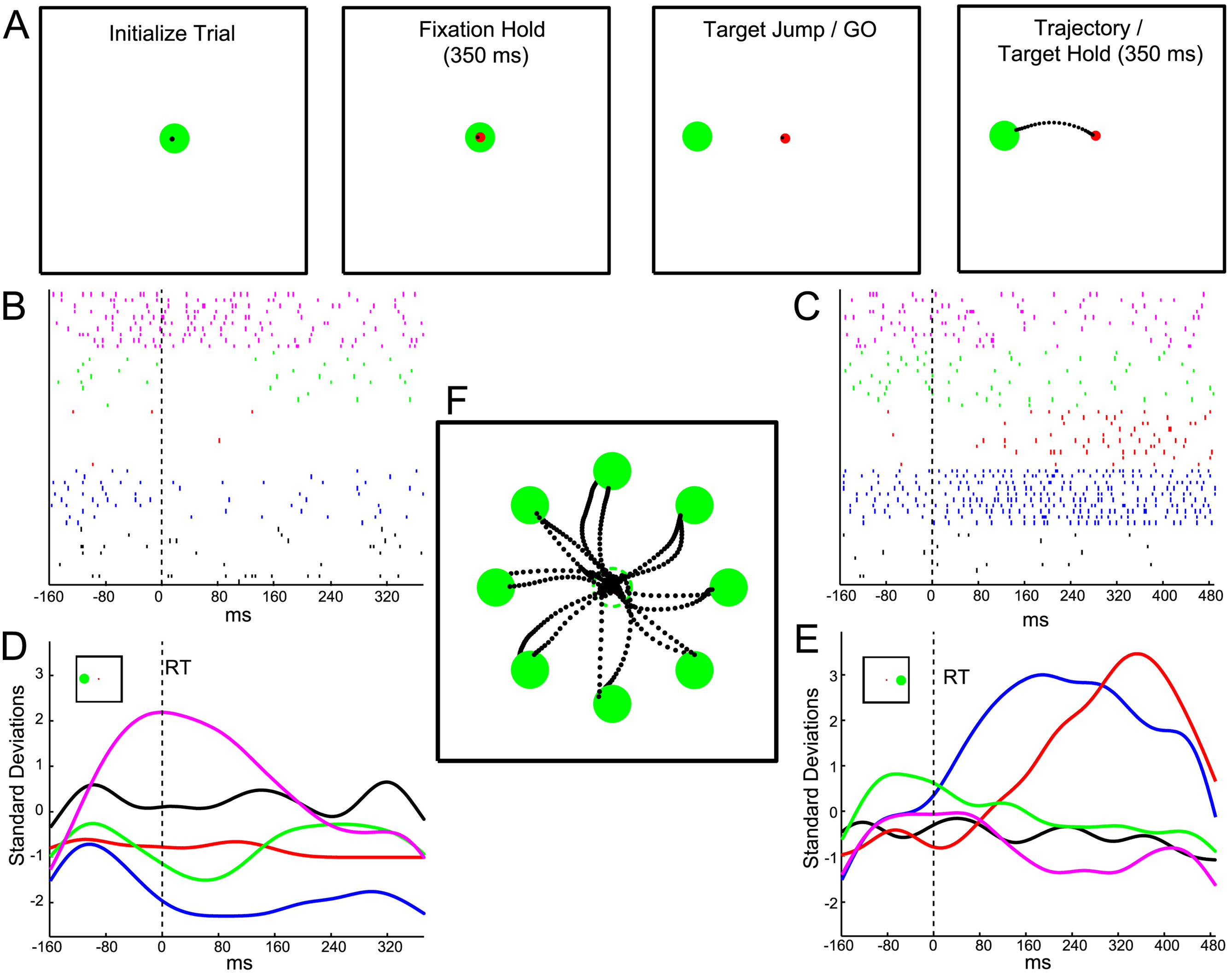

Figure 1. Center-out joystick task and example neural recordings obtained using adjustable-depth multielectrode array (AMEA). ***A***, Monkeys initiated each trial by guiding the cursor inside a central green circle. A concentric red circle then appeared, directing the monkeys to fixate centrally for 350 ms. The target was randomly jumped to 1 of 8 (or 12) possible targets, at which point the monkey initiated a trajectory to the peripheral target location. Monkeys held the cursor inside the target for at least 350 ms (100 ms for brain control) before receiving a juice reward. Raster plots show responses of 5 simultaneously recorded neurons during example trajectories to two different target locations, leftward (180°) (***B***) and rightward (0°) (***C***). Neural activity is aligned to the time of movement initiation (dashed vertical line) and is plotted up to 80 ms after the cursor entered the target zone. Standardized firing rate time courses for all 5 neurons (sorted by color) are plotted below their respective raster plots for both leftward (***D***) and rightward (***E***) target conditions. Note the spatial tuning present for two targets in this ensemble of 5 neurons. Smoothed (Gaussian kernel, SD = 20 ms) firing rate traces were generated for illustrative purposes here, while binned standardized firing rates (80 ms) were in fact used to train decoding algorithms (see Materials and Methods). ***F***, Example trajectories made by monkey 1 for all 8 targets. The dashed green circle is the starting location of the target and is not visible once the target has been jumped to the periphery. Dots represent cursor position sampled at 15 ms intervals along the trajectory.

component, which can vary in size depending upon the method used to obtain β, thereby producing an important tradeoff to be considered during model selection (Geman et al., 1992). The well known least-squares solution for β, which can be obtained in a single step, yields the minimum-variance, unbiased estimator. However, a zero-bias estimator often suffers from high MSE due to a large-variance component of the error. Often, it is beneficial during model selection to slacken the constraint on the bias to further reduce the variance component of the error. In particular, this is beneficial when confronted with a high-dimensional input space in which many neural signals may be correlated (e.g., overlapping receptive fields or autocorrelated firing rate inputs from different lag time steps), which may result in estimators that exhibit large variability over a number of different training sets. Ridge regression is a method that can optimize the bias-variance tradeoff by penalizing the average size of the coefficients in order to reduce the variance component of the error, while allowing a smaller increase in the bias (Hoerl and Kennard, 1970). For ridge regression, the traditional least-squares objective function is augmented by a complexity term, which penalizes coefficients for having large weights (Hastie et al., 2001), such that

$$\hat{\beta}^{\text{ridge}} = \underset{\beta}{\arg\min}\left\{ \underbrace{\sum_{k=1}^{M}\Big(x(k) - \beta_0 - \sum_{i,j}\beta_{i,j}r(k-i,j)\Big)^2}_{\text{Least-squares Term}} + \underbrace{\lambda\sum_{i,j}\beta_{i,j}^2}_{\text{Complexity Term}} \right\}, \tag{2}$$

where M is the number of training samples used in a session. The regularization parameter, λ, was determined iteratively by gradient descent using a momentum search algorithm, thereby minimizing MSE by converging to a set of coefficients that appropriately balances the bias and variance components of the reconstruction error. For a given value of λ, a unique solution for the ridge coefficients can be expressed conveniently in matrix notation, for example when estimating the 2D cursor position,

$$\hat{\beta}^{\text{ridge}} = (R^T R + \lambda I)^{-1} R^T X, \tag{3}$$

where $R \in \Re^{M \times 3N}$ is the standardized firing rate matrix sampled at 3 lag time steps into the past, $X \in \Re^{M \times 2}$ is the mean-subtracted, 2D position matrix, and $\beta \in \Re^{3N \times 2}$ are the model coefficients unique to a particular λ. Note that for $\lambda = 0$, the ridge filter is equivalent to the least-squares solution. Consequently, such a coefficient shrinkage procedure will reduce the effective degrees of freedom of the model, which is a scalar value that can be calculated using the expression (derivation available at www.jneurosci.org as supplemental material)

$$N_{df}(\lambda) = tr[R(R^T R + \lambda I)^{-1} R^T]. \tag{4}$$

Since ridge regression retains all of the original coefficients in the model, albeit shrunken versions of them, to make the model more parsimonious we also reduced the actual dimensionality of the neural input space by subselecting the largest (according to their squared magnitude) N_{df} coefficients and removing all others from the model. This imposed a kind of hard threshold on the noisy predictors that held little predictive power. We then repeated ridge regression using only those neural inputs that remained in the truncated input space, which typically improved cross-validated performance further over the initial ridge solution. This two-step, ridge-selection process was iterated until a minimum cross-validation error was reached. For our data sets, we found that this method performed comparably to or better than other regularization and variable selection methods such as lasso, the elastic net, or least angle regression (LARS) (Tibshirani, 1996; Efron et al., 2004; Zou and Hastie, 2005).

Goal-based Kalman filter. In addition to ridge regression we implemented a state-space decoding model, specifically a variant of the discrete Kalman filter (Kalman, 1960). The Kalman filter was originally proposed by Black and colleagues for decoding continuous trajectories from M1 ensembles (Wu et al., 2003) and has more recently been applied using electrocorticographic (ECoG) signals recorded from hand/arm motor cortex (Pistohl et al., 2008). Here we review its basic operation and emphasize differences taken in our implementation. Note that before training the Kalman filter we first applied ridge regression to the training data as described above to select the most informative subset of neural inputs to be used (as determined by the effective degrees of freedom). In addition to standard kinematic state variables [i.e., position (p), velocity (v), acceleration (a)] used in the above-mentioned M1 studies, we incorporated goal information into the state-space representation by augmenting the kinematic state vector with the static target position, T, that is,

$$x_k = [\,p_x, p_y, v_x, v_y, a_x, a_y, T_x, T_y]^T. \tag{5}$$

To avoid confusion with the standard Kalman filter, we will refer to this model as the goal-based Kalman filter (G-Kalman filter) for the remainder of this report.

