A high-performance neural prosthesis incorporating discrete state selection with hidden Markov models

Jonathan C. Kao*, *Student Member, IEEE,* Paul Nuyujukian*, *Member, IEEE,* Stephen I. Ryu, *Member, IEEE*, Krishna V. Shenoy, *Senior Member, IEEE*

***Abstract*—Communication neural prostheses aim to restore efficient communication to people with motor neurological injury or disease by decoding neural activity into control signals. These control signals are both analog (e.g., the velocity of a computer mouse) and discrete (e.g., clicking an icon with a computer mouse) in nature. Effective, high-performing, and intuitive-to-use communication prostheses should be capable of decoding both analog and discrete state variables seamlessly. However, to date, the highest-performing autonomous communication prostheses rely on precise analog decoding, and typically do not incorporate high-performance discrete decoding. In this report, we incorporated a hidden Markov model (HMM) into an intracortical communication prosthesis to enable accurate and fast discrete state decoding in parallel with analog decoding. In closed-loop experiments with non-human primates implanted with multielectrode arrays, we demonstrate that incorporating an HMM into a neural prosthesis can increase state-of-the-art achieved bitrate by 13.9% and 4.2% in two monkeys ($p < 0.01$). We found that the transition model of the HMM is critical to achieving this performance increase. Further, we found that using an HMM resulted in the highest achieved peak performance we have ever observed for these monkeys, achieving peak bitrates of 6.5 bps, 5.7 bps, and 4.7 bps in Monkeys J, R, and L respectively. Finally, we found that this neural prosthesis was robustly controllable for the duration of entire experimental sessions. These results demonstrate that high-performance discrete decoding can be beneficially combined with analog decoding to achieve new state-of-the-art levels of performance.**

*JCK and PN contributed equally to this work. JCK, PN, SIR, and KVS are with the Electrical Engineering Department, PN and KVS are with the Bioengineering Department, PN is with the School of Medicine and Neurosurgery Department, and KVS is with the Neuobiology Department, the Neurosciences Program, and the Stanford Neurosciences Institute; Stanford University, Stanford, CA. SIR is with the Palo Alto Medical Foundation, Palo Alto, CA. KVS is also a Howard Hughes Medical Institute Investigator.

This work was supported by the National Science Foundation Graduate Research Fellowship (JCK); the Stanford Medical Scholars Program, Howard Hughes Medical Institute Medical Research Fellows Program, Paul and Daisy Soros Fellowship, Stanford Medical Scientist Training Program (PN); Christopher and Dana Reeve Paralysis Foundation (SIR and KVS); and the following to KVS: Burroughs Welcome Fund Career Awards in the Biomedical Sciences, Defense Advanced Research Projects Agency Revolutionizing Prosthetics 2009 N66001-06-C-8005 and Reorganization and Plasticity to Accelerate Injury Recovery N66001-10-C-2010, US National Institutes of Health National Institute of Neurological Disorders and Stroke Collaborative Research in Computational Neuroscience Grant R01-NS054283 and Bioengineering Research Grant R01-NS064318 and Transformative Research Award T-R01NS076460, and US National Institutes of Health EUREKA Award R01-NS066311 and Director's Pioneer Award 1DP1OD006409.

Manuscript received Aug 18, 2015.

I. Introduction

INTRACORTICAL neural prostheses, also known as brain-machine interfaces, decode spiking activity in motor cortex to drive prosthetic devices, such as computer cursors or robotic arms (e.g., [1]–[11]). The ability of these devices to restore efficient, high-performance communication is critical to their clinical viability (e.g., [12]–[14]). In the past fifteen years, substantial progress has been made to improve the performance of "continuous decoders," where analog variables, such as intended position and velocity, are decoded to drive a computer cursor or robotic arm (e.g., [5], [8], [11], [15]–[17]) . These decoders allow the user to move the prosthesis continuously through the workspace; for example, the user may use a continuous decoder to move a cursor on a computer screen, or make a reach with a robotic arm.

