

Stochastic Optimal Control as a Theory of Brain-Machine Interface Operation

Manuel Lagang

laganojunior@gmail.com

Neural Signal Processing Laboratory, Department of Radiology, University of California Los Angeles, Los Angeles, CA 90095-7437, U.S.A., and Department of Computer Science, California Institute of Technology, Pasadena, CA 91125, U.S.A.

Lakshminarayan Srinivasan

ls2@nsplab.org

Neural Signal Processing Laboratory, Department of Radiology, University of California Los Angeles, Los Angeles, CA 90095-7437, U.S.A.

The closed-loop operation of brain-machine interfaces (BMI) provides a framework to study the mechanisms behind neural control through a restricted output channel, with emerging clinical applications to stroke, degenerative disease, and trauma. Despite significant empirically driven improvements in closed-loop BMI systems, a fundamental, experimentally validated theory of closed-loop BMI operation is lacking. Here we propose a compact model based on stochastic optimal control to describe the brain in skillfully operating canonical decoding algorithms. The model produces goal-directed BMI movements with sensory feedback and intrinsically noisy neural output signals. Various experimentally validated phenomena emerge naturally from this model, including performance deterioration with bin width, compensation of biased decoders, and shifts in tuning curves between arm control and BMI control.

Analysis of the model provides insight into possible mechanisms underlying these behaviors, with testable predictions. Spike binning may erode performance in part from intrinsic control-dependent constraints, regardless of decoding accuracy. In compensating decoder bias, the brain may incur an energetic cost associated with action potential production. Tuning curve shifts, seen after the mastery of a BMI-based skill, may reflect the brain's implementation of a new closed-loop control policy. The direction and magnitude of tuning curve shifts may be altered by decoder structure, ensemble size, and the costs of closed-loop control. Looking forward, the model provides a framework for the design and

Color versions of some figures in this article are presented in the supplemental material available online at http://www.mitpressjournals.org/doi/suppl/10.1162/NECO_a.00394.

simulated testing of an emerging class of BMI algorithms that seek to directly exploit the presence of a human in the loop.**1 Introduction**

The closed-loop operation of brain-machine interfaces (BMI) provides a framework to study the mechanisms behind neural control of behavior through a severely restricted output channel. In the basic prototype, measurements of the subject's neural signals are translated by a BMI algorithm into movements of a computer cursor (or other device) that the subject attempts to control, bypassing the need for overt movement. As such, BMI is a therapeutic possibility for severe paralysis from stroke, trauma, and degenerative disease. Despite significant ongoing improvements in the design of closed-loop BMI systems, a fundamental, empirically validated theory of closed-loop BMI operation is lacking.

Although the BMI concept has its roots in operant conditioning (Fetz, 1969), present-day BMI algorithms have built on open-loop decoding, where natural arm movements are reconstructed from concurrently measured neural activity (Brockwell, Rojas, & Kass, 2004; Eden, Frank, Barbieri, Solo, & Brown, 2004; Georgopoulos, Schwartz, & Kettner, 1986; Wu, Gao, Bienenstock, Donoghue, & Black, 2006). Subsequently, these same algorithms were implemented in closed-loop decoding (Carmena et al., 2003; Serruya, Hatsopoulos, Paninski, Fellows, & Donoghue, 2002; Taylor, Tillery, & Schwartz, 2002). This means that the subject was allowed to view the output of the algorithm in real time, represented in the movement of a cursor or robotic arm. In many demonstrations, subjects have been able to control closed-loop cursor behavior without making overt arm movements.

In various experiments with BMI, closed-loop decoding significantly outperforms open-loop decoding. At least three additional phenomena of closed-loop BMI control are rigorously described in the literature. First, closed-loop control deteriorates with increasing bin width in spike-based BMI (Cunningham et al., 2011). Second, closed-loop decoding is capable of compensating for BMI algorithms that demonstrate systematic bias in open-loop decoding (Chase, Schwartz, & Kass, 2009; Jarosiewicz et al., 2008; Koyama et al., 2010). Third, neural ensembles in motor cortex demonstrate that preferred direction of velocity tuning curves shifts between arm control and closed-loop BMI control (Carmena et al., 2003; Ganguly & Carmena, 2009; Ganguly, Dimitrov, Wallis, & Carmena, 2011; Jarosiewicz et al., 2008; Taylor et al., 2002), as well as between closed-loop operation of different BMI algorithms (Ganguly et al., 2011).

While it is generally believed that feedback to the subject explains dramatic improvements in closed- versus open-loop BMI performance, no comprehensive model has been developed to validate this hypothesis. Such a model would need to generalize beyond closed-loop performance

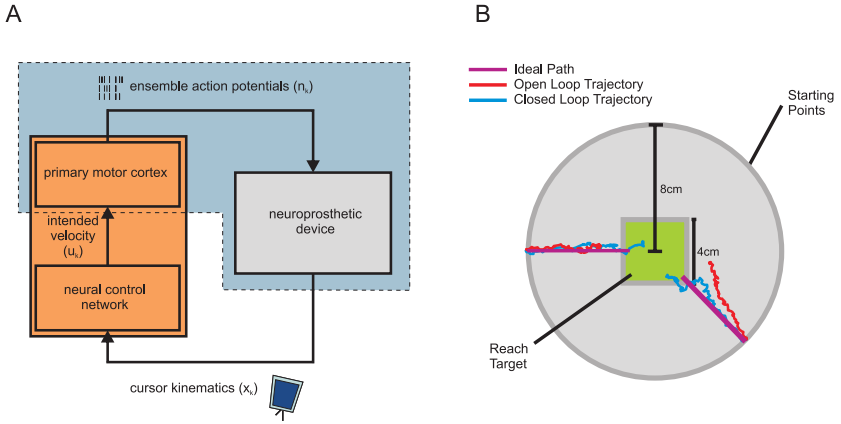


Figure 1

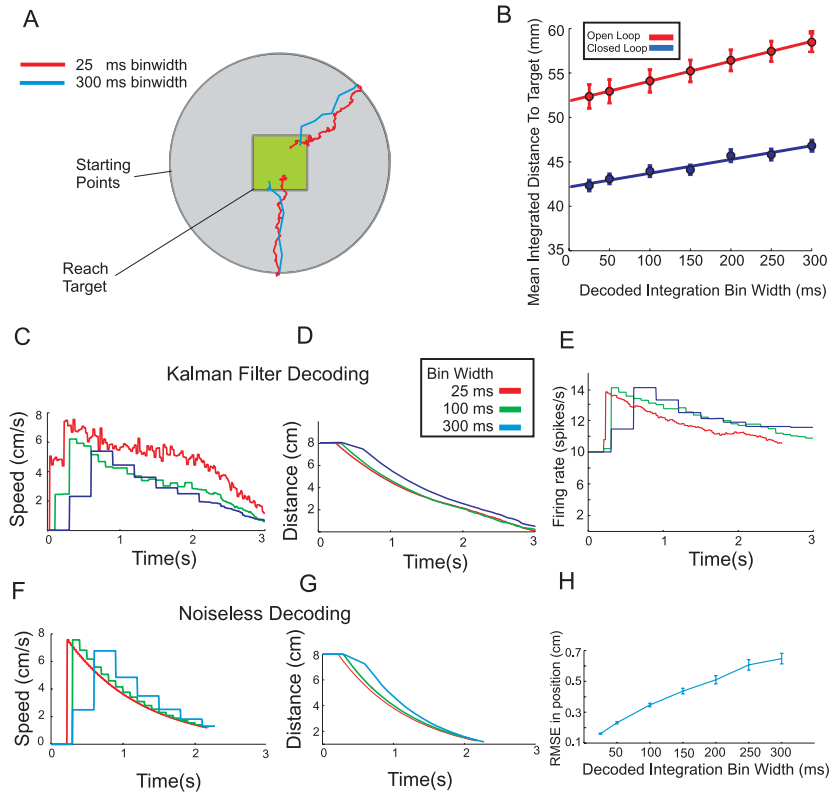


Figure 2

improvements to simultaneously explain various other coexisting phenomena: deterioration with increasing bin width (Cunningham et al., 2011), compensation of decoder bias (Chase et al., 2009; Jarosiewicz et al., 2008; Koyama et al., 2010), and shifting tuning curves (Carmena et al., 2003; Ganguly & Carmena, 2009; Ganguly et al., 2011; Jarosiewicz et al., 2008; Taylor et al., 2002). Here, we propose to explain these various phenomena of BMI operation by describing the brain in closed-loop mode as a stochastic optimal controller. This class of models has been extensively explored in the description of natural motor coordination (Diedrichsen, Shadmehr, & Ivry, 2009; Scott, 2004; Todorov, 2004). In related work, suboptimal control based on iterative parameter optimization was introduced in a previous modeling study of learning in BMI (Heliot, Ganguly, Jimenez, & Carmena, 2010).

In our BMI model framework (see Figure 1A), stochastic optimal control is implemented by a neural control network that attempts to steer neural activity (n_k) in primary motor cortex using a representation of velocity (u_k) with the goal of directing the observed cursor (x_k) to desired targets. In effect, the neural control network steers a composite system (see Figure 1A, enclosed by the dashed outline) that includes primary motor cortex and the neuroprosthetic device. By analogy to the motor control literature, the principle of optimality represents performance when learning is complete rather than describing intermediate performance. The resulting stochastic dynamical systems model demonstrates a richness of behavior sufficient to reproduce differences in closed- versus open-loop BMI performance, while

Figure 1: Stochastic optimal control model for closed-loop BMI operation. (A) The neural control network must compensate the coupled dynamics of noisy output neurons and the BMI algorithm, inside the dashed region. (B) Sample open- and closed-loop trajectories of the model during a goal-directed task with standard work space and target dimensions. Ninety-six neurons were used with a 25 ms bin width and Kalman filter decoding.

Figure 2: Closed-loop performance with varying bin width. (A) Sample trajectories at two bin widths. (B) Mean integrated distance to target (MID) versus bin width for open- and closed-loop modes. (C) Speed profiles at various bin widths using a Kalman filter. (D) Corresponding distance profiles using a Kalman filter. (E) Sample neuronal firing rates using a Kalman filter. (F) Speed profiles with zero neuronal noise. (G) Corresponding distance profiles with zero neuronal noise. (H) Root-mean-squared error (RMSE) in decoded position versus bin width. Each point in panels B and E is a mean of 100 trials, with error bars representing 95% confidence intervals on the estimated mean. All panels use 96 neurons with a 25 ms bin width and Kalman filter decoding.

simultaneously explaining key phenomena of closed-loop BMI operation through a concise mathematical framework.

2 Methods

2.1 Models for the Neuroprosthetic Device. We explored models for the neuroprosthetic device corresponding to three frequently used BMI algorithms: Kalman filter (KF), optimal linear estimator (OLE), and population vector algorithm (PVA). Each neuroprosthetic device model receives spike times from an ensemble of simulated primary motor cortical neurons as input and updates the position and velocity of the simulated cursor as output. Equations for the implementation of these various filters are provided in appendix B. Parameters of the KF, PVA, and OLE were fit using maximum likelihood estimation.

In our analysis, the neuroprosthetic device is a system, defined by a state and dynamics. The state x_k is a 5×1 vector containing cursor position, velocity, and the number 1 (used to implement affine dynamics). The dynamics are governed by the corresponding BMI filter equations, which specify the future state x_{k+1} in terms of the previous state x_k and neural signals n_k . The specific mathematical form of these dynamics can simplify or complicate the derivation of a neural control network model. In particular, we desire linear dynamics governed by parameters that are invariant with time (called LTI systems).

For the OLE and PVA, the corresponding neuroprosthetic device is LTI. For the KF, parameters change with the Kalman gain. As a result, the standard KF is linear but not time invariant. Under typical conditions, the Kalman gain has an asymptotic value that is independent of the neural signal input. By initializing Kalman gain to this steady-state value, the KF-based device becomes LTI. This variation on the KF is called the steady-state KF, which substantially simplifies derivation of the neural control network model. As an implementation detail on the KF case, note that the neural control network assumes the KF-based neuroprosthetic device is a steady-state KF, while the simulated device itself executes the standard KF.

2.2 Model for Primary Motor Cortex. Our model for primary motor cortex is cosine-tuned velocity-dependent firing (Moran & Schwartz, 1999; Truccolo, Eden, Fellows, Donoghue, & Brown, 2005). The specific form is obtained from a recent experimental study relating closed-loop performance and bin width (Cunningham et al., 2011). Formally, let b_i be the preferred vector and c_i be the minimum firing rate of neuron i . The mean firing rate of this neuron is then

$$\lambda = b_i \cdot v_k + c_i. \quad (2.1)$$

The dot in this equation represents an inner product. The number of spikes recorded at the i th bin of width Δ seconds is distributed as

$$n_i \sim \text{Poisson}(\max(\lambda * \Delta, 0)). \quad (2.2)$$

In our model, we choose $c_i = 10$ spikes/s and $b_i = (\cos(\theta), \sin(\theta)) \times \frac{14 \text{ spikes/s}}{20 \text{ cm/s}}$ with θ uniformly drawn on $[0, 2\pi]$. This resulted in uniformly random preferred directions, minimum firing rate of 10 spikes/s, and a firing rate of 24 spikes/s for intended movements at 20 cm/s along the preferred direction, as used previously (Srinivasan, Eden, Willsky, & Brown, 2006). As a technical aside, substantially higher firing rates have been used in conjunction with this model in a three-dimensional simulated environment (Cunningham et al., 2011). While our preliminary work initially employed those higher firing rates, we found decoding accuracy to be unrealistically high in our two-dimensional work space.