Unlike the feedforward ridge filter, the G-Kalman filter operates in a recursive manner using two governing equations instead of one: an observation equation that models the firing rates (observation) as a function of the state of the cursor, x_k, and a process equation that operates in a Markov manner to propagate the state of the cursor forward in time as a function of only the most recent state, x_{k-1} (Welch and Bishop, 2006). Therefore, firing rates are not binned in multiple lag time steps before the measured state x_k, but instead reflect the firing rate of a single 80 ms bin just before (i.e., $t - 40$ ms) the measured state of the cursor. The G-Kalman filter assumes that these two models are linear stochastic functions, which operate under the condition of additive Gaussian white noise according to

$$x_k = A_k x_{k-1} + B u_{k-1} + w_{k-1}, \text{ (process equation)} \tag{6}$$

$$r_k = H_k x_k + v_k \qquad \text{(observation equation)}. \tag{7}$$

The control term, u, is assumed to be unidentified and is therefore set to zero in our model, excluding B from the process model. Following the approach of Wu and colleagues, we made the simplifying assumption that the process noise ($w \in \Re^{8 \times 1}$), observation noise ($v \in \Re^{8 \times 1}$), transition matrix ($A \in \Re^{8 \times 8}$), and observation matrix ($H \in \Re^{N \times 8}$) were fixed in time (Wu et al., 2003). Simplifying,

$$x_k = A x_{k-1} + w, \text{ where } p(w) \sim N(0,Q),\ Q \epsilon \Re^{8 \times 8}, \tag{8}$$

$$r_k = H x_k + v, \text{ where } p(v) \sim N(0,V),\ V \epsilon \Re^{N \times N}, \tag{9}$$

where A and H were solved using least-squares regression and Q and V are the process and observation noise covariance matrices, respectively. Note that the transition matrix, A, took the form

$$A = \begin{bmatrix} A^{\text{kin}} & C \\ 0 & I \end{bmatrix}, \tag{10}$$

where I is the 2×2 identity matrix, indicating that the target was fixed in time during the trial. The estimate of the kinematic state of the cursor, $x_k = [p_x,\ p_y,\ v_x,\ v_y,\ a_x,\ a_y]^T$, was therefore a linear combination of the previous kinematic state, which was weighted by the transition matrix $A^{\text{kin}} \in \Re^{6 \times 6}$, and the estimated target position, which was weighted by the linear operator $C \in \Re^{6 \times 2}$. That is, C biased the estimate of the kinematic state to be spatially constrained by the target position, which was inferred from the neural observation in Equation 9.

To estimate the state of the cursor at each time step k, the output of the process model, $\hat{x}_k^-$ (i.e., a priori estimate), was linearly combined with the difference between the output of the observation model and the actual neural measurement (i.e. the neural innovation) using an optimal scaling factor, the Kalman gain, $K_k \in \Re^{8 \times N}$, to produce an a posteriori estimate of the state of the cursor,

$$\hat{x}_k = \hat{x}_k^- + K_k(r_k - H\hat{x}_k^-). \tag{11}$$

This standard two-step discrete estimation process, consisting of an a priori time update and an a posteriori measurement update, was iterated recursively to generate an estimate of the state of the cursor at each time step in the trajectory. Note that neural innovation arises due to the deviation between the expected neural measurement (i.e., instantaneous firing rate that would be predicted by the observation model, given the a priori state estimate generated by the process model) and the actual firing rate measurement. When these two rates differ, a nonzero, weighted correction term is added to the a priori state estimate (i.e., an additive update), the magnitude of which is determined by the Kalman gain (a more detailed description of this process is available at www.jneurosci.org as supplemental material).

The initial position of the cursor was assumed to be known, but the initial velocity, acceleration, and target position were set to the expected value of their respective training distributions. Similarly, the initial variance of the position was set to zero, reflecting no uncertainty about the starting location of the cursor, while the variance for velocity, acceleration, and target position were derived from their training set distributions:

$$x_0 = [x_0 \ \ E[\dot{x}] \ \ E[\ddot{x}] \ \ E[T]] \tag{12}$$

$$P_0 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & var[\dot{x}] & 0 & 0 \\ 0 & 0 & var[\ddot{x}] & 0 \\ 0 & 0 & 0 & var[T] \end{bmatrix}, \tag{13}$$

where E is the expected value operator.

We assessed the decoding performance of the G-Kalman filter for the entire duration of the trajectory, though the G-Kalman filter typically converged to a steady-state value (i.e. Kalman gain and error covariance decreased exponentially to asymptote) <1 s after movement onset (further information available at www.jneurosci.org as supplemental material).

Model assessment. We quantified the off-line reconstruction performance of each decoding algorithm using a single statistical criterion, the coefficient of determination, or R^2. R^2 values for X and Y directions were averaged to give a single R^2 value for position, velocity, etc.

We also constructed neuron-dropping curves for each session to

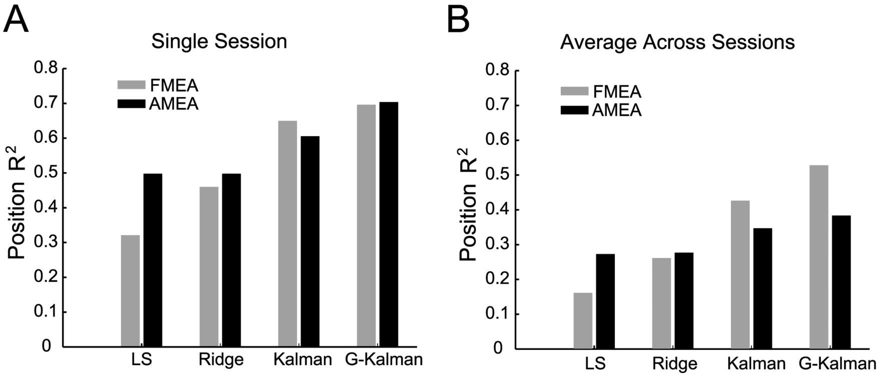

Figure 2. Off-line decoding performance for trajectory reconstruction. ***A***, Single session R^2 values for position estimation using the AMEA and fixed-depth multielectrode array (FMEA) techniques. Performances of several models are compared in the bar chart, including least-squares (LS), ridge regression, the Kalman filter, and the goal-based Kalman filter (G-Kalman). ***B***, Average R^2 performances for 10 FMEA sessions and 8 AMEA sessions for each of the four models.

quantify how well trajectories could be reconstructed using PPC ensembles of different sizes. To assess the performance of a particular ensemble size, q, we randomly selected and removed $N_{\mathrm{df}} - q$ neural units from the original ensemble, which contained N_{df} total neural units. Next, we performed fivefold cross-validation using the remaining q neural units to obtain an R^2 value, then replaced them, and repeated this procedure 50 times to obtain an average R^2 for each ensemble size, q. Note that an equal number of randomly selected trials were used from each session (300 trials); however we typically did not observe a significant improvement in performance after including much more than 100 trials.

We computed model fits for the resultant neuron-dropping curves so that they could be extrapolated to theoretical, larger ensemble sizes, allowing us to estimate the average R^2 performance for an arbitrary ensemble size. An exponential recovery function was fit to the data according to

$$R^2 = 1 - e^{-q/z}, \qquad (14)$$

where q was the ensemble size used and z was a fitted constant. For all FMEA and AMEA plots, the quality of fit obtained using this model was very good (average $R^2 > 0.98$).

Closed-loop brain control analysis. We assessed behavioral performance during brain control using two different measures. First, we computed the smoothed success rate as a function of trial number (trial outcome point process was convolved with Gaussian kernel, SD = 30 trials), as well as the average daily success rate for each of 14 brain control sessions. Second, we computed the average time necessary for the monkey to guide the cursor into the target zone successfully for each session, which was measured from target cue onset to 100 ms after the cursor entered the target zone. To calculate a chance level for success rate, we randomly shuffled firing rate bin samples for a given neural unit recorded during brain control, effectively preserving each neural unit's mean firing rate but breaking its temporal structure. Chance trajectories were then generated by simulation, iteratively applying the actual ridge filter to the shuffled ensemble of firing rates to generate a series of pseudocursor positions. Each chance trajectory simulation was allowed up to 10 s for the cursor to reach the target zone for at least 100 ms, the same criterion used during actual brain control trials. This procedure was repeated hundreds of times to obtain a distribution of chance performances for each session, from which a mean and SD were derived.

Neural unit waveform analysis. We quantified the signal-to-noise ratio (SNR) of neural unit waveforms according to the method described by Suner and colleagues (Suner et al., 2005). Using this method, SNR was calculated by dividing the average peak-to-peak amplitude for an aligned spike waveform by a noise term. The noise term was equal to twice the SD of the residuals that remained after subtracting the mean spike waveform from all waveform samples for a particular neural unit. The minimum threshold SNR required for multiunit classification was 2. Single units were identified using off-line sorting and always had interspike intervals >2 ms and typically had SNR values >4 when using this method.

Results

Off-line decoding

Our objective was to construct a decoding model that estimated task-related behavioral parameters (e.g., position, velocity, acceleration, target position) at multiple time steps in a trajectory from parallel observations of PPC neural activity. Training sets used for fivefold cross- validation were constructed from ~300 joystick trials per session. We analyzed a total of 20 different recording sessions off-line and found that we could reconstruct joystick trajectories with very good accuracy using ensemble activity from PPC, accounting for >70% of the variance in the cursor position during a single session ($R^2 = 0.71$). Ten recording sessions were performed using FMEAs implanted in one monkey, yielding recordings which were comprised of a combination of single units and multiunits (7.8 ± 5.2 single units, 107 ± 27 multiunits, $N_{\mathrm{df}} = 65 \pm 26$ effective neural units, mean ± SD). An additional set of 8 sessions was collected using AMEA recordings from a second monkey, all 8 of which contained exactly 5 simultaneously recorded single units (no multiunit activity was included in the AMEA ensembles). Note that off-line decoding analyses using FMEA data were performed for sessions recorded before the monkey had ever experienced brain control. Therefore, all off-line analyses reflected our ability to decode from PPC ensembles before the development of any learning effects that occurred as a result of brain control.