In addition to an analog component, everyday control tasks incorporate discrete actions, such as clicking an icon with a computer mouse. A neural prosthesis that decodes both analog and discrete control signals well would therefore give the user intuitive control over the neural prosthesis. Furthermore, decoding discrete state signals in parallel with analog continuous signals has the potential to increase the performance of neural prostheses. Consider the example of controlling a computer cursor to type on a virtual keyboard. To convey the intent to type one key out of many potentially selectable keys requires a selection mechanism. In previous studies achieving state-of-the-art communication rates using only a continuous decoder (ReFIT-KF), key selections were conveyed by holding a neurally-driven cursor still over the desired key for a certain amount of time (e.g., [15], [16]). However, this selection mechanism has limitations. First, it requires a mandatory hold time to communicate selection which decreases the overall selection rate. Second, it is prone to inadvertent selections if the user is not actively focused on controlling the cursor (e.g., as a result of being under other cognitive loads, such as contemplating what to write) or if the user moves the cursor too slowly and accidentally holds on an incorrect key. A potentially better selection mechanism is to have a discrete decoder, running in parallel with the continuous decoder, detect the user's intent to select a key. This approach circumvents the required hold time and would reduce the number of inadvertent selections that may arise from accidentally dwelling on an incorrect key while under other cognitive loads.

While it is intuitive that having a discrete decoder operating in parallel with a continuous decoder should increase the

performance and utility of a neural prosthesis, much like being able to click with a computer mouse greatly increases its utility, to date the highest bitrates (or information throughputs) achieved by communication prostheses have been implemented with purely continuous neural prostheses with a mandatory hold time [15], [16]. While previous studies have incorporated discrete decoders, the performance of these studies have not exceeded the performance of mandatory hold time systems. A previous study which combined a continuous decoder with a binary discrete decoder observed that on average in two human participants, it took 2.5s and 6.9s respectively for a discrete decoder to correctly click a desired target, and moreover, that approximately 45% and 65% of clicks did not occur when the continuous decoder was over the correct target [18]. Yet another study found that using linear discriminant analysis (LDA) to decode a discrete control signal resulted in acquisition rates on the order of 10 characters per minute [19]. A recent study reported that using such discrete decoders resulted in selection rates that were over $2\times$ longer than simply holding the target for 500 ms with a continuous decoder [20].

A major reason why quick and accurate discrete decoding is difficult is that neural observations are noisy on single trials. Although one solution to ameliorate noise is to integrate neural activity for longer amounts of time, this has the negative effect of slowing selection rate. Thus, a quick and accurate discrete state decoder must (1) integrate neural activity for only tens of milliseconds as opposed to hundreds of milliseconds (as in [6], [18]) and (2) be robust in the presence of substantial noise, not making frequent spurious transitions between discrete states. Furthermore, the capability of decoding an arbitrary number of desired discrete states, rather than just a binary state, may also substantially increase the performance and utility of the discrete decoder.

We propose addressing these challenges by using hidden Markov models (HMMs) as a framework for discrete decoding (e.g., [21]–[23]). In addition to its emissions model, which describes the probabilites of the neural observations given a discrete state, the HMM incorporates a transition model. The transition model describes the probabilities of transitioning between discrete states, providing a prior on the distribution of discrete states even before the observation of neural data. Informally, this imparts a sense of "continuity" to discrete states, mitigating the effect of observation noise and reducing spurious discrete state transitions. This is in contrast to discrete decoders such as naïve Bayes classifiers (e.g., [9], [10], [24], [25]), LDAs (e.g., [19], [26]–[28]), and support vector machines (e.g., [29], [30]), which do not incorporate a transition model and may therefore more often switch between states due to observation noise. As we show in this manuscript, it is also possible to design arbitrary probabilistic finite-state machines that govern the transitions between discrete states. These more complex transition models enable the HMM to decode an arbitrary number of discrete states.