2.3 Model for Neural Control Network. Our next challenge was to represent the brain in closed-loop operation of the neuroprosthetic device. Experimental results have suggested that feedback may explain the diversity of phenomena in closed-loop BMI operation. The stochastic optimal control framework was a natural choice to examine this specific hypothesis. This controller incorporates feedback on the neuroprosthetic cursor position and velocity (x_k) to steer neural activity recorded from motor cortex by determining the intended cursor velocity (u_k) using a rule (or function) called a control policy. The control policy embodies the input-output relationship of the neural control network (see the system diagram in Figure 1). Equations for the derivation and implementation of this control policy are provided in appendix C. Below, we provide a brief introduction to the neural control network with a focus on the high-level concepts.

The first major component that shapes the control policy implemented by the neural control network is the cost function. The cost function penalizes both cursor kinematics (x_k) and intended velocity (u_k) to encourage spike patterns that result in goal-directed cursor movement that stops at the origin:

$$E_{w_k} \left\{ \sum_{k=0}^{\infty} x_k^T Q x_k + u_k^T R u_k \right\}. \quad (2.3)$$

Q and R are square matrices that determine the relative costs associated with state and control, respectively. Choice of Q results in reaching movements. Choice of R produces realistic neuronal firing rates. In our specific example, x_k is a 5×1 vector that contains position and velocity in two dimensions at

time step k , as well as the constant 1 in the fifth row, to allow for affine control policies. Also, u_k is a 2×1 vector containing the intended 2D velocity at time step k . We employed basic choices of Q and R to match these dimensions:

$$Q = \begin{pmatrix} \alpha & 0 & 0 & 0 & 0 \\ 0 & \alpha & 0 & 0 & 0 \\ 0 & 0 & \beta & 0 & 0 \\ 0 & 0 & 0 & \beta & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad (2.4)$$

$$R = \begin{pmatrix} \gamma & 0 \\ 0 & \gamma \end{pmatrix} \quad (2.5)$$

where α , β , and γ are scalars. The relative weighting of these two costs (Q and R) was tuned by hand in order to achieve realistic reaching movements and firing rates. When the R was too low, neural firing rates were unrealistically high. When Q was too low, realistic reaching movements were not produced. Nevertheless, model behavior was preserved under a wide range of parameters, as described in appendix E.

The second major component that shapes the control policy implemented by the neural control network is the plant, a term from control theory for the system that the control network is attempting to steer. Importantly, this system is not just the neuroprosthetic device but the combined effect of the output neuronal layer and the neuroprosthetic device itself. Both the output neuronal layer and neuroprosthetic device models are described in the previous two sections. Although the plant represented by the combined effect of the output neuronal layer and neuroprosthetic device is not strictly linear, gaussian, or in all cases time invariant, our neural control network makes these simplifying assumptions in determining the control policy. In other words, the neural control network acts as if it is controlling a plant with standard linear time-invariant dynamics and gaussian additive noise (w_k):

$$x_{k+1} = Ax_k + Bu_k + w_k. \quad (2.6)$$

Here, A and B represent the entire plant, including the combined effect of output neurons and the neuroprosthetic device. For our example, A is 5×5 (to match the dimensions of x_k), and B is 5×2 (because u_k is 2×1). This formulation greatly simplifies the model implementation and analysis, bringing it into the realm of established theory on linear quadratic regulators (Bertsekas, 2005).

Conveniently, the control policy u_k that optimizes this cost turns out to be a linear, time-invariant function of the state x_k (Bertsekas, 2005):

$$u_k^*(x) = Lx_k. \quad (2.7)$$

In the full closed-loop simulation, this control policy, equation 2.7, is coupled with the actual nongaussian neural signal model and true (not necessarily time-invariant) neuroprosthetic device equations (described in the previous two sections) to simulate closed-loop BMI-controlled cursor movements toward a target in a two-dimensional work space.

Four caveats and implementation details are helpful regarding this abbreviated description of the neural control network, more fully described in the appendix. First, note that the use of optimal control results in a neural control network model that represents performance when learning is complete. In fact, our controllers are only approximately optimal, because our simulated primary motor cortical neurons emit point-process signals that the controller approximates as gaussian. We also briefly examine the effect of exact gaussian neural activity in Figure 3B (which more reasonably approximates local and distant field potentials rather than spikes), where our controller is exactly optimal.

Second, while feedback to the user in standard experiments is available in time steps determined by the screen refresh rate, the various BMI algorithms (KF, OLE, PVA) update at intervals determined by the bin width, which is substantially longer. To capture this constraint in the model, the controller was afforded the capability to project state dynamics over 5 ms steps, which was at a finer resolution than the smallest decoding bin width of 25 ms. While allowing for a more realistic model, accounting for feedback at finer temporal scales than the decoding bin width did not change the stated trends. The appendix describes our solution to this problem in full.

Third, our model also includes a nonzero reaction time of 200 ms. Although the choice of reaction time had no bearing on the major trends, a nonzero reaction time was needed to match the general magnitude of mean integrated distance to target (MID) seen in experimental data. As would be expected from the definition of MID, finite reaction time increases the minimum achievable MID for any condition, which is the y -intercept of the MID versus bin width trend.

Fourth, although our model does not assert a particular anatomical location for the neural control network, experimental data have demonstrated changes in patterns of activity from neurons that are not directly connected to recording electrodes (Ganguly et al., 2011). In fact, the few neurons involved in directly modulating the neuroprosthetic device are connected indirectly to a network of millions of neurons that could subserve this role. Our neural control network model implements feedback control with a level of abstraction that allows clarity in analysis while sacrificing the cellular resolution provided by a large-scale biophysical model.

2.4 Parameter Selection and Sensitivity. Parameters of our model were held constant across the various analyses rather than being individually optimized for each separate analysis. Parameters of the neuroprosthetic device and primary motor cortex were fully determined by the neural signal model, which in turn is determined from the literature (Cunningham et al., 2011; Moran & Schwartz, 1999; Srinivasan et al., 2006; Truccolo et al., 2005). Remaining parameters included (α, β, γ) within the cost function of the neural control network and a deterministic reaction time parameter. Equations 2.3 to 2.5 define the cost function in terms of these parameters. We fixed $\beta = \gamma, \alpha/\beta = 1.8$, and reaction time = 200 ms for simplicity, reducing our model to a total of 1 free parameter, namely, γ/α .

The α and β parameters determine the importance of achieving the target location at zero velocity. The γ parameter indirectly determines the importance of energy dissipated from spiking activity by penalizing intended velocity (u_k). Balance between these parameters is needed to produce both goal-directed reaching movements and realistic firing rates. As described in appendix E, the trends reported in this study hold over four orders of magnitude in γ/α .

The deterministic reaction time parameter modulated the y -intercept offset of MID score in Figure 2B. We found it necessary to introduce reaction time (set to 200 ms in this analysis) to match the magnitude of MID values reported from experimental data. However, the choice of this parameter had no bearing on the relationship of closed-loop performance with bin width, the relevant trend in this analysis.

2.5 Reach Task. As a basis for comparison against experimental data, we employed a standard center-out reach task reported in a recent study relating closed-loop performance and bin width (Cunningham et al., 2011). Rather than initiating center-out movements, we equivalently required out-to-center movements for mathematical simplicity, without loss of generality. Starting points were drawn uniformly on a circle of radius 8 cm. The target zone was a square measuring 4 cm in width, centered on the origin. Successful target acquisition required a hold period, where the cursor was to be held within the target zone for at least 500 ms. Each trial was terminated after target acquisition, or at 3 seconds, whichever came first. These conditions were identical to the corresponding experimental setup (Cunningham et al., 2011) except that our work space was 2D rather than 3D.

3 Results

We studied the behavior of our model for closed-loop BMI operation using 96 simulated neurons in a target acquisition task designed to reproduce standard testing conditions (Cunningham et al., 2011). Details of the model implementation and task were provided in section 2. Here, we report the results of our analysis.

3.1 Open-Loop and Closed-Loop Performance Versus Bin Width. As a preliminary analysis, we compared single trials of open-loop versus closed-loop performance in the reach task using a Kalman filter (KF) model of the neuroprosthetic device (see section 2). Figure 1B demonstrates sample paths for two trials. Cursor movements begin on the circular periphery in an attempt to approach the origin within the target zone. Open-loop trajectories demonstrate a meandering path. In comparison, using the same population of neurons, closed-loop control affects more directed paths that acquire the target.

We then used mean integrated distance to target (MID) to quantify open- and closed-loop performance (Cunningham et al., 2011). MID is computed by averaging the sequence of distances measured from the cursor to the target over the duration of the trial. All trials were incorporated into the MID calculation, including those with unsuccessful target acquisition. In concordance with experimental data, our model demonstrates that closed-loop mode outperforms open-loop mode (see Figure 2B).

Subsequently, we systematically varied bin width to examine its effect on closed-loop performance in the reach task (see Figure 2A). Our results show deterioration in closed-loop MID with increasing bin width (see Figure 2B), as well increasing time to target (see Figure 3A) and target acquisition failure rate (see Figure 3B). These trends concur with the experimental data (Cunningham et al., 2011). There is also a similar trend in the open-loop performance, although first-order trends in the experimental data on open-loop performance versus bin width are ambiguous (Cunningham et al., 2011). Moreover, we were unable to reproduce the strongly quadratic-appearing trends demonstrated in some examples of open-loop MID versus bin width (Cunningham et al., 2011).

To understand why closed-loop performance depends on bin width, we examined two possible explanations, motivated by inspection of the sample trajectories (see Figure 2A). First, this dependence might represent an intrinsic property of closed-loop control, even in the absence of neural signal noise. Surprisingly, the zero-noise condition also exhibited a linear trend in MID versus bin width, almost identical to the original model (see Figure 3C), although with significantly decreased time to target (see Figure 3D). In plots of the cursor speed (see Figure 2C) with noise and (see Figure 2F) without noise, as well as distance to target versus time (see Figure 2D) with noise and (see Figure 2G) without noise, smaller bin widths allow faster initial acceleration in cursor velocity and smoother deceleration, resulting in a faster approach to target and lower MID. Speed and distance profiles reflect a 200 ms deterministic reaction time (see section 2). Note that neurons are permitted to transition instantly between various firing rates (see Figure 2E).

Second, closed-loop performance deterioration with increasing bin width may also reflect the integration of velocity decoding errors over longer bin widths. A countervailing force in this explanation is that larger

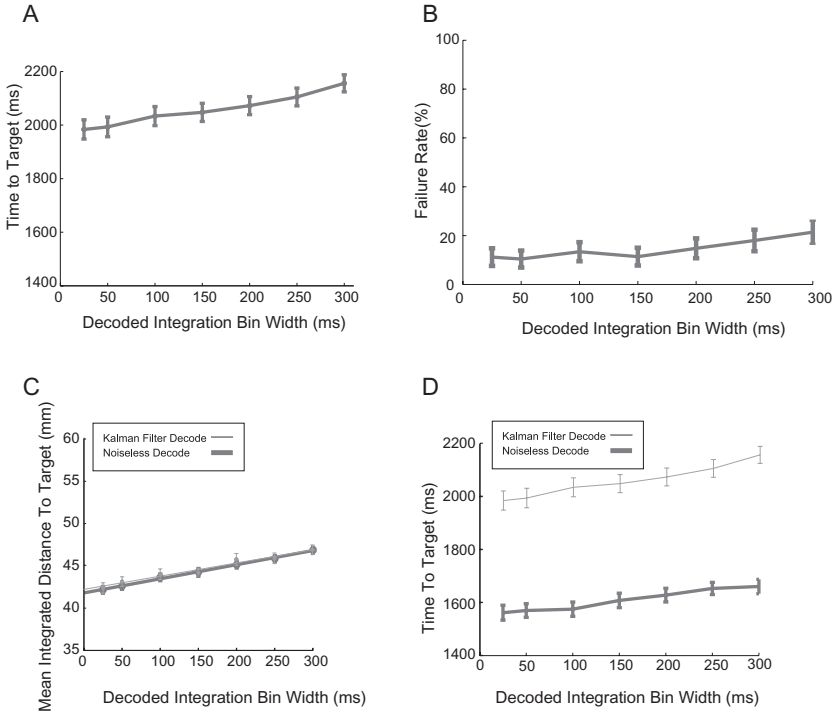


Figure 3: Further analysis of closed-loop BMI control. (A) Time to target versus bin width for successful trials in closed-loop operation of the Kalman filter (KF) decoder. (B) Failure rate versus bin width for closed-loop operation using the KF. (C) Comparison of mean integrated distance to target using the KF and a noiseless decoder. (D) Comparison of time to target using the KF and a noiseless decoder. Simulation conditions are identical to Figure 2.

bin widths might better conform to assumptions by the KF that spike counts are gaussian. To determine which effect dominated in this model, we performed open-loop decodes over the duration of the bin width, calculating root mean squared error in the final position at the end of the bin width. The resulting trend (see Figure 2H) demonstrates that longer bin width results in larger errors in cursor position at the end of the bin width period.

As a technical point, note that the cursor exhibits movement prior to the 200 ms reaction time (see Figure 2C). This is because before the reaction time, spiking at the baseline firing rate is decoded by the filter into a nonzero velocity. Moreover, this prereaction-time decoded velocity exceeds velocities achieved at the end of the trial. This occurs because at the end of the trial, the controller actively compensates to achieve zero velocity as the

cursor gets closer to the goal, whereas this capability is not engaged prior to the reaction time. Nevertheless, high velocities in the first 200 ms reaction period do not manifest in the cursor position trajectory (see Figure 2D) because those velocities are undirected on average.