Position decoding performance: model comparisons

We evaluated the performances of several decoding algorithms, including least-squares, ridge regression, a standard Kalman filter and the G-Kalman filter. In general, when estimating the position of the cursor, we found that the G-Kalman filter outperformed the other three decoding algorithms, followed next by the Kalman filter, then the ridge filter and finally the least-squares model (Fig. 2*A*,*B*). For instance, for FMEA recordings, we found that ridge regression significantly improved the decoding performance when compared with the least-squares model ($p = 0.002$, two-sided sign test), on average resulting in a median improvement of 60% [first quartile (Q_1) = 45%, third quartile (Q_3) = 99%]. However, as expected we did not observe a significant change ($p = 0.25$, two-sided sign test) when using ridge for AMEA-recorded activity (0% median improvement, $Q_1 = 0$, $Q_3 = 1$). This discrepancy likely reflects the benefit of using ridge regression and subselection for high-dimensional input spaces (i.e., FMEA) that exhibited multicollinearity among neural inputs and/or contained little or no tuned activity on some channels. Consistent with this reasoning, we observed that the complexity parameter, λ, which was used to shrink the average size of the model coefficients, was significantly larger for FMEA data sets than for AMEA data sets ($p < 0.007$, Wilcoxon rank sum test). In addition, we found that for both AMEA and FMEA sessions, the standard Kalman filter significantly outperformed the ridge filter ($p = 0.001$, two-sided sign test) [42% improvement ($Q_1 = 34$, $Q_3 = 91$)]. This finding is in agreement with evidence from another study that showed that the Kalman filter outperformed the least-squares filter when decoding trajectories from M1 ensembles (Wu et al., 2003), and likely reflects the benefit of harnessing velocity information (that is encoded by some PPC neurons and inferred via Eq. 9) to estimate the changing position of the cursor (via the process model of Eq. 8).

Previous neurophysiological studies have established that PPC neurons encode a reach plan toward an intended goal location in eye-centered coordinates (Snyder et al., 1997; Andersen and Buneo, 2002; Gail and Andersen, 2006; Quiroga et al., 2006). In addition, other studies have shown that PPC maintains a strong representation of the target location even during continuous control of a movement (Ashe and Georgopoulos, 1994; Mulliken et al., 2008). Therefore we tested whether information about the static target location could be extracted from PPC and used to improve the decoding accuracy of cursor position in our task. Target information encoded by PPC can enter into the G-Kalman filter model via the neural innovation (i.e., Eq. 11), which is optimally transformed (i.e., by the Kalman gain) from a firing rate back into an 8-dimensional additive corrective term used to refine the a priori state vector estimate of Equation 8, ultimately influencing the accuracy of other key state variables, such as position and velocity. We found that the decoding performance was significantly improved when using the G-Kalman filter ($p < 1 \times 10^{-4}$, two-sided sign test), on average by 17% (median, $Q_1 = 9$, $Q_3 = 28$)), relative to the standard Kalman filter. This result demonstrates that significant goal information could be extracted from PPC, which was independent from the kinematic encoding of the state of the cursor. Furthermore, this information was harnessed by the G-Kalman filter to consistently improve the estimate of the dynamic position of the cursor.

It would be interesting to test the extent to which decoding performance can be improved if more accurate target information could be sought out and extracted from PPC [e.g., stronger target representations have generally been found in deeper regions of the IPS (Mulliken et al., 2008)]. Therefore, we simulated the hypothetical situation in which the target position could be inferred perfectly from the neural activity by assuming that the initial target position was given, and thereby remained fixed throughout the trial. This ideal case (though unrealistic, due to neural noise) would result in an approximately twofold average increase in performance (110% median increase, $Q_1 = 85$, $Q_3 = 163$) over the standard Kalman filter, resulting in a theoretical upper bound of R^2 values of 0.86 ± 0.06 (mean ± SD). Therefore a strong knowledge of the goal can provide valuable information that when integrated into a state-space framework significantly improves the ability to decode the dynamic state of the cursor. Example trajectory reconstructions (as well as average reconstructions) for multiple target locations obtained using the G-Kalman filter as well as the ridge filter are illustrated in Figure 3, along with the monkey's actual trajectories. Notice that the G-Kalman filter visibly out-performed the ridge filter, reconstructing paths that more closely followed the monkey's original trajectory.

AMEA versus FMEA decoding performance

On average, trajectory reconstructions obtained using the 64-channel FMEA technique were significantly more accurate than those obtained using the 6-channel AMEA technique ($R^2 = 0.53 \pm 0.10$ and 0.38 ± 0.19 for FMEA and AMEA, respectively). However, we also sought to test how decoding performance varied as a function of the PPC ensemble size. In other words, we aimed to quantify the efficiency of decoding, on a per neural unit basis, for the FMEA and AMEA techniques.

Figure 4*A* shows neuron-dropping curves for the single best FMEA and AMEA sessions, which plot R^2 for decoding cursor position as a function of ensemble size. First, notice how the G-Kalman filter significantly outperformed the ridge filter (*t* test, $p < 0.05$). In other words, using the G-Kalman filter, fewer neural units were needed on average, to obtain an R^2 equal to that achieved by the ridge filter. Second, although both recording techniques ultimately achieved high decoding accuracy for these particular sessions (e.g., using the G-Kalman filter, AMEA: $R^2 = 0.71$, FMEA: $R^2 = 0.70$), the gain in performance per neural unit was significantly higher for the AMEA session than for the FMEA session [e.g., for ensemble sizes ranging from 1 to 5 units (*t* test, $p < 0.05$)]. For example, decoding from just 5 single units using the AMEA technique, we were able to account for >70% of the variance ($R^2 = 0.71$) in the cursor position. In comparison, the best FMEA session yielded an average R^2 of 0.05 ± 0.11 using 5 neural units. By extrapolating the FMEA trend forward, it would require ~191 units for the FMEA technique to match the R^2 of 0.71 achieved by the AMEA.

To verify that these single-session trends were representative of the decoding efficiency differences between AMEA and FMEA techniques in general, we averaged neuron dropping curves from 8 AMEA sessions and 10 FMEA sessions. Note that when averaging performances across all 10 FMEA sessions, the maximum possible number of neural units considered was constrained by an upper limit of 75, the smallest ensemble size, N_{df}, of any of the 10 FMEA sessions we considered. As expected, session-averaged neuron-dropping curves were shifted downward from the single best session curves due to variation in decoding performance across multiple sessions (Fig. 4*B*). Nonetheless, trends analogous to those reported for single best sessions in Figure 2*A* were preserved after averaging across sessions. When decoding with 5 neural units, an R^2 of 0.38 ± 0.12 was achieved using the AMEA technique while an R^2 of 0.08 ± 0.08 was obtained using the FMEA approach for just 5 units. According to the FMEA curve, it would require ~58 neural units, on average, to reach an R^2 of 0.38, which the AMEA approach was able to achieve using just 5 neural units. These data suggest that the AMEA technique was significantly more efficient at extracting information from neural ensembles in PPC than the FMEA technique.

Reconstruction of behavioral and task parameters using PPC activity

We also tested how well the G-Kalman filter could estimate several other behavioral and task parameters. Figure 5 illustrates a 4 s time course of position, velocity, and acceleration of the cursor as well as the target position, which were constructed by concatenating segments from 9 test trials randomly selected from a single session. Superimposed onto the plot are the estimated values of these parameters, which were generated using the G-Kalman filter. Notice that the estimated values of position, velocity, and target position followed their corresponding experimental values reasonably closely. As expected, estimation of acceleration from PPC activity was contrastingly less reliable since physical measurements of acceleration tend to be comparably noisy due to the higher frequency characteristic of acceleration. For this session, using the G-Kalman filter we found that PPC activity was capable of yielding R^2 estimation accuracies of up to 0.71, 0.57, 0.25, and 0.64, for position, velocity, acceleration, and target position, respectively. On average, across all FMEA sessions, when using the G-Kalman filter we found that $R^2_{\mathrm{position}} = 0.53 \pm 0.11$, $R^2_{\mathrm{velocity}} = 0.33 \pm 0.08$, $R^2_{\mathrm{acceleration}} = 0.10 \pm 0.05$, and $R^2_{\mathrm{target}} = 0.40 \pm 0.08$. In comparison, for the 8 AMEA sessions we found that $R^2_{\mathrm{position}} = 0.38 \pm 0.22$, $R^2_{\mathrm{velocity}} = 0.26 \pm 0.18$, $R^2_{\mathrm{acceleration}} = 0.10 \pm 0.08$, and $R^2_{\mathrm{target}} = 0.32 \pm 0.22$.