We note that a prior offline study used an HMM to detect neural state transitions [23], but did not combine it with a continuous state-decoder, as proposed here. Further, we note other offline studies have combined continuous and discrete decoding (e.g., [31]–[34]) where a discrete decoder modulates the continuous decoder. Some studies have begun to explore these decoders online (e.g., [35]). These studies investigated how discrete decoders may be used to improve continuous decoders, rather than to execute discrete commands in a communication task. The goal of this work is to demonstrate a high-performance neural prosthesis for use in closed-loop systems, providing intuitive and accurate continuous and concurrent discrete control. To achieve high-performance discrete control, we designed and incorporated an HMM into a neural prosthesis. Specifically, we evaluated if it could be used beneficially with a state-of-the-art continuous decoder, the ReFIT-KF [5]. We evaluated performance with rhesus macaques in closed-loop experiments, and demonstrated that discrete selection with an HMM could increase communication rates of a neural prosthesis. Further, we demonstrate that the transition model of the HMM is critical to achieving high-performance discrete state selection.

II. Materials & Methods

A. Experimental setup and data acquisition

All surgical and animal care procedures were performed in accordance with National Institutes of Health guidelines and were approved by the Stanford University Institutional Animal Care and Use Committee. Experiments were conducted with three adult male rhesus macaques (J, R & L) implanted with 96 electrode Utah arrays (Blackrock Microsystems Inc., Salt Lake City, UT) using standard neurosurgical techniques. Electrode arrays were implanted in dorsal premotor cortex (PMd) and primary motor cortex (M1) as visually estimated from local anatomical landmarks. Monkeys J & R had two arrays, one in M1 and one in PMd, while Monkey L had one array implanted on the M1/PMd border.

The monkeys made point-to-point reaches in a 2D plane with a virtual cursor controlled by the contralateral arm or by a neural prosthetic decoder. This task has previously been described in prior work (e.g., [5], [15], [16], [36], [37]). The virtual cursor and targets were presented in a 3D environment (MSMS, MDDF, USC, Los Angeles, CA) [38]. Hand position data were measured with an infrared reflective bead tracking system (Polaris, Northern Digital, Ontario, Canada). Spike counts were collected by applying a single negative threshold, set to $-4.5\times$ root-mean-square of the spike voltage per neural channel. Behavioral control and neural decode were run on separate PCs using the Simulink/xPC platform (Mathworks, Natick, MA) with communication latencies of 3 ms. This enabled millisecond timing precision for all computations. Neural data were initially processed by the Cerebus recording system (Blackrock Microsystems Inc., Salt Lake City, UT) and were available to the behavioral control system within 5 ms $\pm$ 1 ms. Visual presentation was provided via two LCD monitors with refresh rates at 120 Hz, yielding frame updates of 7 ms $\pm$ 4 ms. Two mirrors visually fused the displays into a single three-dimensional percept for the user, creating a Wheatstone stereograph [36], but all tasks were limited to 2D. This setup is illustrated in Fig 1a.

For continuous decoding, we used the ReFIT-KF decode algorithm [5], [15], which was used in the highest-reported