3.2 Decoder Bias in Open Versus Closed-Loop Behavior. Experiments with BMI have also suggested that closed-loop control can partially compensate decoder bias, which is systematic error in the decoded velocity (Chase et al., 2009; Jarosiewicz et al., 2008; Koyama et al., 2010). To examine this possibility, we compared open-loop versus closed-loop behavior in the PVA (Georgopoulos et al., 1986), which is known to be a biased variation on the OLE (Chase et al., 2009). Error in the decoded angle of movement was averaged over multiple trials for each of eight movement directions. As a matter of convenience, this was achieved by averaging error in the first time step across 1000 trials, separately for each starting point. A plot of decoding bias versus movement direction (see Figure 4A) demonstrates that PVA biases were reproducibly reduced in closed-loop mode relative to open-loop mode. The OLE in contrast, demonstrated no significant change in decoder bias, which was lower than the PVA in either case.

We sought an explanation for how closed-loop control reduces decoder bias. The mathematical analysis (see appendix C) reveals that the controller is explicitly capable of adjusting intended velocities to compensate for decoder bias. However, the analysis also predicts that residual bias can persist in closed-loop mode due to two factors. First, nongaussian neural activity violates assumptions made by the controller, so that even the OLE demonstrates decoder bias (see Figure 4A). Second, cost associated with intended velocity (and consequently motor cortical neuron firing rates) should prevent the neural control network from perfectly compensating PVA bias.

To validate the predictions on persistence of bias made by this analysis, we reexamined decoder bias with neural activity described as a gaussian process rather than an inhomogeneous Poisson process, using the identical mean and variance in spike counts from the original model. Under these conditions (see Figure 4B), decoder bias in the OLE disappeared. In contrast, bias in the closed-loop PVA persisted, presumably reflecting the energetic cost of intended velocities needed to compensate decoder bias. To confirm this intuition, we removed cost associated with intended velocity under gaussian neural signals. This allowed the neural control network to bring closed-loop PVA bias to zero (see Figure 4C).

3.3 Tuning Curves in Arm Versus BMI Control. Neural ensembles in motor cortex demonstrate velocity tuning curves that shift preferred direction between arm control and closed-loop BMI control (Carmena et al., 2003; Ganguly & Carmena, 2009; Ganguly et al., 2011; Jarosiewicz et al., 2008;

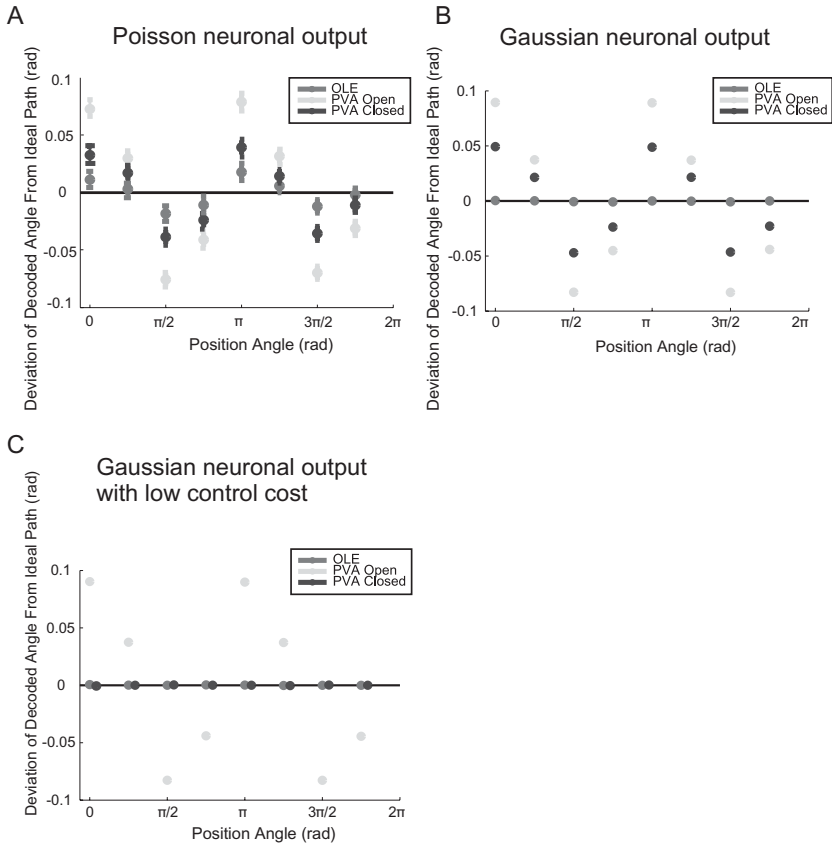


Figure 4: Closed-loop compensation of decoder bias. (A) Deviation from intended movement direction in open- and closed-loop modes for the optimal linear estimator (OLE) and population vector algorithm (PVA). The OLE points represent identical bias for both modes. (B) Deviation from intended movement direction with gaussian neuronal output. (C) Same as in panel B, but with cost on intended velocity set to $\gamma = 10^{-6}$. Each point is generated as the circular mean of 50,000 sampled angles with 95% confidence intervals based on the von Mises distribution as detailed in Berens (2009). Error bars are not visible in panels B and C because they are smaller than the plotted point. All panels use 96 neurons with a 25 ms bin width. (A color version of this figure is available in the supplemental material.)

Taylor et al., 2002) as well as between closed-loop operation of different BMI algorithms (Ganguly et al., 2011). These studies have suggested that tuning curve shifts represent skill learning through neuronal adaptation to the BMI algorithm, although systematic shifts in particular directions have not been consistently demonstrated across publications. To understand precisely why tuning curve shifts manifest during proficient BMI control, we compared tuning curves during stochastic optimal PVA and OLE control with the original motor cortical tuning curves, which reflect arm control. Directional tuning curves during BMI control were determined by regressing externally observable intended cursor direction (given by target direction relative to the current cursor position) against spiking activity with the cosine-tuned model form used in Cunningham et al. (2011). It is important to note that externally observable intended cursor direction may be different from the direction of u_k , the internal representation of intended velocity.

We studied preferred direction under BMI and arm control in polar coordinates under various conditions (see Figure 5). These plots represent typical tuning curve shifts, with a progression from large shifts (see Figure 5A) to imperceptible shifts (see Figure 5D). Our analysis reveals that shifts reflect a control policy within the brain that redirects neuronal activity to compensate decoder bias. As a result, decoders with progressively less bias demonstrate decreased tuning curve shift. For example, the PVA with 96 neurons (see Figure 5C) demonstrates less shift than the PVA with 10 neurons (see Figure 5B). Similarly, the OLE with 96 neurons (see Figure 5D) demonstrates less shift than the PVA with 96 neurons (see Figure 5C).

We also see that costs on control influence the magnitude of tuning curve shifts. Tuning curves without control costs (see Figure 5A) demonstrate larger shifts than equivalent tuning curves with control costs in effect (see Figure 5B). To understand this, recall (see Figure 4) that costs on control prevent the brain from entirely compensating decoder bias. When the brain only partially compensates decoder bias, its control strategy under arm and BMI control are more similar, resulting in smaller tuning curve shifts.

Finally, the direction of tuning curve shifts is determined by the inverse of the BMI function, which maps cursor movement onto firing rate. In the process of exerting closed-loop control, the brain's tuning curves shift toward the dominant eigenvector of this mapping (see Figures 5A and 5B). When there is no dominant eigenvector (such as the PVA constructed under large numbers of neurons), the direction of shift is less systematic. In this case, tuning curves are pulled toward a multiplicity of eigenvectors (see Figure 4D). Implications of these results are described below.

4 Discussion

To operate a BMI, the brain must solve the control of a system that differs from the arm in natural movement. This problem is made more challenging

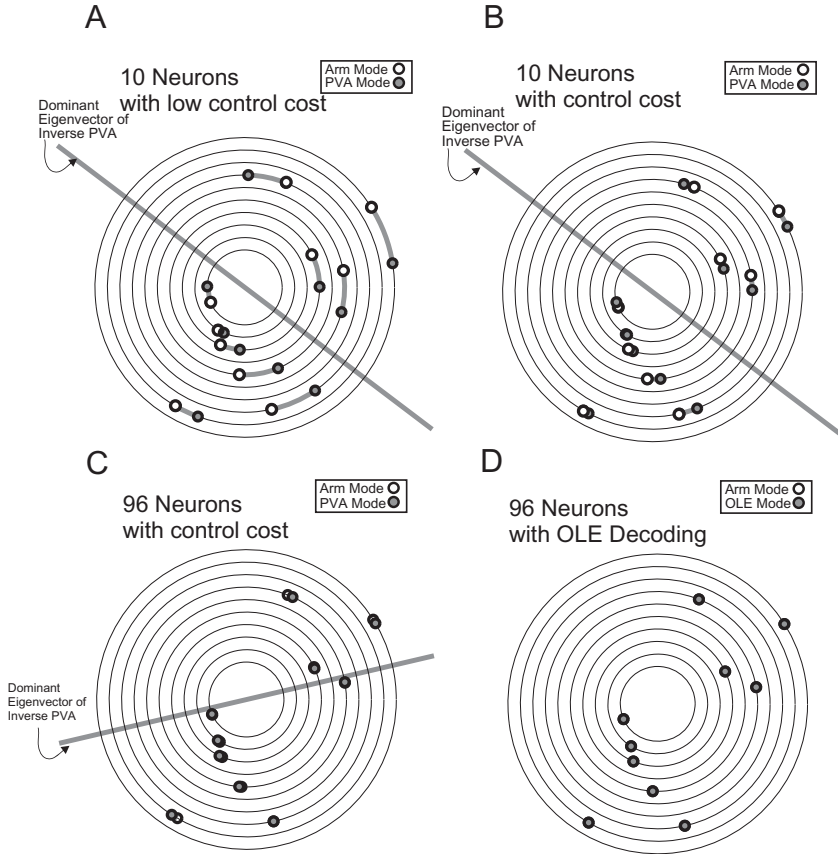


Figure 5: Shifting motor cortical tuning curves in various operation modes. The same subset of neurons is displayed across panels A–D for direct comparison. (A) In an ensemble of 10 neurons with no control cost, preferred direction, switching from arm to PVA control, tends toward the line representing the dominant eigenvector of the inverse PVA mapping. (B) In an ensemble of the same 10 neurons with positive control cost, closed-loop bias compensation of the PVA mapping is incomplete, resulting in smaller shifts. (C) In an ensemble of 96 neurons with the PVA mapping, plotting a subset of 10 neurons identical to those in panels A and B under arm mode, no single eigenvector in the inverse mapping is dominant, resulting in less systematic tuning curve shifts. (D) The magnitude of shift is diminished entirely under OLE, using the same 96 neurons from panels C. In panels A–D, arm control preferred directions were calculated directly from true neural parameters. BMI control preferred directions for each neuron were calculated from 10,000 time points, based on tuning curves that mapped intended direction (from the current cursor position towards the target) onto observed firing rate. (A color version of this figure is available in the supplemental material.)

by a highly constrained channel for control, embodied by a few tens of output neurons rather than the millions that form the corticospinal tract in health. BMI movements represent the combined effect of a neural controller and the BMI algorithm. Understanding neuroprosthetic behavior requires a coherent theory of these two systems and their interaction.

In this analysis, we showed that stochastic optimal control provides a compact description of closed-loop BMI. Control theory is a natural choice to describe the brain in its capacity to adjust neural output in response to sensory feedback. The principle of optimality represents proficiency, demonstrating performance after learning. The stochastic component of this theory incorporates the intrinsic randomness present in neural output. Our model is challenged to work within the constraints of a real BMI system: a noisy neural output channel, biased decoders, and discrete bin width decoding algorithms. The resulting theory for closed-loop BMI posits explanations for why performance deteriorates with increasing bin width, how decoder bias diminishes under closed loop control, and why tuning curves differ between arm movement versus BMI operation. Below, we outline various testable predictions, limitations of the model, use of the model as a simulation platform for BMI algorithm development, and implications for BMI algorithm design.

4.1 Testable Predictions of BMI Based on the Stochastic Optimal Control Model. Three testable predictions of closed-loop BMI operation emerge from the stochastic control model.

First, our model predicts that even BMI controlled with hundreds or thousands of neurons would suffer in performance with increasing bin width. This follows from our analysis that the dependence of closed-loop performance on bin width, studied rigorously in previous work (Cunningham et al., 2011), persists even when neural signal noise is completely removed. By examining speed and position versus time, we demonstrated that smaller bin widths allow the controller to achieve faster, more transient velocities, even in the absence of neuronal noise. Although smaller bin widths favored the Kalman filter in our testing regime, the degree of model mismatch may be a countervailing or positively reinforcing factor in performance with small bin width in any specific decoder. For example, prior work suggests that point-process models may actually decrease model mismatch in single-millisecond-scale bin widths versus a gaussian counterpart (Eden et al., 2004).