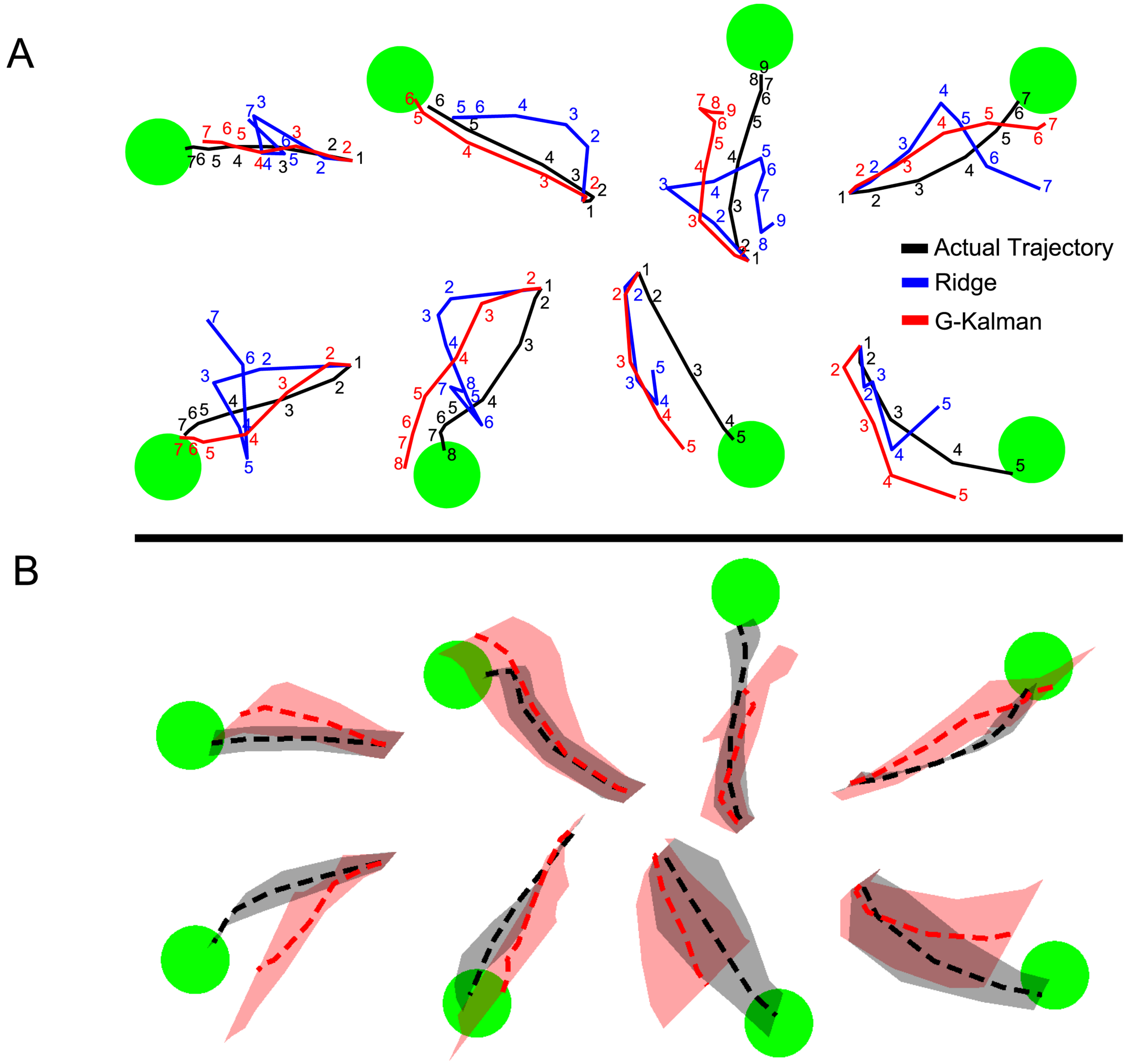

Figure 3. Representative off-line trajectory reconstructions from single best session. ***A***, Single-trial reconstructions from 8 different test set trials during a single AMEA session. Actual trajectories as well as reconstructions obtained using both ridge regression and the G-Kalman filter are shown for each trial. Numbers along each trajectory indicate the temporal sequence (time steps are labeled every 80 ms) of the cursor's path. Note the visible performance improvement obtained when using the G-Kalman filter compared with ridge regression. ***B***, Average reconstructions for a particular target (dashed red and black traces denote mean G-Kalman prediction and actual trajectory, respectively). Confidence bands denote SDs of *X* and *Y* position from the mean position at each time step. Target-specific averages were performed only for trajectories of equal duration (i.e., the mode of the distribution of trajectory durations measured for a particular target). Therefore, these trajectory bands represent a useful visual representation of typical reconstruction performance measured during this AMEA session, but importantly underestimate the actual variability normally present in the full trajectory data set.

Lag time analysis

The temporal encoding properties of PPC neurons have been studied before for continuous movement tasks (Ashe and Georgopoulos, 1994; Averbeck et al., 2005; Mulliken et al., 2008). Recently, we suggested that PPC neurons best encode the changing movement direction (and velocity) with approximately zero lag time (Mulliken et al., 2008). That is, the firing rates of PPC neurons were best correlated with the current state of the movement direction, a property consistent with the operation a forward model for sensorimotor control (Jordan and Rumelhart, 1992; Wolpert et al., 1995). Here, we test an extension of this hypothesis, at what lag time can the dynamic state be decoded best from PPC population activity? We decoded the state of the cursor shifted in time relative to the instantaneous firing rate measurement, with lag times ranging from −300 to 300 ms, in 30 ms steps (where negative lag times correspond to past movement states and positive lag times correspond to future movement states). (Note that since the prediction accuracy for acceleration was poor and the target position was stationary in time, results for those parameters are not included.) Since firing rates were averaged over 80 ms bins, the instantaneous firing rate was assumed to be measured at the middle of a bin (i.e., 40 ms mid-point). To fairly combine results across sessions that had variable decoding performances, each session's R^2 values were normalized by the maximum R^2 measured for that session (for both position and velocity). The optimal lag time (OLT) for decoding velocity using the G-Kalman filter was 10 ms in the future, consistent with previous claims that PPC best represents the current state of the velocity (Mulliken et al., 2008). The position of the cursor was best decoded slightly further into the future, at an OLT of ~40 ms. These temporal decoding results suggest that the current or upcoming state of the cursor could be best extracted from the PPC population using the G-Kalman filter.

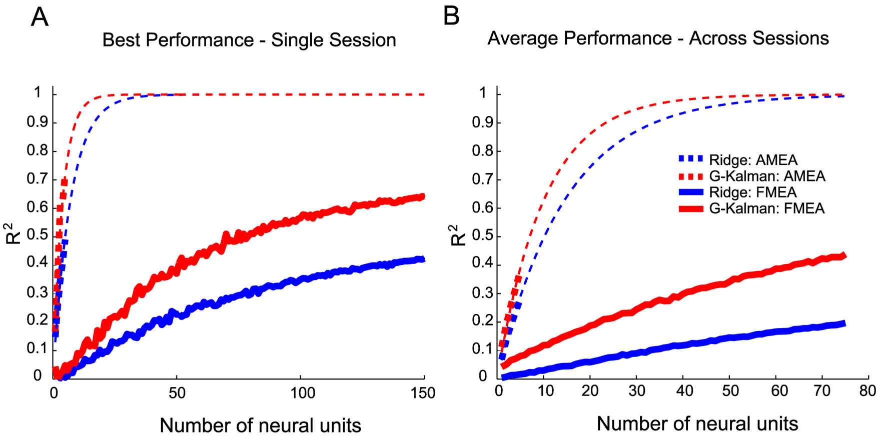

Figure 4. Neuron-dropping curves comparing AMEA and FMEA decoding efficiencies. ***A***, Recordings from a single session showed significant decoding efficiency (R^2/unit) advantages when using the AMEA technique compared with the FMEA approach. ***B***, These decoding efficiency differences were consistently observed when averaged across multiple sessions (8 AMEA, 10 FMEA sessions). It is important to note that extrapolations made for AMEA neuron-dropping curves were performed only for illustrative purposes and should not be considered, and are not used as, as accurate quantitative estimates of R^2 for ensemble sizes much larger than 5 neurons. Future AMEA studies will be necessary to collect data to confirm or deny such speculative AMEA projections. Although not shown, this FMEA session reached an R^2 of 0.70 at ~180 neural units when using the G-Kalman filter.

Neural responses that encode the current state of the cursor are most likely not derived solely from delayed passive sensory feedback into PPC [proprioceptive delay >30 ms, visual delay >90 ms, i.e., OLT < −30 ms (Flanders and Cordo, 1989; Wolpert and Miall, 1996; Ringach et al., 1997; Petersen et al., 1998)], nor is such a representation compatible with the generation of outgoing motor commands [M1 neurons generally lead the state of the movement by >90 ms, i.e., OLT >90 ms (Ashe and Georgopoulos, 1994; Paninski et al., 2004)]. Furthermore, it is unlikely that a current-state representation may simply reflect a passive mixture of distinctly sensory and distinctly motor signals present within PPC. For instance, previous analyses have shown that the central tendency of hundreds of PPC neurons (i.e., the median of a roughly unimodal distribution of OLTs) largely encoded the current state of the movement direction (i.e., velocity) (Mulliken et al., 2008). Furthermore, those cells that exhibited near 0 ms OLTs encoded significantly more information about the movement direction than cells with other OLTs (e.g., purely sensory or purely motor), which is consistent with PPC actively computing these current state estimates. Therefore, we suggest that the dynamic state representation in PPC may best reflect the active computational process of a forward model for sensorimotor control, which creates an estimate of the current and upcoming state of a movement using efference copy signals received from frontal areas and a forward model of the movement dynamics of the arm.