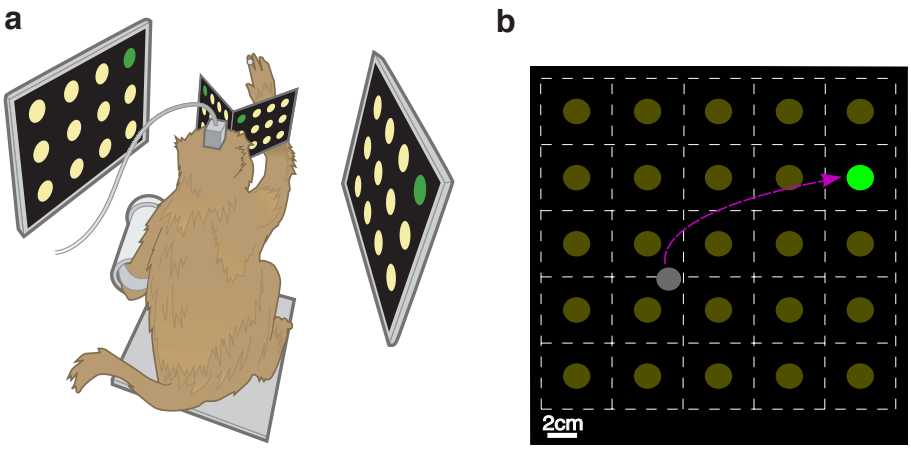

Fig. 1. Rig setup and task, adapted from [15], [16]. **a** Diagram of monkey performing reaches in a virtual task environment. **b** Diagram of the grid task with 25 (5 × 5) array of targets. The white dashed lines denote the acceptance windows around each target, but these were not visible to the monkey. On each trial, one target was prompted (in green). The monkey selected a target by moving the cursor (grey) and triggering a "click" with the HMM. When the HMM triggered a click, whichever target acceptance window the cursor was in would be selected, so that any click occurring outside the acceptance window of the green target was an incorrect selection.

communication rate neural prosthesis to date [16]. We trained the ReFIT-KF using the center-out-and-back task according to a protocol previously described in [5]. The observations for the decoder were binned threshold crossings counted in non-overlapping 10 ms bins for Monkey J, 15 ms bins for Monkey R and 50 ms bins for Monkey L. In all experiments, the monkeys were free to move their contralateral arm, consistent with previous studies [5], [11], [15]–[17], [39]. We chose this animal model because we believe it most closely mimics the neural state of a human subject who would be employing a neural prosthesis in a clinical study [40]. This model is limited in that proprioceptive feedback is present in the neural activity [40], [41]. However, we favor this model over a restrained arm model where the monkey would generate neural activity that presumably largely resides in a nullspace of cortical activity [42]. The animal model we employ recognizes that a human subject using a neural prosthesis could generate neural activity that would have been capable of driving muscles.

B. Grid task

We used the grid task, described in [15] and illustrated in Fig 1b, to evaluate the performance of the neural prosthesis. In the grid task, an array of mutually exclusive targets tile a 24 cm × 24 cm workspace. Every target is selectable at every point in time; if the monkey clicks while on a correct (incorrect) target, then a correct (incorrect) selection will have been made. We enforced a 200 ms lock-out period following target selection during which no target could be selected to account for the reaction time of the monkey [15]. As any target can be selected throughout the course of a trial, selecting one correct target uniformly chosen from N targets conveys $\log_2(N-1)$ bits of information. The factor $N-1$ arises due to our assumtion that one taraget is reserved as a backspace key. The parameters of the grid task were different for each monkey and experiment. In direct performance comparisons, we chose the grid size to favor the ReFIT-KF by using previously reported grid parameters that are optimal for ReFIT-KF [15]. We note that the study by Nuyujukian and colleagues [15] did not include the 6×6 grid configuration, which we found led to higher bitrates than the 5×5 grid configuration in Monkey J. Therefore, we used a 6×6 grid for Monkey J and a 5×5 grid for Monkey R. In long-run performance evaluations of the HMM, we chose the grid size for which the HMM achieved highest performance, which was a 7×7 grid for Monkey J and a 5×5 grid for Monkeys R and L. For Monkey L, we were only able to perform long-run performance evaluations of the HMM. When we attempted direct performance comparisons with Monkey L, we found that his behavior had unfortunately declined over time due to age and other factors. The 5×5 grid was composed of targets with square acceptance windows of length 4.8 cm, the 6×6 grid of length 4 cm, and the 7×7 grid of length 3.45 cm.

To quantify performance, we evaluated the achieved bitrate of the decoder, which has previously been described in [15]. Briefly, we calculated achieved bitrate conservatively by assuming that every incorrect selection had to be compensated by a correct selection (much like incorrectly selecting a key on a keyboard requires hitting the backspace key). If in T seconds, c correct selections were made while ℓ incorrect selections were made on a grid with N targets, then the achieved bitrate is:

$$I = \frac{(c-\ell)\log_2(N-1)}{T}, \text{if } c \geq \ell \qquad (1)$$

and 0 if $\ell < c$, i.e., if the monkey performs the task at or less than 50% success rate.