Second, although the brain compensates biased decoders in closed-loop experiments (Chase et al., 2009; Jarosiewicz et al., 2008; Koyama et al., 2010), the model predicts that this capability does not necessarily come for free (see Figure 4). In our example with nongaussian neural activity, discrepancies between the PVA and OLE in terms of bias in decoded movement direction are decreased in closed-loop mode in comparison with open-loop

mode. Nevertheless, PVA bias compensation is not complete, even in the case of pure gaussian neural observations where the OLE has no residual direction bias. PVA bias is fully compensated by entirely removing cost associated with intended velocity. This suggests that in practice, fully compensating PVA bias results in increased cost associated with intended velocity. In the full spike-based closed-loop BMI, this cost corresponds to dissipating chemical energy in the production of action potentials at the motor cortical output layer. While the brain in closed loop may be capable of mitigating certain decoder deficiencies, the energetic demand of this compensation to the patient may be a necessary design consideration. Comparison of total spike output in fully trained BMI control with the OLE versus biased PVA is an initial test of this prediction that could be performed retrospectively.

Third, our model predicts decoder-dependent tuning curve shift. This means that tuning curve shifts are expected to depend on the extent of decoder bias in reconstructing intended movement directions (see Figure 5C versus 5D). Several experimental reports have demonstrated that tuning curves can shift between arm movement and BMI control (Carmena et al., 2003; Ganguly & Carmena, 2009; Ganguly et al., 2011; Jarosiewicz et al., 2008; Taylor et al., 2002), as well as between different BMI. Our analysis suggests that changes in output neuron tuning curves in the learned state reflect the closed-loop control policy implemented by the brain. In other words, this shift could be a necessary consequence of proficient BMI control that involves different plant dynamics from the arm and varies across different BMI. Decoder-dependent shift is a testable consequence of this conclusion.

As a specific example of decoder-dependent tuning curve shift, the model suggests that tuning curve shifts are expected to differ between PVA algorithms that use differing numbers of neurons. Our eigenvector analysis demonstrates that PVA tuning curves shift toward the dominant eigenvector of the inverse PVA mapping (see Figures 5A–5C). When more neurons with randomly chosen preferred directions are introduced into the PVA mapping (see Figure 5C), the dominant eigenvector is less influential, resulting in smaller shifts. In other words, if preferred directions are more uniformly distributed, no single eigenvector will dominate, resulting in smaller, more nonsystematic rotations in preferred direction.

As another example of decoder-dependent tuning curve shift, the model suggests that tuning curve shifts will differ between the PVA and OLE methods (see Figure 5). This difference could be explained by the fact that the OLE represents a nearly unbiased estimator, unlike the PVA with a limited number of neurons (see Figure 4). In an experimental validation of this prediction, we expect that the OLE will elicit more significant tuning curve shifts than illustrated here, because real neurons will violate the idealized neural signal model (see equation 2.1), introducing decoder bias into the OLE.

4.2 Limitations of the Stochastic Control Model in Describing BMI.

Here, we examine various limitations of stochastic control in describing closed-loop BMI operation.

Is the behavior of this model invariant to the choice of cost function? Our parameter sensitivity analysis demonstrates that experimental trends reproduced in Figures 1, 2, 4, and 5 are robust across multiple orders of magnitude in parameters of our quadratic cost function. This cost function shapes the behavior of our system by penalizing for action potential production, distance from cursor to target, and cursor velocity. These three components allow realistic firing rates, goal-directed movement, and cursor movements that stop at the target. However, our analysis does not assert uniqueness: other cost functions may result in similar behavior. Nevertheless, this model provides a starting point for subsequent work to investigate which costs are most important to the biological system in operating a BMI. This future direction will require additional technical development related to adaptive estimation of model parameters from empirical data, as well as model quality measures. Moreover, it is not immediately obvious that measurable experiment data will be sufficient to discriminate between meaningfully different cost functions; this problem may be ill posed, although this statement is speculative.

Although our model reproduces a linear closed-loop MID versus bin width trend reported previously (Cunningham et al., 2011), it does not reproduce the associated quadratic open-loop trend. Does this open-loop discrepancy reflect differences between data statistics in these two studies? The difference in the open-loop decoding curve can arise from one (or both) of two sources: the neural signal model and the class of movement trajectories that are open-loop decoded. Our present work and Cunningham et al. (2011) employ identical neural signal models. As a result, the source of this discrepancy can be attributed to differences in the open-loop trajectories. Trajectories reconstructed in open loop by the Cunningham model are three-dimensional arm movements of a human subject, which may have resulted in more complex paths than the simple, directed, two-dimensional point-to-point movements used in our open-loop decoding procedure. This would explain the onset of open-loop performance degradation in Cunningham et al. (2011) with smaller bin widths, which results in a quadratic trend.

This difference in open-loop MID curves does not propagate into differences in closed-loop MID curves because these curves are calculated differently, although they share the same neural signal model. The open-loop decoder operates on neural activity from a reach state equation, while the closed-loop decoder operates on neural activity from the simulated neural control network. Because the closed-loop and open-loop MID curves are calculated differently, it follows that one MID curve might reproduce (Cunningham et al., 2011), while the other would not. Our MID results suggest that the neural control network is a reasonable approximation to the closed-loop strategy of BMI subjects in Cunningham et al. (2011), while the reach

state equation may be overly simple relative to the trajectories employed by Cunningham et al. (2011) in open-loop decoding.

Our model allows discontinuous changes in neuronal firing, resulting in unrealistic step-like profiles in firing rate (see Figure 2E). Would an additional cost on changes in neural output, aimed at smoothing these firing rate profiles, significantly alter the core behaviors of our closed-loop model? Although a comprehensive analysis is needed to definitively address this question, a preliminary examination suggests this might not be the case. In Figure 2, we showed that a linear increase in MID versus bin width resulted from the propagation of velocity decoding errors over longer time periods (see Figure 2H), as well as coarsened speed profiles even in the absence of decoding error (see Figure 2F). The additional cost of changes in neuronal output during closed-loop operation would have no bearing on Figure 2H, which represents a one-time-step open-loop decoding property. The additional cost would, however, result in more gradually rising speed profiles (see Figure 2F). Because of the open-loop decoding effect (see Figure 2H), the relative differences between various bin widths would likely persist with the dampened rise in speed. The combined result of dampened speed profiles and the open-loop decoding effect might actually potentiate the MID versus bin width trend. On a more speculative note, it may be the case that gradual changes in neuronal firing rates represent an emergent property of closed-loop behavior in specific experimental paradigms rather than intrinsic limitations of neural activity, which can demonstrate rapid firing rate changes relative to 25 to 300 ms bins, for example, in transition from stasis to movement (Richardson, Borghi, & Bizzi, 2012).

Although the optimality principle represents proficient BMI operation, the model does not explain the process and dynamics of learning to control the BMI. In subsequent work, connections between our framework and the adaptive control perspective with iteratively tuned parameters (Heliot et al., 2010) will provide a clearer understanding of the dynamics of BMI in learning. Additionally, our controller is provably optimal for gaussian observation processes (such as field potentials) but only adequate (and not proven optimal) for the inhomogeneous Poisson process model of motor cortical spiking. Related to this point, alternate control strategies may be needed to analyze point-process filters (Eden et al., 2004) and adaptive filters (Dangi, Gowda, Heliot, & Carmena, 2011; Eden et al., 2004) that adjust online to changing neural parameters. These filters are time varying (and in some cases, nonlinear), where our neural control network was constructed for linear and time-invariant cases: the steady-state KF, OLE, and PVA. Uncertainty in neural representations of sensory feedback is another unexplored theme in the theoretical analysis of BMI, where new hardware techniques are emerging for the direct patterning of neural activity (Wang et al., 2012).

Could our model explain differential changes in preferred direction among perturbed and nonperturbed units as described in Jarosiewicz et al.

(2008)? It is unlikely that the model in its current form, with the neural control network as a single monolithic feedback controller, could accommodate this more complex behavior. Reproducing this additional data might require invoking the brain's capability for implementing multiple processes that separately control subpopulations of output motor neurons, although this statement is speculative. In future work, our model formulation could be explicitly refined against this additional experimentally documented behavior, such as by comparison to the iterative parameter tuning model for BMI learning reported previously (Heliot et al., 2010), that appears to have been successful in this regard. Although our analysis predicts that shifting occurs to a variable extent across neurons based on distance between the preferred direction and inverse mapping eigenvector (see Figure 4), this emergent phenomenon of the model may be unrelated to the differential changes in preferred direction documented in Jarosiewicz et al. (2008) between perturbed and nonperturbed neurons. This question is unresolved in this letter and requires additional examination. Additional perspective on shifting tuning curves may be obtained by empirically documenting the effects of metabolic and attentional states on neuronal activity at the output layer.

4.3 Simulation-Based BMI Development with the Stochastic Control Model. The stochastic control model may assist in future BMI algorithm development. Simulation of closed-loop performance is a beneficial complement to the experimental pipeline for BMI validation because it offers a fast path for testing and debugging. Simulation is similarly used in other fields where empirical validation is costly, including aeronautics, bridge construction, and communication systems. To date, there has been limited use of simulation in BMI development, such as with open-loop decoding using point-process neural signal models (Kemere & Meng, 2005; Shanechi, Wornell, Williams, & Brown, 2010; Srinivasan et al., 2006). This may be in part because open-loop simulation does not embody the human-in-the-loop aspect of closed-loop BMI operation. Recent work demonstrated that closed-loop operation can be simulated by tracking overt arm movements of a healthy human in the loop (Cunningham et al., 2011), as well as by iteratively tuned control through gradient descent (Heliot et al., 2010). Our work expands the path toward computer-based simulation of proficient closed-loop control, which may complement the use of iteratively tuned control to simulate the BMI learning process. The next phase of development in BMI algorithms is beginning to consider closed-loop effects (Dangi et al., 2011; Gage, Ludwig, Otto, Ionides, & Kipke, 2005; Gilja et al., 2010; Golub, Yu, Schwartz, & Chase, 2011; Jarzebowski, Srinivasan, & Coleman, 2008; Ma, Aghasadeghi, Jarzebowski, Bretl, & Coleman, 2012; Orsborn, Dangi, Moorman, & Carmena, 2012). In select cases, the possibility of fast, noniterative closed-loop simulation demonstrated by our modeling approach may provide an efficient choice for preclinical validation. An important limitation

of this letter in this regard is that we have not specified a generic method to compute an approximately optimal (or at least, proficient) control policy for every possible neuroprosthetic device, for which additional work is needed. A solution may involve an efficient adaptation of the iterative-tuning perspective (Heliot et al., 2010).

Some researchers have suggested in passing that open-loop decoding may be a poor surrogate for closed-loop performance in determining the relative importance of various decoding strategies (Cunningham et al., 2011). Our results temper this notion, demonstrating that poor open-loop performance directly degrades closed-loop performance (see Figure 2H). Moreover, we show that compensating for biased decoders is not complete or free of cost (see Figure 4). This view is more in line with previous work, which concludes that some advantages of a given BMI algorithm seen in open-loop mode (although not all) are expected to persist in closed-loop mode (Chase et al., 2009).

4.4 Implications for BMI Algorithm Design. While our letter does not aim to propose a new BMI algorithm, the work presented here contributes to the theoretical basis for new BMI algorithms focused on optimizing closed-loop performance. In order to apply our results to BMI controller design, it is important to recognize a stark division among the various existing BMI algorithms that loosely draw on the thematic elements of closed-loop control. A critical examination of BMI algorithms that invoke the language of closed-loop control reveals two fundamentally distinct categories. The first category focuses on optimizing open-loop reconstruction error. The second category focuses on optimizing closed-loop performance, exploiting the potential of human-in-the-loop dynamics.

The first category, based on recursive Bayesian estimation, proposes state equations (latent variable models) that describe the dynamics of a controller operating a linear gaussian plant (Kemere & Meng, 2005; Shanechi et al., 2010; Srinivasan et al., 2006). Although these three papers are titled with different emphasis (“state space analysis,” Srinivasan et al., 2006; “feed-forward control,” Kemere & Meng, 2005; “feedback-control,” Shanechi et al., 2010), they all propose nearly mathematically equivalent state-space models that represent closed-loop controllers on linear-gaussian plants. The “feed-forward control” title in Kemere and Meng (2005) is particularly understated in that the authors actually develop a linear feedback control policy as the basis for their state equation (equation 8). There is little practical resulting mathematical difference between the various methods. Earlier work (Diedrichsen et al., 2009; Scott, 2004; Todorov, 2004; Todorov & Jordan, 2002) provides the theoretical justification for the premise of these algorithms that the brain is a closed-loop controller in the operation of limb movement. This category of algorithms is geared to optimize trajectory reconstruction error, described in the experimental BMI literature as “open-loop decoding error.” Despite modeling closed-loop dynamics, these papers do not exploit

the closed-loop dynamics of the human operating the actual brain-machine interface itself. For example, Shanechi et al. (2010) employ a state equation that models the closed-loop control of a muscle-like plant rather than the control of the BMI itself. In fact, these authors choose to illustrate the capability of their resulting method in offline decoding from synthetic neural data, where an algorithm that explicitly incorporated human control of the BMI itself would inherently need closed-loop validation. Partly as a result of a BMI design philosophy rooted in estimating offline arm movements, all three methods (Kemere & Meng, 2005; Shanechi et al., 2010; Srinivasan et al., 2006) loosely invoke the language of control without actually exploiting the potential of human-in-the-loop dynamics.