Brain control

In addition to reconstructing trajectories successfully off-line, we tested whether we could continuously control a cursor in real-time using a ridge filter that operated directly on the monkey's neural activity to estimate the position of the cursor. We decided to use the ridge filter initially because it provides a simple framework (i.e., feedforward, one-to-one linear mapping between neural activity and a single parameter, the 2D position of the cursor) that is convenient for systematically assessing the learning effects that occur in PPC. The ridge filter was trained for each session using data recorded during the joystick training segment, and remained fixed throughout the brain control segment of each session. Typically, the monkey performed several hundred brain control trials per session.

Behavioral performance

We found that the monkey was able to successfully guide the cursor to the target using brain control at a level of performance much higher than would be expected by chance. Figure 6*A* illustrates the monkey's behavioral performance for the first brain control session, which was fair. The 30-trial moving average of success rate varied up and down during the first session, but on average was 32% for 8 targets, which was significantly higher than the chance level calculated for that session (chance = 5.2 ± 2.3%, mean ± SD). However, after just three additional sessions, the monkey's performance had improved substantially, reaching a session-average success rate of 80%, and stabilizing around that level for several days (Fig. 6*C*). For instance, during session 6 the session-average success rate was 80% and 30-trial average climbed as high as 90% during certain periods of the session, performing far above chance level (Fig. 6*B*). Figure 6*C* also illustrates the median time needed for the cursor to reach the target for each of the 14 brain control sessions. Concurrent with an increase in success rate during the first several sessions, we observed a complementary decrease in the time to target. For instance, during the first session the median time to target was slightly more than 3 s and by the fourth session it had dropped to 883 ms. As expected, these two parameters were strongly anticorrelated ($\rho = -0.96$, $p < 1 \times 10^{-7}$, Pearson correlation coefficient). Both an increase in success rate and a concomitant decrease in the time to target show that the monkey was able to learn to proficiently control the cursor using visual feedback of the decoded cursor position on the computer screen. Remarkably, these learning mechanisms became evident over a brief period of 3–4 d. Improvements in brain control performance began to saturate after several days and remained high until the recording quality of the FMEA implant started to degrade.

Our initial assumption was that after many sessions, the monkey would voluntarily cease to move his hand (without restraint) as his neural activity became progressively better at controlling the cursor. However, the monkey's hand did continue to move during many brain control trials. Interestingly, in a control session we observed that joystick movement was not necessary for successful brain control performance after immobilizing the joystick (i.e., the monkey still performed brain control at 60% performance with a rigidly fixed joystick). It is conceivable that had our chronic recording lasted for a longer time period, the monkey

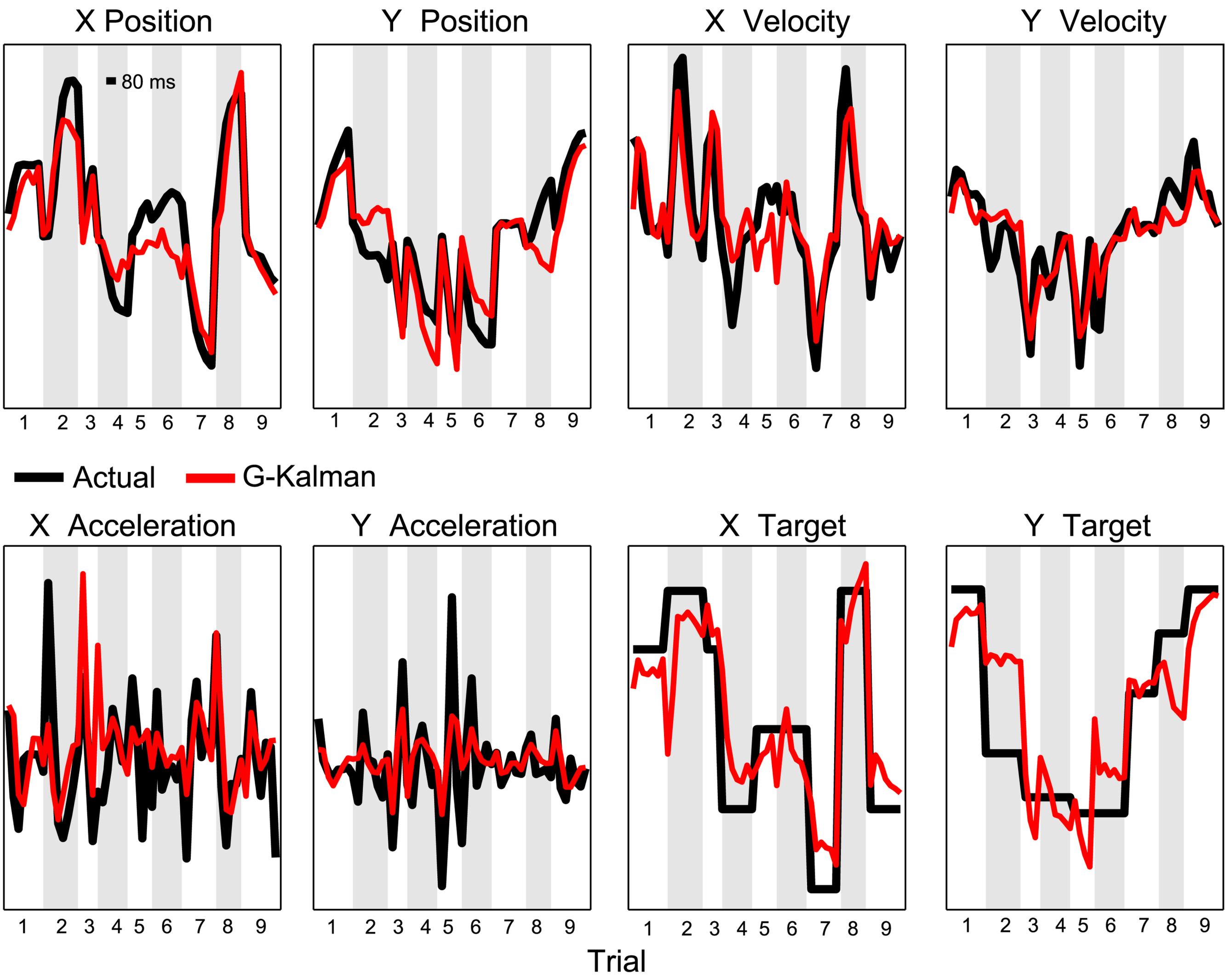

Figure 5. Reconstruction of various trajectory parameters using PPC activity. Actual behavior and decoded estimates of position, velocity, acceleration, and target position time series for 9 concatenated trials that were randomly selected from a single AMEA session. All estimates shown were generated using the G-Kalman filter. Alternating gray and white backgrounds denote time periods or different trials. The scale bar in the "*X* Position" panel depicts 80 ms duration.

might have stopped moving his hand altogether as has been demonstrated by other studies in M1 (Carmena et al., 2003). However, future investigations will be necessary to more thoroughly explore this possibility in PPC.

Several example successful trajectories made by the monkey during brain control are illustrated in Figure 7. Brain control trajectories for which the monkey guided the cursor rapidly and directly into the target zone are illustrated in Figure 7*A*. Figure 7*B* shows examples of trajectories that initially headed away from the target and required correction to reach the target. The ability of the monkey to rapidly adjust the path online suggests that he learned to modulate his neural activity in order to steer the cursor to the goal using continuous feedback of the visual error.

Brain control learning effects

Changes in neural activity were monitored in parallel with behavioral performance trends by analyzing PPC population activity recorded during each session's joystick training segment, before the brain control segment. Specifically, we averaged the learning effects across all neural units that were included in a session's ensemble. Importantly, when defining the PPC ensemble used for off-line decoding assessment, we did not assume that we recorded from exactly the same ensemble of neural units from session to session (though this probably occurred to some extent) because (1) we did not continuously track the spike waveforms 24 h per day so as to ensure that neurons were identical, and (2) it is difficult to robustly determine to what extent multiunit activity consistently reflected the same combination of single units from day to day. Instead, for each session we chose to include only those neural units whose spike waveform signal-to-noise ratio (SNR) exceeded some arbitrary threshold value (see Materials and Methods) when decoding off-line. In general, the average SNR value for an ensemble (2.7 ± 0.08) as well as the number of single units (5.9 ± 2.9) did not vary significantly over the course of 14 brain control sessions, nor did we any observe any significant linear trend in their respective values [the slopes of line fits for these two trends were not significantly different from zero, that is, the 95% confidence intervals (CI) of the slope included zero].