C. HMM training

The HMM, designed for a closed-loop neural prosthesis, is the major contribution of this report. As described in the introduction, we chose the HMM as a potential discrete decoder that would be able to achieve high-performance discrete state decoding by (1) integrating neural activity over short periods of time, specifically tens of milliseconds, and (2) being robust in the presence of noisy single-trial neural activity so as to not making frequent spurious transitions between discrete states.

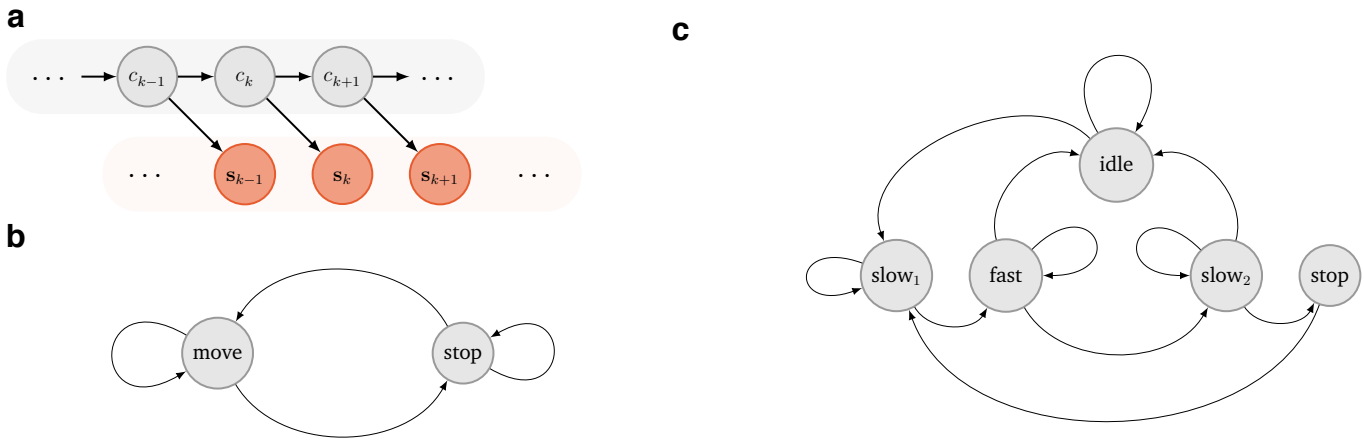

Fig. 2. Graphical representation of the hidden Markov model (HMM) and transition models. **a** Graphical model of the HMM, where a discrete random variable, c_k, evolves over time through a transition matrix. The output of the HMM at each point is the projected neural activity, $\mathbf{s}_k$. **b** The transition model used in online experiments for Monkey J and R incorporated two states, "move" and "stop." **c** The transition model used in online experiments for Monkey L incorporated an "idle" state and partitioned the movement phase into a sequence of "slow" to "fast" to "slow" transitions.

A graphical representation of the HMM is shown in Fig 2a. Neural observations at time k, denoted by $\mathbf{s}_k$, are used to infer the distribution of a probabilistic discrete state variable, c_k, which evolves over time. We denote the collection of discrete states as $\mathcal{C}$. The distribution of c_k, which we also call the probability vector, describes the probability of being in each discrete state in $\mathcal{C}$. The neural observation $\mathbf{s}_k$ is a 5-dimensional projection of the observed neural binned spike counts found via principal components analysis (PCA) as in previous reports (e.g., [43], [44]). As optimal parameters from offline simulations may differ from those in closed-loop control (e.g., [36], [45], [46]), we performed a cursory optimization of performance as a function of the number of PCs. We determined that using the top 5 PCs led to adequate control, although it is possible that a different number of PCs may result in even higher performance. The HMM uses the distribution on c_{k-1}, and the current neural observation, $\mathbf{s}_k$, to compute the distribution of c_k. This is graphically illustrated by Fig 2a. This computation involves two components.