In contrast, the second, newer, category of BMI algorithms attempts to exploit the brain's dynamics as a closed-loop controller in order to optimize closed-loop performance (Dangi et al., 2011; Gage et al., 2005; Gilja et al., 2010; Golub et al., 2011; Jarzebowski et al., 2008; Orsborn et al., 2012). Our letter is relevant to this second category of BMI algorithms: it provides theoretical justification for their premise that the brain is a closed-loop controller in the operation of BMI algorithms. Early work by Jarzebowski et al. (2008) redesigned the P300-based speller by explicitly modeling the brain as a closed-loop controller, resulting in a provably optimal strategy for dynamically adjusting alphanumeric menus (sensory feedback). Separate work by Gilja et al. (2010), and subsequently Dangi et al. (2011) and Orsborn et al. (2012) examined modified "cursorGoal" state equations that incorporate the current state of the actuator (an on-screen pointer) in the context of a recursive Bayesian decoder. Newer work along these lines incorporates feedback information theory (Omar et al., 2011) and team decision theory (Kim & Coleman, 2011) in the closed-loop design of BMI algorithms. Most recently, Golub et al. (2011) proposed a BMI algorithm that demonstrates particularly poor open-loop decoding error but enhanced closed-loop control relative to other competing methods that have better open-loop decoding error. The development of a channel selection strategy that incorporates the closed-loop perspective may be another opportunity for revision to existing BMI algorithm theory (see section E.6). The various new strategies discussed here (and likely others beyond this limited literature search) represent exciting new efforts to exploit the dynamics and limitations of learning (as frequently described in the neuroscience literature) and signal generation from the biological neural systems (humans) that control them.

Glossary

Unless otherwise noted, capital variables used in formulas are matrices and lowercase variables are vector random variables or scalar constants, as defined below.

General Parameters

Name	Dimension	Description
T	Symbol	Denotes matrix transpose
N	Scalar	Number of neurons
k	Scalar	Index representing discrete time
M	Scalar	Number of dimensions in the state. In this study, there are five parameters (see x_k)
K	Scalar	Number of dimensions in the control input, here $K = 2$ (see u_k)
Δ	Scalar	The time step used in discretizing the simulation
p_x, p_y	Scalar	x, y position of the cursor
v_x, v_y	Scalar	x, y velocity of the cursor
x_k	$M \times 1$	State of the closed-loop system, incorporating various kinematics of the cursor. Equivalent to the output of the neuroprosthetic device decoder. In this study, $M = 5$ to accommodate 2D cursor position, 2D cursor velocity, and the scalar 1 to allow affine functions of the state
n_k	$N \times 1$	Outputs of the observation process, that is, a vector of the binned spike counts from all neurons in the ensemble at time step k

Modeling the Output Neural Signals (see equation (A1))

Neural Signal Model Parameters (see equation A.1)

N (number of neurons)	96
c_i (baseline firing rate, in spikes/sec)	10
b_i (preferred direction parameters, in units of $\frac{\text{spikes/s}}{\text{cm/s}}$)	$\frac{14 \text{ spikes/s}}{20 \text{ cm/s}} \times (\cos \theta, \sin \theta)$ where $\theta \sim \text{uniform}(0, 2\pi)$

Modeling the Neuroprosthetic Device

Kalman Filter

\tilde{x}_k	$M \times 1$	The Kalman filter's hidden (latent) state variable, equivalent to the user's intentions for the cursor kinematics (2D position and velocity). In this study, $M = 5$ to accommodate 2D intended cursor position, 2D intended cursor velocity, and the scalar 1 to allow affine functions of the state.
K_k and K	$M \times N$	"Kalman gain." This matrix transforms observations into changes in the posterior of the state vector. The Kalman gain defines how much deviations in the predicted observations (the "innovation" in estimation theory) translate to deviations in the predicted state in calculating the posterior state estimate. Generally this matrix (K_k) changes as the posterior variance is updated. The steady-state Kalman gain (K) is the asymptotic value of K_k .

Continued.

Kalman Filter

$x_{k+1 k}$	$M \times 1$	Mean of the prediction density on user intention at time step k .
$V_{k+1 k}$	$M \times M$	Covariance of the prediction density on user intention at time step k .
$x_{k k}$	$M \times 1$	Mean of the posterior density on user intention at time step k .
$V_{k k}$	$M \times M$	Covariance of the posterior density on user intention at time step k .
F	$M \times M$	State transition matrix used for the random walk state equation (equation B.2).
H	$N \times M$	Multiplies the state in the gaussian observation model (equation B.7).
ε_k	$M \times 1$	Zero-mean additive gaussian noise in the random walk state equation (latent variable model, equation B.2) used by the KF to describe how user intention is expected to evolve one step into the future.
q_k	$N \times 1$	Zero-mean additive gaussian noise in the observation equation (observation model, equation B.7) used by the KF to approximate how intended velocity drives neural activity.
Σ	$M \times M$	Covariance of random gaussian vector ε_k in the state equation, B.2.
Θ	$N \times N$	Covariance of random gaussian vector q_k in the observation equation, B.7.

Optimal Linear Estimator (OLE) and Population Vector Algorithm (PVA)

P	$N \times 2$	Preferred direction matrix, calculated in equation B.13. Each row contains the 2D normalized preferred direction for a different neuron.
S	$N \times N$	Normalizing matrix, combined with H to calculate P in equation B.13.
\bar{n}_k	$N \times 1$	Recentered and rescaled binned spiked counts
D	$2 \times N$	Matrix that maps observations to the decoded velocity, calculated differently for the OLE and PVA

Modeling the Neural Control Network (Linear Quadratic Regulator)

u_k	$K \times 1$	The control input that the controller applies to affect the state. In this study, the control input is the user's intention for the 2D velocity of the cursor.
A	$M \times M$	The state transition matrix in the model of the LQR. This is typically not equal to F .
B	$M \times K$	Matrix that defines how control inputs affect the state
w_k	$M \times 1$	Zero-mean gaussian noise added to the state at each step of LQR control.
Q	$M \times M$	Matrix that defines the quadratic form of the cost terms related to state.

Continued.

R	$K \times K$	Matrix that defines the quadratic form of the cost terms related to control inputs
L	$K \times M$	Matrix defining the linear mapping from the current state to the optimal control input, where the optimal control policy is $u^*(x) = Lx$.
J	Scalar	Number of sensory feedback steps for each decoding bin. Allows the model to represent the brain as it receives sensory feedback at the fixed refresh rate of the monitor while decoding bin width is independently varied.
Neural Control Network (LQR) cost parameters (described in Equations C.4, C.10, C.11)		
α	Scalar	Weight for cursor distance from target. Unless otherwise specified, $\alpha = 0.18$.
β	Scalar	Weight for nonzero cursor speed. Unless otherwise specified, $\beta = 0.1$.
γ	Scalar	Weight for nonzero intended speed. Unless otherwise specified, $\gamma = 0.1$.

Appendix A: Modeling the Output Neural Signals

Our model for the output layer of neural signals is identical to Cunningham et al. (2011), which simulates cosine-tuned velocity-dependent spiking activity from motor cortical neurons. In this model, each neuron spikes as an inhomogeneous Poisson process dependent on intended velocity. Each neuron i has a preferred direction b_i and a minimum firing rate c_i . The instantaneous mean firing rate of neuron i at time k is then

$$\lambda_k^i = b_i \cdot v_k + c_i. \quad (\text{A.1})$$

Based on this formula, the preferred direction of a neuron is the intended direction of movement that elicits maximal firing rate. For neuron i , the number of spikes generated for a discrete amount of time Δ is drawn from the following distribution:

$$n_k^i \sim \text{Poisson}(\max(\lambda_k^i * \Delta, 0)). \quad (\text{A.2})$$

Parameters (see the Glossary) are chosen to produce realistic spiking rates, with a minimum firing rate of 10/s and a firing rate of 24/s if the cursor is moving at 20 cm/s along the preferred direction.

Appendix B: Modeling the Neuroprosthetic Device

The conventional neuroprosthetic device implements a neural signal decoder that maps signals from output neurons into an estimate of the user's

intention for the device. Our models for the neuroprosthetic device are identical to the implementations of three mainstream neural signal decoders: the Kalman filter (KF), optimal linear estimator (OLE), and population vector algorithm (PVA). In this section, we recapitulate the implementation of these decoders in the context of our closed-loop model.

B.1 Kalman Filter. An alternative introductory textbook account of the KF from the probabilistic perspective is found elsewhere (Stengel, 1994). In general, the KF attempts to produce recursive estimates for a hidden signal \tilde{x}_k from a set of noisy observations on that signal $\{n_0, n_1, \dots, n_k\}$. In neuroprosthetic devices, \tilde{x}_k represents user intentions for the device at time step k , and n_k are neural signals that reflect those intentions. In our model, the device is simply a computer cursor, so $\{\tilde{x}_k\}$ represents intended kinematics (position \tilde{p}_x, \tilde{p}_y and velocity \tilde{v}_x, \tilde{v}_y in 2D):

$$\tilde{x}_k = (\tilde{p}_x \quad \tilde{p}_y \quad \tilde{v}_x \quad \tilde{v}_y \quad 1)^T. \quad (\text{B.1})$$

The constant 1 in this vector is used to implement an affine state equation. More generally, the state vector \tilde{x}_k can incorporate any parameters that relate to the neuroprosthetic device, such as position, angles of joints, or force. In order to estimate \tilde{x}_k from $\{n_0, n_1, \dots, n_k\}$, the KF defines a state equation and an observation equation. We describe the meaning and implementation of these components below.

B.1.1 Defining the KF State Equation. The state equation represents the Kalman filter's notion for how the user's subsequent intention \tilde{x}_{k+1} for the device depends on the current intention \tilde{x}_k . In Bayesian terminology, the state equation embodies a prior distribution on the user's intention. In the standard implementation, this equation is a gaussian random walk:

$$\tilde{x}_{k+1} = F\tilde{x}_k + \varepsilon_k, \quad (\text{B.2})$$

where F is an $M \times M$ fixed linear map and $\varepsilon_k \sim N(0, \Sigma)$. Commonly, and in our implementation, F is chosen to represent simple kinematics that propagates velocity into position at discrete time step widths Δ seconds. For the $5 \times 1 \tilde{x}_k$ described in equation B.1, F is given by

$$F = \begin{pmatrix} 1 & 0 & \Delta & 0 & 0 \\ 0 & 1 & 0 & \Delta & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \quad (\text{B.3})$$

Note that the 0s and 1s in equation B.3 are unitless, and Δ is in seconds. The covariance is set to be nonzero only in the velocity-related entries, to represent the KF's expectation that the user's intended velocity will change smoothly (termed stochastic continuity) over time:

$$\Sigma = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \Delta \cdot 10^2 \text{cm}^2/\text{s}^3 & 0 & 0 \\ 0 & 0 & 0 & \Delta \cdot 10^2 \text{cm}^2/\text{s}^3 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \quad (\text{B.4})$$

Note that we have s^3 in the denominator to allow final units of cm^2/s^2 in the corresponding entry. The value of these diagonals matches the order of magnitude seen in the changes of velocity from simulated reaches in the actual reach generation. The covariance was scaled with bin width Δ , as described in equation B.4.

B.1.2 Defining the KF Observation Model. The KF decoder models binned spike counts for neuron i at time step k as they depend on the intended velocity:

$$n_k^i = \tilde{a}_i[\tilde{v}_x]_k + \tilde{b}_i[\tilde{v}_y]_k + \tilde{c}_i + \omega_k^i, \quad (\text{B.5})$$

where $\omega_k \sim N(0, \tilde{\Theta})$ are identically and independently distributed across k . Note that this observation model is mismatched because neural observations are actually generated as an inhomogeneous Poisson process (see equation A.2).

The parameters of this model are estimated from binned spiking activity using linear regression. Specifically, cursor velocity is regressed against binned spike count. Data for this regression are obtained from a training set of eight simulated reaching movements. Reaches begin at evenly spaced points on the circle ($0, \pi/4, \pi/2, \dots, 7\pi/4$) and end at the origin, generated by a linear quadratic controller (LQR). Appendix C explains the LQR in detail. (For an introductory text on this subject, see Bertsekas, 2005.) To estimate the covariance $\tilde{\Theta}$, the mean of squared residuals of observed spikes from the linear model is used as an estimate of the variance of each neuron's spike generation. The covariance estimate $\tilde{\Theta}$ is set to a diagonal matrix with the i th neuron's estimated variance at the i th diagonal entry.

Results from linear regression are compiled into the H matrix (dimensions $N \times M$). Note the tilde is removed from a , b , and c to indicate that estimates of these parameters (rather than their true values) are used to

form H from the linear regression:

$$H = \begin{pmatrix} 0 & 0 & a_1 & b_1 & c_1 \\ 0 & 0 & a_2 & b_2 & c_2 \\ 0 & 0 & \vdots & \vdots & \vdots \\ 0 & 0 & a_N & b_N & c_N \end{pmatrix}, \quad (\text{B.6})$$

where H relates the binned neural activity n_k (dimensions $N \times 1$) and the intended cursor kinematics \tilde{x}_k (dimensions $M \times 1$):

$$n_k = H\tilde{x}_k + q_k. \quad (\text{B.7})$$

Here q_k is a zero-mean gaussian variable with covariance Θ .

B.1.3 Kalman Filter Equations. The KF combines the state and observation equations to estimate the user's intention from output neural activity. Because this neural activity is a noisy representation of the user's intention, the KF uses a gaussian probability distribution (also called probability density) to represent its estimate of intention given neural activity. The notations $x_{k|j}$ and $V_{k|j}$ are used to indicate the mean and variance of this distribution, respectively, for the user's intention at time step k , given neural activity up to and including time step j . The KF alternates between two steps:

1. Prediction: compute $x_{k+1|k}$ and $V_{k+1|k}$ from $x_{k|k}$ and $V_{k|k}$.
2. Update: compute $x_{k+1|k+1}$ and $V_{k+1|k+1}$ from $x_{k+1|k}$ and $V_{k+1|k}$.