We observed noteworthy changes in the neural activity that demonstrated strong evidence of learning in the PPC population. First, we calculated the average R^2 for decoding cursor position from the ensemble (off-line analysis using ridge filter), which is illustrated for all 14 sessions in Figure 8*A*. Notice that the R^2 trend approximately followed the trend for behavioral perfor-

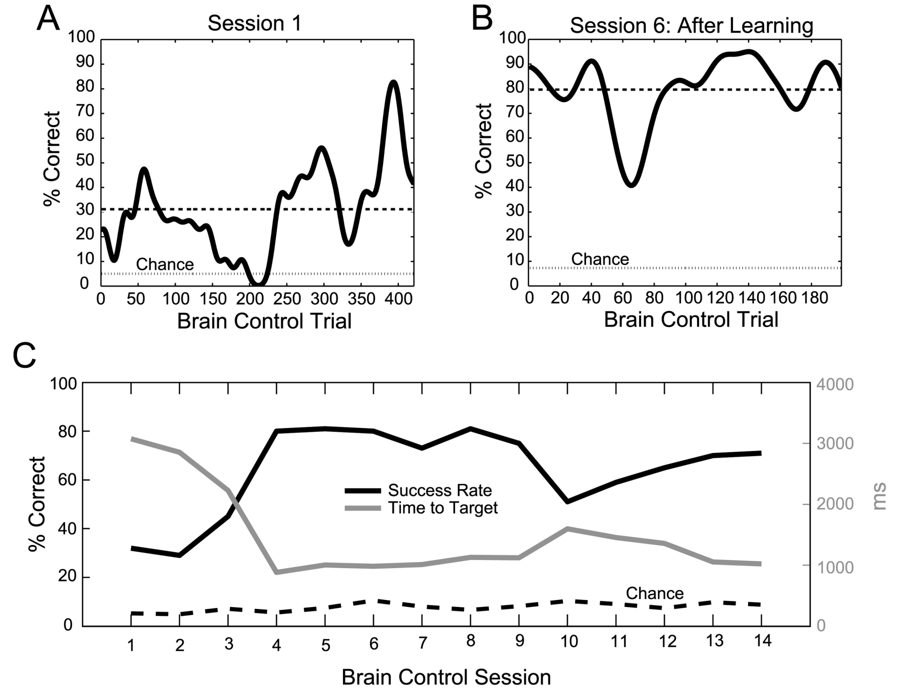

Figure 6. Brain control performance improvement over multiple sessions. ***A***, Thirty-trial averaged success rate during the first closed-loop, brain control session. Dashed line denotes average success rate for the session, and lighter dashed line denotes the chance level calculated for that session. ***B***, Improved brain control success rate measured during session 6, after learning had occurred. ***C***, After several days, behavioral performance improved significantly. Session-average success rate increased more than twofold and the time needed for the cursor to reach the target decreased by more than twofold.

mance, increasing shortly after the first brain control session and leveling off after several more sessions. The maximal session R^2 using ridge regression was 0.64 (or $R^2 = 0.80$ if G-Kalman had been used), more than doubling the decode performance obtained on the first day of brain control, which had an R^2 of 0.25. The off-line decoding R^2 was strongly correlated with online brain control performance on a session by session basis ($\rho = 0.60$, $p = 0.02$, Pearson correlation coefficient). This result suggests that when presented with continuous visual feedback about the decoded position of the cursor during brain control, PPC neurons were able to collectively modify/improve their encoding properties (as evidenced by an increase in off-line decoding performance), effectively making more information available to the ridge filter for controlling the cursor during subsequent brain control sessions.

In addition to assessing decoding performance while the monkey learned to perform under brain control conditions, we investigated the trends of various tuning properties of the PPC population that might have been responsible for the increase in R^2. When assessing changes in tuning, we included only the most informative neural units in the ensemble as determined by the effective degrees of freedom provided by the ridge model (i.e., the subset of the original ensemble that contributed most significantly during off-line and online decoding, $N_{\mathrm{df}} = 77 \pm 20$ neural units, mean ± SD). For each joystick training session, we constructed a 2D position tuning curve for each neural unit in the ensemble; using firing rates belonging only to the most recent lag time bin. (Note that similar results were obtained for all three lag time bins, but strongest tuning was typically observed for the first lag time bin). *X* and *Y* cursor positions were discretized into a 6 × 6 array of 36 bins, extending ±10° in the *X*–*Y* plane. Accordingly, each *X*–*Y* bin contained a distribution of firing rate samples corresponding to sample cursor positions measured at a particular 2D position (typically >50 samples). First, to quantify the tuning depth, we performed a nonparametric Wilcoxon rank sum test (i.e., normal approximation method) using the two firing rate bin distributions that contained the maximum and minimum mean values. The normal *Z*-statistic that resulted from this test was defined as the tuning depth of the position tuning curve. Using this metric, we then computed the ensemble's average tuning depth for each session. We found that tuning depth increased most substantially over the first 8 brain control sessions (by ~70%) and approximately leveled off for the remaining sessions. When fitting a line to this increasing trend, we found that the slope was significantly larger than zero ($m = 0.09$, 95% CI: 0.04, 0.14). An increase in ensemble tuning depth can be interpreted as an expansion of the firing rate dynamic range, boosting the effective gain of the ensemble for the purpose of encoding position (Fig. 8*A*). Second, we quantified the spread of 2D tuning for the PPC population by calculating the SD of the preferred positions of all neural units in an ensemble. The preferred position of a neural unit was defined as the *X* and *Y* position that corresponded to the bin with the maximum average firing rate. To obtain a scalar measure for dispersion of tuning for the ensemble, we averaged the SD of the *X* and *Y* preferred position distributions. Similar to the trend we observed for tuning depth, we found that the average spread of preferred positions increased significantly throughout the brain control sessions (Fig. 8*B*), ultimately to ~35% above its starting level). The slope of a line fit to this trend was also significantly larger than zero ($m = 2.11$, 95% CI: 1.18, 3.04). This increase in the spread of tuning by the PPC population presumably enabled the monkey to control the cursor over an increasingly broader range of 2D space on the computer screen during brain control. Third, we tracked the tuning curve overlap between all possible pairs of neural units in an ensemble, which were then averaged to give a scalar ensemble tuning overlap value. (Note, tuning overlap was computed as the normalized dot product of two vectorized tuning curve matrices.) We found that the ensemble tuning overlap increased only slightly (by ~6%) (Fig. 8*B*), but significantly, over 14 brain control sessions ($m = 0.004$, 95% CI: 0.002, 0.007). This increase in ensemble tuning overlap suggests that the PPC population code became slightly more redundant during brain control learning.

Summarizing these trends in position tuning, we found that both the tuning depth and tuning spread of the ensemble increased substantially during brain control. Importantly, both tuning depth and tuning spread showed strong correlations with R^2 decoding performance, which were highly significant ($\rho = 0.71$, $p = 0.004$ and $\rho = 0.73$, $p = 0.003$, respectively, Pearson's linear correlation coefficient), and consequently with brain control performance as well ($\rho = 0.68$, $p = 0.007$ and $\rho = 0.61$, $p = 0.02$, respectively). These correlation results suggest that adjustment of these particular tuning properties was necessary in order for the ensemble to improve off-line decoding performance, and ultimately for the monkey to improve his performance during brain control. Tuning overlap was also correlated with R^2 performance ($\rho = 0.62$, $p = 0.02$). (Note however that tuning overlap

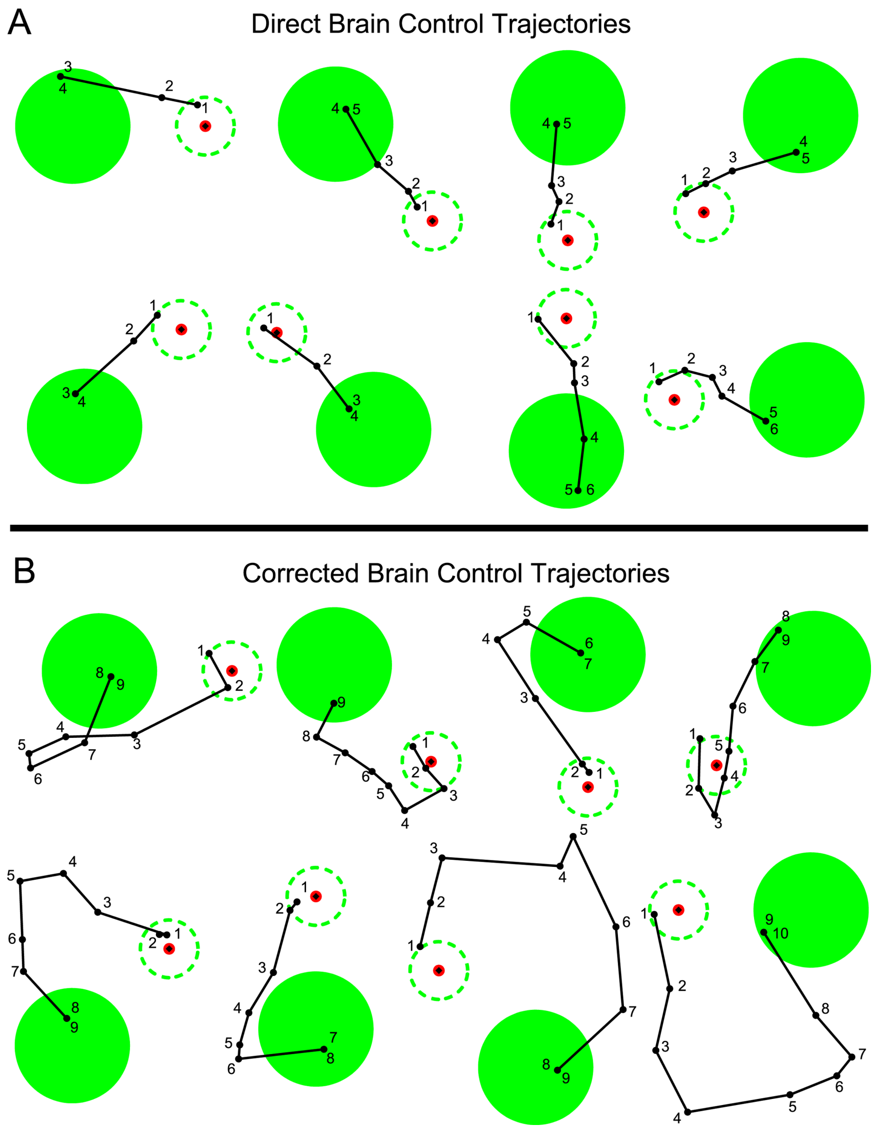
A
Direct Brain Control Trajectories
B
Corrected Brain Control Trajectories

Figure 7. Examples of successful brain control trajectories from monkey 2 during session 8, illustrating trajectories directed toward the target (***A***) as well as trajectories that initially headed off-course and therefore required online correction (***B***). Brain control targets were made approximately twice as large as target stimuli presented during the training segment (i.e., during joystick trials) to allow the monkey to perform the task successfully during early brain control sessions. So that behavioral performance and learning effects could be compared fairly across multiple sessions, we kept the target size constant, even after performance had improved.

showed a weaker (and less significant) correlation with brain control performance ($\rho = 0.46$, $p = 0.09$)). An increase in tuning overlap probably reflected the fact that more neural units became significantly tuned over the course of the 14 brain sessions, thereby increasing the likelihood of overlapping tuning curves within the ensemble. However, it is also possible that the population deliberately broadened the average width of a typical neural unit's tuning curve. Unfortunately, this possibility was difficult to evaluate with our data as positional tuning curves generally comprised a variety of functional forms, including planar, single-peaked, and occasionally multipeaked representations. Future studies will need to address the extent to which redundancy in the population arises due to an increase in the percentage of tuned neural units in the ensemble versus a broadening of the tuning curves of those constituent neural units.