The first component is a time-invariant probability transition model describing the probability of transitioning between states, i.e., $p_{c,c'} = \mathbf{Pr}(c_k = c | c_{k-1} = c')$ for all discrete states $c, c' \in \mathcal{C}$. This transition model gives a prior estimate of the discrete state probability vector in the absence of observations. For Monkeys J and R, we designed a straightforward transition model comprising two states, "move" and "stop," as shown in Fig 2b. For Monkey L, we designed a more complex model comprising five states, as shown in Fig 2c.

The second component is a Gaussian emissions model which approximates the distribution of the neural observations $\mathbf{s}_k$ when in state c. The emissions model is parameterized by the mean and covariance matrix of the neural observations in each state, i.e., $(\mathbf{s}_k | c_k = c) \sim \mathcal{N}(\mu_c, \Sigma_c)$. The transition model and the emissions model are combined to estimate the probability of being in each state.

The parameters of the HMM were learned in a supervised fashion from experimental training data during which the monkey controlled a virtual cursor to move and hold over targets for 500 ms. In the direct performance comparisons for Monkeys J and R, the training set virtual cursor was controlled by the monkey's hand. In the long-run performance evaluations, the training set virtual cursor was controlled by a neural prosthesis for Monkeys J and L. We found that the HMM was capable of achieving high-performance in both scenarios.

We assigned every time bin during the training set as being in a discrete state, $c \in \mathcal{C}$. We defined the "stop" state as the period of time 250 ms (133 ms) into the hold epoch for hand training sets (neural prosthesis training sets) until 100 ms after hold completion. We chose these boundaries to both account for the reaction time of the monkey and cursorily optimize closed-loop performance. It is possible that these boundaries could be further optimized to achieve higher performance. For Monkeys J and R, the rest of the trial was classified as the "move" state, resulting in the transition model shown in Fig 2b. For Monkey L, we split the "move" state into "slow" and "fast" states based on a speed threshold. For any given trial, times when the cursor speed was below 25% of the maximum cursor speed of that trial were assigned to be in the "slow" state while times when cursor speed exceeded 25% of the maximum speed were assigned to be in the "fast" state. We found that this design increased closed-loop performance in Monkey L by preventing early transitions into the "stop" state. Behaviorally Monkey L was also prone to idle during the task due to inconsistent motivation, and so we incorporated an "idle" state corresponding to when Monkey L sat idly. The transition model for Monkey L is shown in Fig 2c.

After binning the neural activity and state sequence per trial, we learned the transition matrix and emissions process for the HMM. The transition matrix was learned by calculating the proportion of transitions between potential states in the training set. For example, in the transition model of Fig 2b,

we calculated the proportion of transitions from "move" to "move", "move" to "stop," "stop" to "move," and "stop" to "stop." These values comprised the transition matrix. For the idle state in Monkey L, we chose the transition probability into the "idle" state to be approximately 10% of the probability of transitioning out of the state. To learn the emissions model, we aggregated all bins of projected neural activity corresponding to a certain discrete state. We then computed the empirical mean and covariance of the projected neural activity, which were treated as the emissions mean and emissions covariance for that discrete state. We note that in Monkey L, the states "slow$_1$" and "slow$_2$" (shown in Fig 2c) had the same emissions model.

To decode the probability of each discrete state at time k, we used the forward algorithm (e.g., [21], [22]). We kept track of the probability $\alpha_k(c) = p(\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_k, c_k = c)$, which is the joint probability of observing all neural observations and the discrete state value of $c_k = c$. Calculating $\alpha_k(c)$ for all $c \in \mathcal{C}$ yields our probability vector over the discrete states at time k. Using the chain-rule for probability and the graph structure of the HMM shown in Fig 2a, we can derive a forward recursion for $\alpha_k(c)$ as

$$\alpha_k(c) \propto f(\mathbf{s}_k | c_k = c) \sum_{c' \in \mathcal{C}} \alpha_{k-1}(c') p_{c,c'}. \tag{2}$$

Here, $f(\mathbf{s}_k | c_k = c)$ is the Gaussian density over the neural observations. The α's were re-normalized at every time-step. We note that using the forward algorithm is in contrast to a prior closed-loop study that incorporated a transition model into discrete state selection. Specifically, in the closed-loop experiments of [6], [18], the discrete decoder assumed that c_k was deterministically in the most likely state in $\mathcal{C}$. This approach does not propagate the probability vector forward in time and loses state distribution information (i.e., how likely each state was) at each time step.