The prediction step uses the following equations, with F and Σ given in equations B.3 and B.4 based on the state equation, B.2:

$$x_{k+1|k} = Fx_{k|k}, \quad (\text{B.8})$$

$$V_{k+1|k} = FV_{k|k}F^T + \Sigma. \quad (\text{B.9})$$

The update step uses the following equations, with H given in equation B.6 and Θ estimated as described in section B.1.2, based on the observation equation, B.7:

$$K_{k+1} = V_{k+1|k}H^T(HV_{k+1|k}H^T + \Theta)^{-1}, \quad (\text{B.10})$$

$$x_{k+1|k+1} = x_{k+1|k} + K_{k+1}(n_{k+1} - Hx_{k+1|k}), \quad (\text{B.11})$$

$$V_{k+1|k+1} = (I - K_{k+1}H)V_{k+1|k}. \quad (\text{B.12})$$

n_{k+1} is a vector of binned spike counts across the neuronal ensemble. The matrix K_k , called the Kalman gain, defines the extent to which new observations n_{k+1} affect the filter's estimate of intended cursor kinematics. A useful property of the Kalman gain is that it does not depend on the observations received. Under appropriate conditions, the Kalman gain converges over time, so that the entire filter, originally linear and time varying, becomes linear and time invariant. This property will be useful in constructing the neural control network that operates the KF neuroprosthetic device, as described in section C.3.1.

B.2 Optimal Linear Estimator and Population Vector Algorithm. An alternate discussion of the OLE and PVA is found elsewhere (Chase et al., 2009). We present these methods using notation that is consistent with the remainder of our model. Both OLE and PVA decode velocity at time step k through a linear transformation of a recentered-and-rescaled vector of binned spike counts at time step k , without incorporating neural activity from previous time steps.

Both decoders use an $N \times 2$ matrix of normalized preferred directions P . In our notation, the preferred directions are encoded in the third and fourth columns of H ($N \times 5$) that relate to the velocity (see equation B.6). The relevant columns of H can be extracted and normalized across rows to get the desired matrix P :

$$P = SH \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}^T. \quad (\text{B.13})$$

The T in this formula denotes the matrix transpose. The matrix to the right of H extracts the third and fourth columns of H . The matrix S ($N \times N$) is a diagonal matrix such that the rows of P are normalized. Accordingly, $S_{i,i} = [(H_{i,3})^2 + (H_{i,4})^2]^{-1/2}$. In other words, $S_{i,i}$ is the reciprocal of the L_2 norm of the i th row of H $\begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}^T$. The recentered-and-rescaled vector of binned spike counts is given by

$$\tilde{n}_k = S(n_k - H(0 \ 0 \ 0 \ 0 \ 1)^T) = Pu_k + Sq_k. \quad (\text{B.14})$$

Define additionally:

$$D_{OLE} = (P^T P)^{-1} P^T, \quad (\text{B.15})$$

$$D_{PVA} = \frac{2}{N} P^T. \quad (\text{B.16})$$

Using the respective D matrices, the OLE and PVA can be summarized by two steps:

1. Move the current cursor position by the current cursor velocity times bin width Δ .
2. Estimate the new cursor velocity at time step k as $D\bar{n}_k$.

Recall that an estimator is unbiased if the average difference between estimate and true value is zero. Given valid model assumptions, the OLE is an unbiased decoder. In contrast, the PVA is biased when the preferred directions in P are not distributed uniformly, resulting in systematic differences between intended and decoded velocity during open-loop decoding, as explored in Figure 4.

Appendix C: Modeling the Neural Control Network

The neural control network is modeled as a stochastic optimal controller. The stochastic optimal control model has been used extensively in the analysis of natural arm movements. In our model, this controller uses sensory feedback on cursor kinematics to drive primary motor cortical activity. The term *stochastic* refers to the controller's ability to perform despite uncertainty in decoded cursor movements (in this case, resulting from output neural activity). The term *optimal* refers to the minimization of costs associated with cursor kinematics and intended velocity, explicitly rewarding goal-directed cursor movements while indirectly conserving chemical energy related to the production of action potentials.

The linear quadratic regulator (LQR) is one type of stochastic optimal controller, used for problems where the state to be controlled (in our case, cursor kinematics) evolves linearly with the addition of gaussian noise. Because our binned output neural activity is only approximately gaussian with linear neuroprosthetic devices (KF, OLE, PVA), the LQR framework provides an approximate stochastic optimal controller. Nevertheless, the LQR is a computationally tractable choice to model the neural control network that sufficiently replicates experimental data as described in the main text.

To generate reasonable cursor movements, the neural control network must account for a system that includes both output neurons and the actual neuroprosthetic device. These components, enclosed by the dashed outline in Figure 1A, represent the "plant" in standard control theory nomenclature. In devising a control policy, we define a state vector that describes the plant, in similar fashion to the state vector used for the KF (see equation B.1):

$$x_k = (p_x \quad p_y \quad v_x \quad v_y \quad 1)^T. \quad (\text{C.1})$$

In this case, we have removed the tilde symbol from x_k to indicate that the relevant state to the control network is decoded 2D cursor position and velocity. The constant 1 is added to accommodate affine state equations.

The output neurons are described using the neural observation model (see equation B.7), but rewritten to emphasize that the intended 2D velocity is actually the 2×1 control input u_k determined by the neural control network:

$$n_k = H(0 \quad 0 \quad u_k \quad 1)^T + q_k. \quad (\text{C.2})$$

The LQR also requires a state equation that describes how cursor kinematics x_k are expected to evolve in time. Unlike the KF state equation (see section B.1.1), the plant state equation additionally specifies how control inputs u_k from the neural control network influence cursor kinematics. In section C.1, we introduce equations that implement the LQR for a generic linear gaussian state equation. In section C.2, we describe how LQR behavior can be shaped to implement reaching movements of the neuroprosthetic cursor while conserving spikes. In section C.3, we specify how the generic LQR equations from section C.1 can be customized to control neuroprosthetic devices based on the KF, OLE, and PVA. In section C.4, we discuss how the final model represents sensory feedback that occurs at a timescale that is finer than the bin width of the neuroprosthetic device.

C.1 Generic Equations for the Linear Quadratic Controller. In this section, we introduce equations that implement the LQR for a generic linear gaussian state equation. (This material can also be found in an introductory text: Bertsekas, 2005.) The LQR gives an optimal control policy for linear systems with additive gaussian noise under a quadratic cost function.

With the state of the plant (system to be controlled) defined in equation C.1, the state dynamics is expressed with the following state equation:

$$x_{k+1} = Ax_k + Bu_k + w_k. \quad (\text{C.3})$$

This is similar to the KF state equation, B.2, with the F replaced by A and ε_k replaced by w_k . However, the plant state equation, C.3, includes an additional term for the control input u_k that is decided by the neural control network. In the generic formulation, the LQR gives an optimal control policy on how to set the control inputs u_k based on observations of the cursor x_k such that the expected value of a quadratic cost function on both state and control inputs is minimized over all time. The formula for this cost is given by

$$E_{w_k} \left\{ \sum_{k=0}^{\infty} x_k^T Q x_k + u_k^T R u_k \right\}. \quad (\text{C.4})$$

Q and R are square matrices that determine the relative costs associated with state and control, respectively. The optimal control policy given by the LQR derivation is a linear function of the state:

$$u_k^*(x_k) = Lx_k, \quad (\text{C.5})$$

where

$$L = -(B^TGB + R)^{-1}B^TGA, \quad (\text{C.6})$$

and G is computed by solving the algebraic Ricatti equation:

$$G = A^T(G - GB(B^TGB + R)^{-1}B^TG)A + Q. \quad (\text{C.7})$$

An approximate solution for G is obtained by repeating several steps of the following recursion, until the sequence G_i, G_{i+1}, \dots converges to within some tolerance,

$$G_{i+1} = A'(G_i - G_iB(B^TG_iB + R)^{-1}B^TG_i)A + Q, \quad (\text{C.8})$$

where

$$G_0 = Q. \quad (\text{C.9})$$

For this letter, we terminated the recursion when the Frobenius norm of $G_{i+1} - G_i$ was lower than $1e-7$.

C.2 Using the LQR Cost Function to Generate Reaching Behavior.

The cost function represents constraints of the reaching task. The LQR is used to generate reaches toward the origin by setting the cost matrices Q and R such that the origin represents a low-cost state. There are three costs:

1. α —Penalizes squared distance to origin. This parameter incentivizes reaches toward the origin.
2. β —Penalizes square of cursor velocity. This parameter incentivizes reaching and holding to zero velocity. Also, because kinetic energy is proportional to the square of velocity, β relates to energy dissipated in applications that involve movement of a robotic arm or FES-enabled limb.
3. γ —Penalizes square of intended cursor speed. Because spiking rates increase with intended cursor speed, γ is effectively a cost associated with generating spiking activity.

These parameters are set in the cost matrices as follows:

$$Q = \begin{pmatrix} \alpha & 0 & 0 & 0 & 0 \\ 0 & \alpha & 0 & 0 & 0 \\ 0 & 0 & \beta & 0 & 0 \\ 0 & 0 & 0 & \beta & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad (\text{C.10})$$

$$R = \begin{pmatrix} \gamma & 0 \\ 0 & \gamma \end{pmatrix}. \quad (\text{C.11})$$

C.3 Customizing the Generic LQR Framework to Represent the Neural Control Network in Closed-Loop Neuroprosthetic Operation. As an introductory exercise, suppose that neural signals are perfectly decoded in closed-loop neuroprosthetic operation. In order to represent velocity control, as investigated in Cunningham et al. (2011) and elsewhere, we choose the following values for matrices in the state equation C.3, where Δ indicates the decoder bin width in units of seconds:

$$A = \begin{pmatrix} 1 & 0 & \Delta & 0 & 0 \\ 0 & 1 & 0 & \Delta & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \quad (\text{C.12})$$

$$B = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}. \quad (\text{C.13})$$

Equations C.12 and C.13 imply perfect decoding where the neural control network determines cursor kinematics directly. In practice, decoders like the KF, OLE, and PVA are not perfect, and the neural control network acts indirectly on cursor kinematics through noisy output neurons and imperfect decoding. In sections C.3.1 and C.3.2, we provide the necessary corrections to accurately model the KF, OLE, and PVA.

C.3.1 LQR Control of the Kalman Filter. The state dynamics of the generic LQR model given in section 5.1 assumes the neuroprosthetic device is time invariant, whereas the standard Kalman filter is linear but time varying. This is a result of the Kalman gain K_k (see equation B.10), which is indexed by time. The Kalman gain can converge to a steady state under certain common conditions, not discussed here, in part because the evolution of the Kalman gain is also independent of the observation process (Bertsekas,

2005; Stengel, 1994). We estimate a steady-state Kalman gain by running equations B.10 and B.12 forward until the Frobenius norm of the change in the Kalman gain matrix is less than $1e-7$. We denote this steady-state Kalman gain by K . The resulting steady-state KF is linear time invariant, which is fully compatible with the LQR model in section C.1.

Given that the observations n_k are distributed according to equation C.2 and the KF takes the form B.11, the evolution of the full plant (including output motor neurons and KF) can be expressed as a linear transformation of the state (x_k , containing 2D cursor position and velocity) and control inputs (u_k , containing user-intended velocity):

$$x_{k+1} = Fx_k + K \left(HF \begin{pmatrix} p_x \\ p_y \\ u_k \\ 1 \end{pmatrix} + q_k - HFx_k \right). \quad (C.14)$$

Recall that p_x and p_y are actual cursor position, x_k contains actual cursor position and velocity, and u_k is user-intended cursor velocity. Equation C.14 can be separated into a sum of linear transformations on the state and the control input with additive gaussian noise:

$$x_{k+1} = F \begin{pmatrix} p_x \\ p_y \\ 0 \\ 1 \end{pmatrix} + (F - KHF) \begin{pmatrix} 0 \\ 0 \\ v_k \\ 0 \end{pmatrix} + KHF \begin{pmatrix} 0 \\ 0 \\ u_k \\ 0 \end{pmatrix} + Kq_k. \quad (C.15)$$

This matches the original form of the LQR formulation, where

$$A = F \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} + (F - KHF) \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad (C.16)$$

$$B = KHF \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix}, \quad (C.17)$$

$$w_k = Kq_k. \quad (C.18)$$

Note that equation C.16 differs from equation C.12 in that it does not completely discard the current estimate of velocity v_k (because of the nonzero entries in the right-most matrix). This reflects the fact that the Kalman filter integrates over prior neural signal observations, in contrast to the OLE and PVA.