Finally, it is unlikely that the improvement in R^2 performance and enhanced ensemble tuning was trivially due to a sudden increase in newly appearing neurons that happened to have stronger tuning properties than previously recorded ensembles. First, during joystick sessions before the initial brain control session, the off-line R^2 performance achieved using the ridge filter fluctuated slightly from day to day, but typically fell within a limited range. For example, the distribution of performances for the 7 sessions before beginning brain control was $R^2 = 0.27 \pm 0.03$ (mean ± SD). Second, as mentioned above, the number of single units and the session SNR did not change significantly during brain control. Finally, we did not observe a significant correlation between R^2 and the session-averaged SNR ($\rho = 0.28$, $p = 0.33$). Therefore, the most reasonable interpretation of the substantial improvement in decoding performance we observed is that the information content of the PPC ensemble increased due to plasticity effects characterized by the changes in tuning we report, and did not occur by the sudden chance appearance of new, tuned neurons at the tips of our electrodes.

Discussion

In this study, we showed that 2D trajectories of a computer cursor controlled by a hand movement could be reliably reconstructed from the activity of a small ensemble of PPC neurons. We were able to account for >70% of the variance in cursor position when decoding off-line from just 5 neurons. This high decoding efficiency (R^2 per neural unit) likely reflects the ability to reliably isolate neurons in PPC using the AMEA recording technique. Consistent with findings from M1 decoding studies, we verified that state-space models (i.e., Kalman filter variants) significantly outperformed feedforward linear decoders (e.g., least-squares, ridge regression) (Wu et al., 2003). In addition, by incorporating information inferred about the goal of the movement (i.e., target position) into the state-space of the Kalman filter, we were able to significantly improve the accuracy of the dynamic estimate of cursor position. Finally, we showed for the first time that PPC ensembles alone are sufficient for controlling a cursor in real-time for a neural prosthetic application. Furthermore, we observed significant and rapid learning effects in PPC during brain control, which enabled the monkey to substantially improve behavioral performance over several sessions.

Considerations for off-line decoding of trajectories from PPC

Off-line reconstruction of 2D trajectories from PPC activity has been reported in an earlier study using FMEA recordings (Carmena et al., 2003). In one version of their task that was similar to ours, monkeys used a pole to guide a cursor to random target locations on a computer screen. [Note however, since contextual differences exist between these two tasks (e.g., targets could ap-

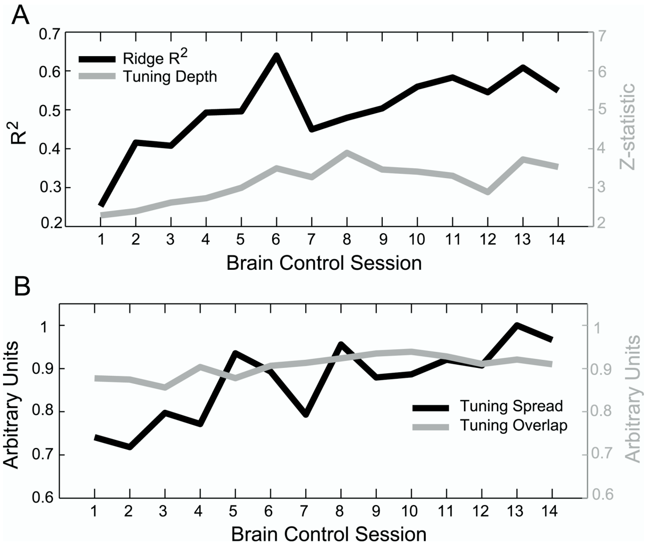

Figure 8. PPC learning effects due to brain control (off-line analyses). ***A***, Off-line decoding performance illustrates that the PPC population was able to increase the amount of information that could be decoded using ridge regression. The tuning properties of the population also showed significant learning trends over 14 brain control sessions. Both the *Z*-statistic of the tuning depth (***A***) and the SD of the preferred positions (***B***) for the ensemble increased significantly over 14 sessions. The average ensemble tuning curve overlap also increased significantly during brain control learning, however to a lesser extent (***B***).

pear anywhere in 2D space in their task while we had 8–12 fixed target locations in our task, our monkeys maintained central fixation while theirs exhibited free gaze), some caution is advised when drawing quantitative comparisons made between these two studies below.] Using a least-squares model, Carmena and colleagues decoded from 64 neural units in PPC (multiunits and single units) and concluded that they could reconstruct cursor position with relatively poor accuracy using PPC activity ($R^2_{\mathrm{pos}} =$ 0.25, single session). In addition, they did not observe a significant improvement in R^2 when using a Kalman filter instead of a least-squares model. For our FMEA recordings, we found that similar performance could be achieved on average using linear feedforward models (Fig. 2*B*), although a single FMEA session yielded an R^2 of 0.46. Nonetheless, we found that position could be decoded with much higher accuracy when using the G-Kalman filter, achieving an average R^2 of 0.53 over 10 FMEA sessions, and up to 0.71 during a single FMEA session. Moreover, when recording from only 5 single units using the AMEA technique, we obtained very accurate trajectory reconstructions (up to $R^2 = 0.71$) that approached the accuracy found for M1 ensembles ($R^2 = 0.73$ for 33–56 single units and multiunits) reported by Carmena and colleagues. Finally, after the monkey learned to improve his performance during brain control sessions, we found that the off-line R^2 rose to 0.80, accounting for 80% of the variance in cursor position. Therefore, we emphasize here that 2D trajectories can be reconstructed very well using PPC activity and further propose that AMEA techniques may be used to improve decoding efficiency when recording from PPC ensembles.

The G-Kalman filter we present here is similar conceptually to state-space decoding algorithms reported by other groups that make use of target-based information. For example, Yu and colleagues recently described a linear-Gaussian state-space framework in which the process model of Equation 6 (i.e., "trajectory model") is effectively comprised of a mixture of trajectory models, each of which is tailored for a distinct target location belonging to a predetermined set of targets (Yu et al., 2007). Another recently proposed state-space model described a solution for the time-varying command term, u_k, of the process model of Equation 6 (Srinivasan et al., 2006) that is conditioned upon the inferred target location, and which assumes a known arrival time for the cursor to reach the target. These methodological advances suggest that goal information extracted from PPC could be used to improve the decoding performance of a standard state-space model. Here we confirm this hypothesis using a simpler model, the G-Kalman filter, which uses a single process model and does not assume that the target arrival time is known.

The G-Kalman filter uses information about the static target location inferred from PPC spiking activity to improve the estimate of the dynamic state of the cursor. Interestingly, a recent neurophysiological study provided evidence that the local field potential (LFP) signal in PPC provides another source of information about the intended endpoint of an impending reach (Scherberger et al., 2005). Therefore, future algorithms that incorporate goal information from the LFP into an aggregate neural observation, both during preparation and execution of the movement, may further improve the accuracy of the dynamic estimate of the state of the cursor.

In a neural prosthetic application, it would be advantageous for PPC to encode target/goal information, which can be exploited by a goal-based decoding algorithm, even under conditions when the target is not visible. We suggest that it is unlikely that the target/goal representation in PPC is purely visual, which would imply that this signal would be abolished under conditions when the target stimulus is invisible. Evidence from memory reach studies have shown that MIP and area 5 encode the intended endpoint of an impending reach well after a target cue has been extinguished (Snyder et al., 1997; Andersen and Buneo, 2002; Musallam et al., 2004). Furthermore, in an anti-reach memory task, PRR neurons were shown to encode the intended reach endpoint even when it was dissociated 180° from an extinguished cue stimulus (Gail and Andersen, 2006). In addition, we recently showed that information about the target was strongly represented by PPC neurons during the middle segment of a trajectory, suggesting that PPC neurons were able to maintain their target representation during online control of a trajectory, even >220 ms after cue onset (Mulliken et al., 2008). Based on these findings, we predict that in a hypothetical memory joystick task, both before and during execution of a trajectory, PPC neurons will continue to encode both the goal/target location in absence of a target stimulus, while also maintaining information about the dynamic state of the trajectory. Nonetheless, this hypothesis needs to be tested directly. Additionally, future studies will also need to address what effect an array of distractor stimuli may have on the representation of a singular movement goal in PPC.

Another open question is the coordinate frame in which both target and kinematic information is encoded by PPC. Interestingly, in our previous study (Mulliken et al., 2008), we presented anatomical evidence that showed that, on average, more information was encoded about the goal/target the deeper one recorded in the IPS, while a dynamic representation of the movement direction of the cursor/hand was more strongly represented in surface areas of PPC. An interesting and testable hypothesis for future investigation is that goal and kinematic signals may operate largely in two different coordinate systems, that is, in eye- and limb-centered coordinates, respectively.

Decoding efficiency: AMEA versus FMEA

The ability to accurately decode trajectories using only five neurons underscores the contrasting decoding efficiency of AMEA and FMEA techniques. Several factors might account for this discrepancy in efficiency. During insertion, the placement of electrodes using the FMEA technique is effectively random with respect to the probable location of a neuron(s). In contrast, AMEA recordings enable the experimenter to "optimally" position the depth of every electrode using electrophysiological feedback. While we did not intentionally target cells that contained specific or desirable tuning properties, we did optimize several other factors using the AMEA technique: (1) we optimized extracellular isolation quality systematically, maximizing a given channel's SNR and importantly, facilitating spike sorting; (2) when recording from PRR neurons that were embedded up to 8–10 mm in the convoluted bank of the IPS, we were able to consistently target gray matter and presumably cortical layers that contained task-related neurons [note, this is in contrast to recordings made in surface areas of cortex (e.g., precentral gyrus of M1), where the depth of targeted neurons, for example layer 4 pyramidal cells, is considerably less variable and can be targeted more reliably]; and (3) we lowered electrodes into a "mapped" region of the recording chamber, known from prior sessions to contain task-related neurons.

Learning to control a cursor using continuous visual feedback

Learning effects in PPC became evident quite early during brain control sessions, more than doubling the off-line decoding performance (R^2) of the population after 5–6 sessions. Studies in M1 have reported brain control learning effects of comparable magnitude to the R^2 changes we observed here, however, these changes typically occurred over the course of 20 sessions or more (Taylor et al., 2002; Carmena et al., 2003). In particular, Carmena and colleagues reported significant learning effects in multiple cortical areas, including PMd, primary somatosensory cortex (S1), and supplementary motor area (SMA) during brain control. However, they did not report any change in R^2 decoding performance for parietal cortex. Instead, they reported relatively small changes (~25–30% increase over 14 sessions) in the tuning depth of parietal neurons compared to motor cortex cells which approximately doubled (100% increase) their tuning depth over the same time period. These minor learning effects in PPC may reflect that their decoded estimates relied more heavily upon contributions from brain areas other than PPC (e.g., M1, which had stronger decoding power in their sample), possibly resulting in stronger learning in those other areas. In contrast, our data suggest that substantial learning can occur in PPC over the course of just several brain control sessions (e.g., R^2 more than doubled and tuning depth increased by ~70%). Finally, the extent to which plasticity occurs in different brain areas during brain control conditions remains an important direction for future experiments. We expect, based on our findings here and PPC's known functional role in combining visual and motor representations that PPC will be particularly well suited to serve as a target for a prosthesis that relies upon visual feedback for continuous control and error-driven learning.

References

Andersen RA, Buneo CA (2002) Intentional maps in posterior parietal cortex. Annu Rev Neurosci 25:189–220.

Ashe J, Georgopoulos AP (1994) Movement parameters and neural activity in motor cortex and area 5. Cereb Cortex 4:590–600.

Averbeck BB, Chafee MV, Crowe DA, Georgopoulos AP (2005) Parietal representation of hand velocity in a copy task. J Neurophysiol 93:508–518.

Batista AP, Buneo CA, Snyder LH, Andersen RA (1999) Reach plans in eye-centered coordinates. Science 285:257–260.

Branchaud EA, Andersen RA, Burdick JW (2006) An algorithm for autonomous isolation of neurons in extracellular recordings. In: IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob). Pisa, Tuscany, Italy.

Buneo CA, Jarvis MR, Batista AP, Andersen RA (2002) Direct visuomotor transformations for reaching. Nature 416:632–636.

Carmena JM, Lebedev MA, Crist RE, O'Doherty JE, Santucci DM, Dimitrov DF, Patil PG, Henriquez CS, Nicolelis MAL (2003) Learning to control a brain-machine interface for reaching and grasping by primates. PLOS Biol 1:E42.

Cham JG, Branchaud EA, Nenadic Z, Greger B, Andersen RA, Burdick JW (2005) Semi-chronic motorized microdrive and control algorithm for autonomously isolating and maintaining optimal extracellular action potentials. J Neurophysiol 93:570–579.

Efron B, Hastie T, Johnstone I, Tibshirani R (2004) Least angle regression. Ann Stat 32:407–451.

Flanders M, Cordo PJ (1989) Kinesthetic and visual control of a bimanual task—specification of direction and amplitude. J Neurosci 9:447–453.

Gail A, Andersen RA (2006) Neural dynamics in monkey parietal reach region reflect context-specific sensorimotor transformations. J Neurosci 26:9376–9384.

Geman S, Bienenstock E, Doursat R (1992) Neural networks and the bias variance dilemma. Neural Comput 4:1–58.

Hastie T, Tibshirani R, Friedman J (2001) The elements of statistical learning. New York: Springer.

Hochberg LR, Serruya MD, Friehs GM, Mukand JA, Saleh M, Caplan AH, Branner A, Chen D, Penn RD, Donoghue JP (2006) Neuronal ensemble control of prosthetic devices by a human with tetraplegia. Nature 442:164–171.

Hoerl AE, Kennard RW (1970) Ridge regression—biased estimation for nonorthogonal problems. Technometrics 12:55–67.

Jordan MI, Rumelhart DE (1992) Forward models—supervised learning with a distal teacher. Cogn Sci 16:307–354.

Kalman RE (1960) A new approach to linear filtering and prediction problems. Trans ASME J Basic Eng 82:35–45.

Kennedy PR, Bakay RAE, Moore MM, Adams K, Goldwaithe J (2000) Direct control of a computer from the human central nervous system. IEEE Trans Rehabil Eng 8:198–202.

Mulliken GH, Musallam S, Andersen RA (2008) Forward estimation of movement state in posterior parietal cortex. Proc Natl Acad Sci U S A 105:8170–8177.

Musallam S, Corneil BD, Greger B, Scherberger H, Andersen RA (2004) Cognitive control signals for neural prosthetics. Science 305:258–262.

Nenadic Z, Burdick JW (2005) Spike detection using the continuous wavelet transform. IEEE Trans Biomed Eng 52:74–87.

Paninski L, Fellows MR, Hatsopoulos NG, Donoghue JP (2004) Spatiotemporal tuning of motor cortical neurons for hand position and velocity. J Neurophysiol 91:515–532.

Patil PG, Carmena JM, Nicolelis MAL, Turner DA (2004) Ensemble recordings of human subcortical neurons as a source of motor control signals for a brain-machine interface. Neurosurgery 55:27–35; discussion 35–38.

Petersen N, Christensen LOD, Morita H, Sinkjaer T, Nielsen J (1998) Evidence that a transcortical pathway contributes to stretch reflexes in the tibialis anterior muscle in man. J Physiol 512:267–276.

Pistohl T, Ball T, Schulze-Bonhage A, Aertsen A, Mehring C (2008) Prediction of arm movement trajectories from ECoG-recordings in humans. J Neurosci Methods 167:105–114.

Quiroga RQ, Snyder LH, Batista AP, Cui H, Andersen RA (2006) Movement intention is better predicted than attention in the posterior parietal cortex. J Neurosci 26:3615–3620.

Ringach DL, Hawken MJ, Shapley R (1997) Dynamics of orientation tuning in macaque primary visual cortex. Nature 387:281–284.

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

Scherberger H, Jarvis MR, Andersen RA (2005) Cortical local field potential encodes movement intentions in the posterior parietal cortex. Neuron 46:347–354.

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

Shenoy KV, Meeker D, Cao S, Kureshi SA, Pesaran B, Buneo CA, Batista AP, Mitra PP, Burdick JW, Andersen RA (2003) Neural prosthetic control signals from plan activity. Neuroreport 14:591–596.

Snyder LH, Batista AP, Andersen RA (1997) Coding of intention in the posterior parietal cortex. Nature 386:167–170.

Srinivasan L, Eden UT, Willsky AS, Brown EN (2006) A state-space analysis for reconstruction of goal-directed movements using neural signals. Neural Comput 18:2465–2494.

Suner S, Fellows MR, Vargas-Irwin C, Nakata GK, Donoghue JP (2005) Reliability of signals from a chronically implanted, silicon-based electrode array in non-human primate primary motor cortex. IEEE Trans Neural Syst Rehabil Eng 13:524–541.

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

Tibshirani R (1996) Regression shrinkage and selection via the Lasso. J R Stat Soc Ser B Stat Methodol 58:267–288.

Welch G, Bishop G (2006) An introduction to the Kalman filter. Technical Report 95-041. Chapel Hill, NC: University of North Carolina at Chapel Hill.

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

Wolpaw JR, McFarland DJ (2004) Control of a two-dimensional movement signal by a noninvasive brain-computer interface in humans. Proc Natl Acad Sci U S A 101:17849–17854.

Wolpert DM, Miall RC (1996) Forward models for physiological motor control. Neural Netw 9:1265–1279.

Wolpert DM, Ghahramani Z, Jordan MI (1995) An internal model for sensorimotor integration. Science 269:1880–1882.

Wu W, Black MJ, Gao Y, Bienenstock E, Serruya M, Shaikhouni A, Donoghue JP (2003) Neural decoding of cursor motion using a kalman filter. In: Advances in neural information processing systems (Becker S, Thrun S, Obermayer K, eds), Vol 15, pp 133–140. Cambridge, MA: MIT.

Yu BM, Kemere C, Santhanam G, Afshar A, Ryu SI, Meng TH, Sahani M, Shenoy KV (2007) Mixture of trajectory models for neural decoding of goal-directed movements. J Neurophysiol 97:3763–3780.

Zou H, Hastie T (2005) Regularization and variable selection via the elastic net. J R Stat Soc Ser B Stat Methodol 67:301–320.