C.3.2 LQR Control of the Optimal Linear Estimator and Population Vector Algorithm. Both the OLE and PVA discard the previous decoded velocity, resulting in an A matrix in the state equation that is identical to the perfect decoder case (see equation C.12). Decoded velocity in both cases can be written as $D\tilde{n}_k$, where \tilde{n}_k represents the recentered-and-rescaled firing rate vector as defined in equation B.14 and D is selected differently for each of the two methods as described in equations B.15 and B.16. Combining the normalized observation model given in equation B.14 with the decoded velocity $D\tilde{n}_k$ gives a formula that relates decoded velocity (v_k) to intended velocity (u_k):

$$v_k = DPu_k + DSq_k. \quad (\text{C.19})$$

These calculations define the components of the plant state equation, C.3, for OLE and PVA:

$$A = \begin{pmatrix} 1 & 0 & \Delta & 0 & 0 \\ 0 & 1 & 0 & \Delta & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \quad (\text{C.20})$$

$$B = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}^T DP, \quad (\text{C.21})$$

$$w_k = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}^T DSq_k. \quad (\text{C.22})$$

Recall from section B.2 that for the OLE, $D = (P^T P)^{-1} P^T$, where P is given in equation B.13, resulting in $B = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$. For the PVA, $D = \frac{2}{N} P^T$, so that $B = \frac{2}{N} P^T P$. For the PVA, the fact that B is not necessarily the identity (depending on the value of P) embodies the notion that the PVA can be a biased decoder, depending on the population of neurons (via the value of P).

C.4 Extending the LQR to Uncouple the Time Resolution of Sensory Feedback and Decoder Bin Width. The LQR framework as described in

the previous sections allows control inputs u_k and state x_k to change simultaneously on every time step. In section 3, the behavior of decoders on different bin widths is analyzed. Under the standard LQR framework, if the decoder bin width is 300 ms, the controller can “perceive” states only every 300 ms. This problem is manifest in the dependence between cost function and bin width. For example, if the bin width is 25 ms, then the cost is evaluated on states every 25 ms. If the bin width is instead 300 ms, then the cost function is evaluated on states every 300 ms. In reality, the human receives cursor state updates at the screen refresh rate rather than the decoding rate, which is typically slower. In order to faithfully compare closed-loop behavior across various decoding bin widths, we needed the controller to receive cursor state at a constant rate (in our letter, 5 ms) independent of the decoding bin width. This is analogous to varying decoder bin width while the computer screen refresh rate is fixed at 200 Hz. Below, we describe how the model used in the main text uncouples the time resolution of sensory feedback and decoder bin width.

We reformulate the LQR framework so that the cost represents ongoing kinematics at 5 ms resolution, while the decoding time step is substantially larger (tested at values ranging from 25 ms to 300 ms). We refer to the time between the decoding time steps as the “drift period,” where cursor position is updated to reflect the latest decoded velocity. For simplicity, suppose the screen is refreshed J times for every Δ second drift period, where Δ is also the decode bin width. Let

$$F = \begin{pmatrix} 1 & 0 & \Delta/J & 0 & 0 \\ 0 & 1 & 0 & \Delta/J & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \quad (\text{C.23})$$

The sequence of cursor states (position and velocity kinematics in 2D) in the drift period between decoder updates is $[x_k, Fx_k, F^2x_k, \dots, F^{J-1}x_k]$. The cost in the state accumulated between decode periods (omitting the control cost) is then

$$x_k^T Q x_k + (F x_k)^T Q (F x_k) + \dots + (F^{J-1} x_k)^T Q (F^{J-1} x_k). \quad (\text{C.24})$$

This can be expressed as a quadratic form of x_k :

$$x_k^T (Q + F^T Q F + \dots + (F^{J-1})^T Q F^{J-1}) x_k. \quad (\text{C.25})$$

The cost accumulated for the control input between decode periods is JR .

With these equations, we can use the simple LQR model to accommodate sensory feedback at finer timescales than the decoding bin width. The

modified state transition, state cost, and control input cost matrices are

$$\tilde{A} = F^{J-1}A, \quad (\text{C.26})$$

$$\tilde{B} = B, \quad (\text{C.27})$$

$$\tilde{Q} = Q + F^T Q F + \dots + (F^{J-1})^T Q F^{J-1}, \quad (\text{C.28})$$

$$\tilde{R} = JR. \quad (\text{C.29})$$

Note that \tilde{B} is used to express the effects of one decode at the end of the J steps. Because the decoder equations are not changed by modeling sensory feedback at finer timescales, $\tilde{B} = B$ as defined in sections C.3.1 and C.3.2.

Appendix D: Numerical Issues and Implementation Details _____

The Ricatti equation recursions for the LQR and the steady-state Kalman gain were terminated when the Frobenius norm of the difference between successive Kalman gains dropped below a threshold. As a caution, note that we had originally implemented a large, constant number of iterations, which did not consistently provide convergence.

The Kalman filter described in section B.1, a linear time-varying algorithm, was made time invariant (LTI) by using the steady-state Kalman gain. This was useful for deriving the appropriate LQR model to match a Kalman filter. However, using the LTI Kalman filter for decoding purposes initially generates inaccurate decodes, since that filter acts as if it has already received many observations from the past that have not actually occurred. To resolve this issue in our closed-loop simulations, the LQR derived from an LTI Kalman filter is used for the controller, whereas the decoder is the standard (linear, time-varying) Kalman filter.

Appendix E: Parameter Sensitivity Analysis _____

Is our result robust? In other words, does our reproduction of experimental behavior (see Figures 2, 4, and 5) require delicate parameter tuning? This question is central to the impact of any theory, including this one. Given that parameters are likely to vary between subjects, we need to be able to demonstrate that our essential claimed model behaviors (see Figures 2, 4, and 5) are preserved across a wide range of parameter choices. Below, we demonstrate that our essential claimed behaviors are preserved across at least four orders of magnitude in parameter choice.

E.1 Number of Parameters. The model employs several different constants that relate to neural spiking properties, neural signal decoder equations, and control. Nearly all of these constants are derived from previously published experimental data. The neural spiking parameters are based on

point-process models of motor cortical neurons that have been fit to primate data (Truccolo et al., 2005). The neural prosthetic decoder equations are constrained by the Kalman filter, PVA, and OLE algorithms, where parameters are fit by maximum likelihood applied to the simulated neural activity. The controller approximates the plant (which includes the output neural activity and neural prosthetic decoder equations) as a linear state equation that is also fit by maximum likelihood. Reaction time is a constant at 200 ms, within range of experimental behavior.

The only parameters not determined by prior experimental evidence are those of the cost function (see appendix C). When equations C.4, C.10, and C.11 are combined, the cost function can be reexpressed in terms of mean squared error (MSE) and energy:

$$\begin{aligned} \text{Cost} = & \alpha(\text{MSE in target position}) + \beta(\text{MSE in target velocity}) \\ & + \gamma(\text{total controller energy dissipated}). \end{aligned} \quad (\text{E.1})$$

In the main text, α/β is fixed at 1.8. Because cost is arbitrarily scaled, γ/α is our only free parameter. Note the physical meaning of this parameter: it embodies the extent to which the human expends energy (in the form of action potentials) versus attempting to achieve an accurate reaching movement with crisp stopping behavior. Having only one free parameter in the model greatly expedites the parameter sensitivity analysis, which consequently involves sweeping one parameter, γ/α , across several orders of magnitude.

E.2 Establishing Whether Model Behavior Is Consistent with Experiment. How do we determine in automated fashion if the essential behaviors displayed in figures 2, 4, and 5 are preserved across multiple orders of magnitude of γ/α ? The essential behaviors the model must reproduce are:

- Figure 2: Mean integrated distance (MID) increases with bin width in both open- and closed-loop control.
- Figure 4: Closed loop control results in a reduction of PVA bias versus open-loop control.
- Figure 5: Measured neuronal preferred directions shift between arm and closed-loop brain-machine interface control for the PVA, but shift is negligible for the OLE.

To automate the process of determining the validity of these claims across a wide number of γ/α , we conduct hypothesis tests as follows:

- Figures 6A and 6B, corresponding to Figure 2B in the main text: H_0 : slope of MID versus bin width = zero. H_1 : slope > zero. The p value on the slope is calculated from the observed Fisher information matrix

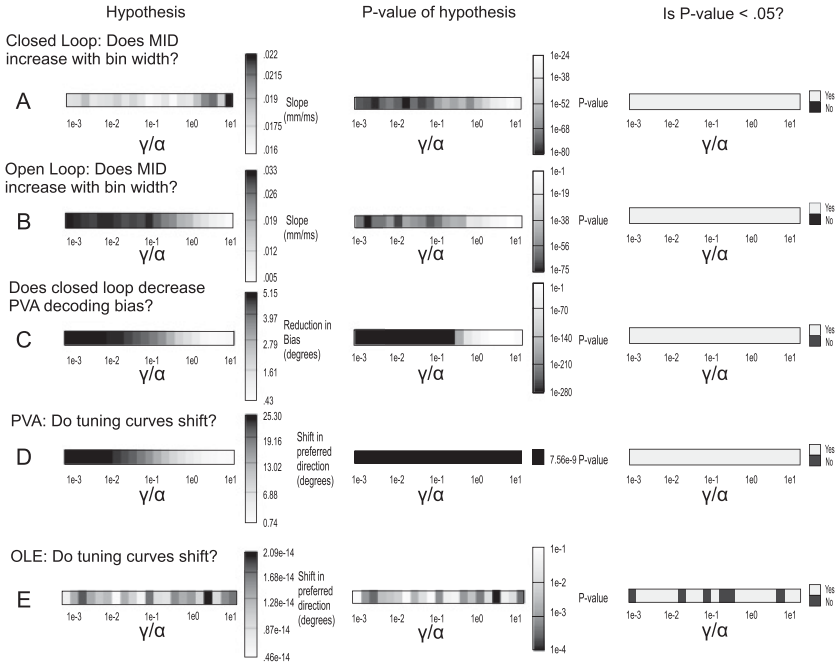


Figure 6: Robustness of results. This figure illustrates statistical tests to establish model robustness over a range of 20 values of γ/α from $1e-3$ to $1e1$, representing variations in the relative cost of control (coefficient γ) and the cost of deviation from the target state (coefficient α). Note α/β is fixed at 2 and logarithmic scale is used to display p values and γ/α . (A, B) Slope of mean integrated distance to target (MID) versus bin width is positive for closed loop (A) and open loop (B) across four orders of magnitude in parameter, consistent with Figure 2. (C) Closed-loop PVA control results in less bias than open-loop PVA control from the intended angle (i.e., less bias) compared to open-loop PVA control across four orders of magnitude in parameter, consistent with Figure 4. (D, E) Measured preferred directions shift between arm control and brain-machine interface control appreciably for the PVA (D), but not appreciably for the OLE (E), across four orders of magnitude in parameter, consistent with Figure 4. Details of methodology for panels A–E provided in sections E.2 and E.3. (A color version of this figure is available in the supplemental material.)

based on maximum likelihood estimation assuming MID is an affine gaussian function of bin width. This test is conducted on MID versus bin width graphs in closed-loop mode (see Figure 6A) and open-loop mode (see Figure 6B).

- Figure 6C, corresponding to Figure 4A in the main text: H_0 : Bias in decoded angle is unchanged between open- and closed-loop control

of the PVA. H_1 : Bias decreases with closed-loop control of the PVA. A one-tailed Wilcoxon signed rank (nonparametric) test is applied.

- Figures 6D and 6E, corresponding to Figures 5A and 5D, respectively: H_0 : No shift in neuronal preferred direction between arm control and brain-machine interface control. H_1 : Nonzero shift with the PVA (see Figure 6D), but negligible shift with the OLE (see Figure 6E). A two-tailed Wilcoxon signed rank (nonparametric) test is applied.

With hypothesis tests described as above, Figure 6 reports the magnitude of the corresponding statistic, as well as the p value, and a binary variable that indicates whether this p value meets a $p < 0.05$ threshold for statistical significance. γ/α is varied evenly on a logarithmic scale between $1e-3$ and $1e1$. The end result is that essential trends in Figures to 1, 2, 4, and 5 are reproduced across several orders of magnitude in γ/α .

E.3 Generating Data Points for the Parameter Sensitivity Analysis.

Neural ensemble sizes of 96 neurons for Figures 6A to 6C and 10 neurons for Figures 6D and 6E were employed for consistency with Figures 2, 4, and 5. Neuron parameters for the entire ensemble were drawn once for each panel in Figure 6 and held fixed across multiple trials such that variation in trial statistics was driven by the randomness of point process neural outputs, but not by variation in neural parameters. γ/α was varied evenly on a logarithmic scale between $1e-3$ and $1e1$.

Figures 6A and 6B each use 100 MID trials per data point, where each of the six data points represents a specific bin width. Every trial uses a newly drawn ensemble of neurons. Six bin widths were examined, as in Figure 2B: 25, 50, 100, 200, 250, and 300 ms. The p -value represents the statistical significance of the nonzero slope of the MID versus bin width graph, computed separately for each γ/α ratio. The slope (and hence p -value) for each γ/α ratio is computed from a regression on 100 MID trials per data point \times 6 data points = 600 MID trials.

Figure 6C uses a single fixed ensemble of neurons to calculate the bias in the first decoded velocity from 50,000 unique reaching trials for each γ/α value. The p -value represents the significance of reduction in PVA decoding bias between arm mode and closed-loop mode.

Figures 6D and 6E each examine the tuning curve shifts of a single neuron in a fixed population of 10 neurons. The shift in preferred direction is computed 100 times, each time with a different set of 1000 binned spike counts. This process is repeated for each value of γ/α .

E.4 Tuning Curve Shifts Occur with Ensembles That Include Neurons of Variable Baseline Firing Rate and Depth of Modulation. We confirmed that tuning curve shifts could be reproduced with an ensemble that included neurons of variable baseline firing rate and depth of modulation (see Figure 7). For 10 neurons, base firing rates are uniformly chosen from [2, 20] spikes/sec, and tuning depths (difference between maximum and

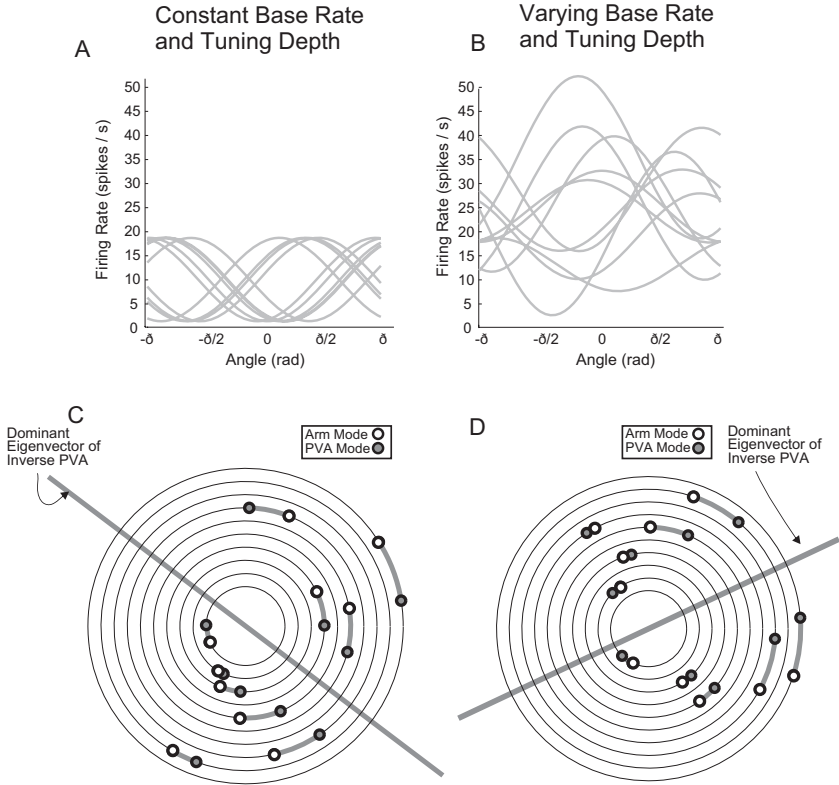


Figure 7: Tuning curve shift in ensembles with diversity of baseline firing rate and depth of modulation. (A) Tuning curves of 10 neurons drawn from the distribution described in section 2.2. (B) Tuning curves of 10 neurons drawn from a distribution where base firing rates are uniformly chosen from [2, 20] spikes/sec and tuning depths (difference between maximum and minimum firing rate) are uniformly chosen from [2, 20] spikes/sec. (C) Shift in measured preferred directions of the neurons plotted in panel A, transitioning between arm to PVA control with no control cost. This figure is identical to Figure 4A, reprinted for convenience. (D) Same as panel C, but using the neurons plotted in panel B. (A color version of this figure is available in the supplemental material.)

minimum firing rate) are uniformly chosen from [2, 20] spikes/sec. Tuning curves again shift with this more diverse set of neurons (see Figure 7).

E.5 Conclusion of Parameter Sensitivity Analysis. The sensitivity analysis concludes that trends claimed in Figures 2, 4, and 5 are robust across four orders of magnitude in parameter choice, as described in the Figure 6 captions. Also, tuning curves shifts persist with increased diversity in neural tuning curve parameters (see Figure 7).

E.6 Implications for Channel Selection in BMI Performance Optimization. Channel selection can have an impact on BMI performance in both open-loop decoding (Vargas-Irwin et al., 2010; Wahnoun, He, & Helms Tillery, 2006) and the resulting closed-loop performance (Wahnoun et al., 2006). Although our work here does not attempt to directly study this phenomenon, our analysis (see Figure 2H) explains how open-loop decoding quality could improve closed-loop performance. A channel selection process that improves open-loop decoding by detecting strong model mismatch might be expected to improve closed-loop performance by decreasing the magnitude of the unpredictable velocity decoding error that the user would need to correct at each time step. In contrast, our model suggests that when channel selection criteria focus on reducing predictable errors (bias), open-loop improvements might translate less effectively into skilled closed-loop performance, consistent with the guiding philosophy of earlier work (Chase et al., 2009).

Acknowledgments

This work was funded in part by the American Heart Association Scientist Development Grant (11SDG7550015) and UCLA Radiology. We thank David Rutledge at the California Institute of Technology and Sanjoy Mitter at MIT for institutional support.

References

- Berens, P. (2009). CircStat: A MATLAB toolbox for circular statistics. *Wiley Interdisciplinary Reviews: Computational Statistics*, 1, 128–129.
- Bertsekas, D. P. (2005). *Dynamic programming and optimal control* (3rd ed.). Belmont, MA: Athena Scientific.
- Brockwell, A. E., Rojas, A. L., & Kass, R. E. (2004). Recursive Bayesian decoding of motor cortical signals by particle filtering. *J. Neurophysiol.*, 91(4), 1899–1907.
- Carmena, J. M., Lebedev, M. A., Crist, R. E., O'Doherty, J. E., Santucci, D. M., et al. (2003). Learning to control a brain-machine interface for reaching and grasping by primates. *PLoS Biol.*, 1, E42.
- Chase, S. M., Schwartz, A. B., & Kass, R. E. (2009). Bias, optimal linear estimation, and the differences between open-loop simulation and closed-loop performance of spiking-based brain-computer interface algorithms. *Neural Netw.*, 22(9), 1203–1213.
- Cunningham, J. P., Nuyujukian, P., Gilja, V., Chestek, C. A., Ryu, S. I., & Shenoy, K. V. (2011). A closed-loop human simulator for investigating the role of feedback control in brain-machine interfaces. *J. Neurophysiol.*, 105(4), 1932–1949. doi:10.1152/jn.00503.2010
- Dangi, S., Gowda, S., Heliot, R., & Carmena, J. M. (2011). Adaptive Kalman filtering for closed-loop brain-machine interface systems. In *Proceedings of the 5th IEEE EMBS Conference on Neural Engineering* (pp. 609–612). Piscataway, NJ: IEEE.

- Diedrichsen, J., Shadmehr, R., & Ivry, R. B. (2009). The coordination of movement: Optimal feedback control and beyond. *Trends Cogn. Sci.*, 14(1), 31–39.
- Eden, U. T., Frank, L. M., Barbieri, R., Solo, V., & Brown, E. N. (2004). Dynamic analysis of neural encoding by point process adaptive filtering. *Neural Computation*, 16, 971–998.
- Fetz, E. E. (1969). Operant conditioning of cortical unit activity. *Science*, 163(870), 955–958.
- Gage, G. J., Ludwig, K. A., Otto, K. J., Ionides, E. L., & Kipke, D. R. (2005). Naive coadaptive cortical control. *J. Neural Eng.*, 2(2), 52–63.
- Ganguly, K., & Carmena, J. M. (2009). Emergence of a stable cortical map for neuroprosthetic control. *PLoS Biol.*, 7(7), e1000153.
- Ganguly, K., Dimitrov, D. F., Wallis, J. D., & Carmena, J. M. (2011). Reversible large-scale modification of cortical networks during neuroprosthetic control. *Nat. Neurosci.*, 14(5), 662–667.
- Georgopoulos, A. P., Schwartz, A. B., & Kettner, R. E. (1986). Neuronal population coding of movement direction. *Science*, 233(4771), 1416–1419.
- Gilja, V., Nuyujukian, P., Chestek, C. A., Cunningham, J. P., Yu, B. M., Ryu, S. I., et al. (2010). A high-performance continuous cortically-controlled prosthesis enabled by feedback control design. *SfN Abstract Viewer/Itinerary Planner*. Program/Poster 20.26.
- Golub, M. D., Yu, B. M., Schwartz, A. B., & Chase, S. M. (2011). Improving cursor stops in closed-loop brain-computer interfaces by leveraging trajectory curvature. *Society for Neuroscience Abstract Viewer/Itinerary Planner*, 142.18.
- Heliot, R., Ganguly, K., Jimenez, J., & Carmena, J. M. (2010). Learning in closed-loop brain-machine interfaces: Modeling and experimental validation. *IEEE Trans. Syst. Man. Cybern. B Cybern.*, 40(5), 1387–1397.
- Jarosiewicz, B., Chase, S. M., Fraser, G. W., Velliste, M., Kass, R. E., & Schwartz, A. B. (2008). Functional network reorganization during learning in a brain-computer interface paradigm. *Proc. Natl. Acad. Sci. USA*, 105(49), 19486–19491.
- Jarzebowski, J., Srinivasan, L., & Coleman, T. P. (2008). Using stochastic control with data compression perspectives to enhance P300 neural communication prostheses. *Proc. IEEE Information Theory Workshop* (pp. 109–113). Piscataway, NJ: IEEE.
- Kemere, C., & Meng, T. H. (2005). Optimal estimation of feed-forward-controlled linear systems. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing* (Vol. 5, pp. 353–356). Piscataway, NJ: IEEE.
- Kim, S., & Coleman, T. P. (2011). Team decision theory and brain-machine interfaces. In *Proceedings of the 5th International IEEE EMBS Conference on Neural Engineering* (pp. 605–608). Piscataway, NJ: IEEE.
- Koyama, S., Chase, S. M., Whitford, A. S., Velliste, M., Schwartz, A. B., & Kass, R. E. (2010). Comparison of brain-computer interface decoding algorithms in open-loop and closed-loop control. *J. Comput. Neurosci.*, 29(1–2), 73–87.
- Ma, R., Aghasadeghi, N., Jarzebowski, J., Bretl, T., & Coleman, T. P. (2012). A stochastic control approach to optimally designing hierarchical flash sets in P300 communication prostheses. *IEEE Trans. Neural Syst. Rehabil. Eng.*, 20(1), 102–112. doi:10.1109/TNSRE.2011.2179560
- Moran, D. W., & Schwartz, A. B. (1999). Motor cortical representation of speed and direction during reaching. *J. Neurophysiol.*, 82(5), 2676–2692.

- Omar, C., Akce, A., Johnson, M., Bretl, T., Ma, R., Maclin, E., et al. (2011). A feedback information-theoretic approach to the design of brain-computer interfaces. *International Journal on Human-Computer Interaction*, 27(1), 5–23.
- Orsborn, A. L., Dangi, S., Moorman, H. G., & Carmena, J. M. (2012). Closed-loop decoder adaptation on intermediate time-scales facilitates rapid BMI performance improvements independent of decoder initialization conditions. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 20, 468–477.
- Richardson, A. G., Borghi, T., & Bizzi, E. (2012). Activity of the same motor cortex neurons during repeated experience with perturbed movement dynamics. *J. Neurophysiol.*, 107, 3144–3154. doi:10.1152/jn.00477.2011
- Scott, S. H. (2004). Optimal feedback control and the neural basis of volitional motor control. *Nat. Rev. Neurosci.*, 5(7), 532–546.
- Serruya, M. D., Hatsopoulos, N. G., Paninski, L., Fellows, M. R., & Donoghue, J. P. (2002). Instant neural control of a movement signal. *Nature*, 416(6877), 141–142.
- Shanechi, M. M., Wornell, G. W., Williams, Z., & Brown, E. N. (2010). A parallel point-process filter for estimation of goal-directed movements from neural signals. In *Proceedings of the IEEE Conference on Acoustics, Speech, and Signal Processing* (pp. 521–524). Piscataway, NJ: IEEE.
- Srinivasan, L., Eden, U. T., Willsky, A. S., & Brown, E. N. (2006). A state-space analysis for reconstruction of goal-directed movements using neural signals. *Neural Computation*, 18, 2465–2494.
- Stengel, R. F. (1994). *Optimal control and estimation*. New York: Dover.
- Taylor, D. M., Tillery, S. I., & Schwartz, A. B. (2002). Direct cortical control of 3D neuroprosthetic devices. *Science*, 296(5574), 1829–1832.
- Todorov, E. (2004). Optimality principles in sensorimotor control. *Nat. Neurosci.*, 7(9), 907–915.
- Todorov, E., & Jordan, M. I. (2002). Optimal feedback control as a theory of motor coordination. *Nat. Neurosci.*, 5(11), 1226–1235.
- Truccolo, W., Eden, U. T., Fellows, M. R., Donoghue, J. P., & Brown, E. N. (2005). A point process framework for relating neural spiking activity to spiking history, neural ensemble, and extrinsic covariate effects. *J. Neurophysiol.*, 93(2), 1074–1089.
- Vargas-Irwin, C. E., Shakhnarovich, G., Yadollahpour, P., Mislow, J. M., Black, M. J., & Donoghue, J. P. (2010). Decoding complete reach and grasp actions from local primary motor cortex populations. *J. Neurosci.*, 30(29), 9659–9669.
- Wahnoun, R., He, J., & Helms Tillery, S. I. (2006). Selection and parameterization of cortical neurons for neuroprosthetic control. *J. Neural Eng.*, 3(2), 162–171.
- Wang, J., Wagner, F., Borton, D. A., Zhang, J., Ozden, I., Burwell, R. D., et al. (2012). Integrated device for combined optical neuromodulation and electrical recording for chronic in vivo applications. *J. Neural Eng.*, 9(1), 016001. doi:10.1088/1741-2560/9/1/016001
- Wu, W., Gao, Y., Bienenstock, E., Donoghue, J. P., & Black, M. J. (2006). Bayesian population decoding of motor cortical activity using a Kalman filter. *Neural Comput.*, 18(1), 80–118.