D. Offline simulations

In all reported offline results, we decoded neural data recorded from datasets where each monkey performed reaches on a center-out-and-back task with 8 peripheral targets spaced 8 cm from the center target [5], [11]. During these reaches, we tagged each bin as either in the "move" or the "stop" state as previously described in HMM training. Bins were classified as "stop" if the probability of being in the "stop" state exceeded a threshold, t_{stop}, and in the "move" state otherwise. (We note that for offline analyses only, we used the simple two-state transition model shown in Fig 2b for Monkey L.) When offline decode error is reported, it is the proportion of incorrect classifications at bin resolution. Finally, all offline performance analyses were performed on withheld testing data.

E. Closed-loop experiments

In closed-loop experiments, the monkeys controlled the ReFIT-KF and HMM in parallel. The ReFIT-KF controlled the velocity of the cursor, while the HMM indicated whether or not to select a target. We call this decoder the "ReFIT-KF + HMM." To be conservative in detecting the selection of a target (which we call a "click"), we enforced that a click only occur after the probability of being in the "stop" state exceeded $t_{\text{stop}} = 0.8$ for at least two consecutive time bins. Following a click, we ended the trial and initiated a new trial where, after a 40 ms pause, a new target was presented for the monkey to acquire. Consistent with how a computer mouse, after receiving an input such as a finger press to indicate a click, will reset to the un-clicked state in the absence of continuous input, we set the probability of being in the "move" ("slow$_1$") state for Monkeys J and R (Monkey L) to be 1 following any target selection. In this fashion, the algorithm performs the analogous and automatic "un-click" of a computer mouse. We note that indicating a continuous hold (e.g., as one might hold their finger down continuously on a computer mouse) may be encoded as a separate state in the HMM, and may be explored in future work.

When comparing the performance of the ReFIT-KF + HMM to the ReFIT-KF with a hold time selection mechanism, we used a hold time of 450 ms, as found from previous optimizations [15]. This hold time is chosen with the intent to maximize the performance of the ReFIT-KF. We note that, as reported by Nuyujukian and colleagues [15], a shorter hold time that allows for quicker selections ultimately results in a lower achieved bitrate due to an increase in incorrect selections (e.g., inadvertently dwelling on an incorrect target en route to the desired target).

We performed three closed-loop experiments. The first two were performed in Monkeys J and R only, as Monkey L's behavior had unfortunately declined over time due to aging and other factors. The third experiment was performed by Monkeys J, R and L.

In the first experiment, we sought to compare the performance of the ReFIT-KF to the ReFIT-KF + HMM on the grid task. Within an experimental day, the monkey controlled the ReFIT-KF for 200 trials, followed by the ReFIT-KF + HMM for 200 trials. The 200 trials were then used to compute a bitrate for each decoder. These decoders were repeatedly tested after each other in an A-B-A-B-A-... fashion, yielding repeated within-day measurements of ReFIT-KF performance and ReFIT-KF + HMM performance. We call one A-B segment (i.e., 200 trials of ReFIT-KF followed by 200 trials of ReFIT-KF + HMM) an experimental block. We always evaluated the ReFIT-KF first in the block, so that any benefits from the HMM were not a result of degrading motivation. Bitrates were paired in each block for statistical testing. We also note that the parameters of the grid were chosen to maximize the performance of the ReFIT-KF [15] and not the ReFIT-KF + HMM; as shown in the third experiment, it was possible to choose the grid density to achieve even higher performance with the ReFIT-KF + HMM.

In the second experiment, we sought to determine the importance of using a transition model in discrete state selection. We built a discrete decoder, termed the quadratic discriminant (QD), which performs classification using only the emissions process of the HMM. Therefore, the only difference between the QD and the HMM is that the HMM incorporates a transition model. The two decoders have the same parameters relating the

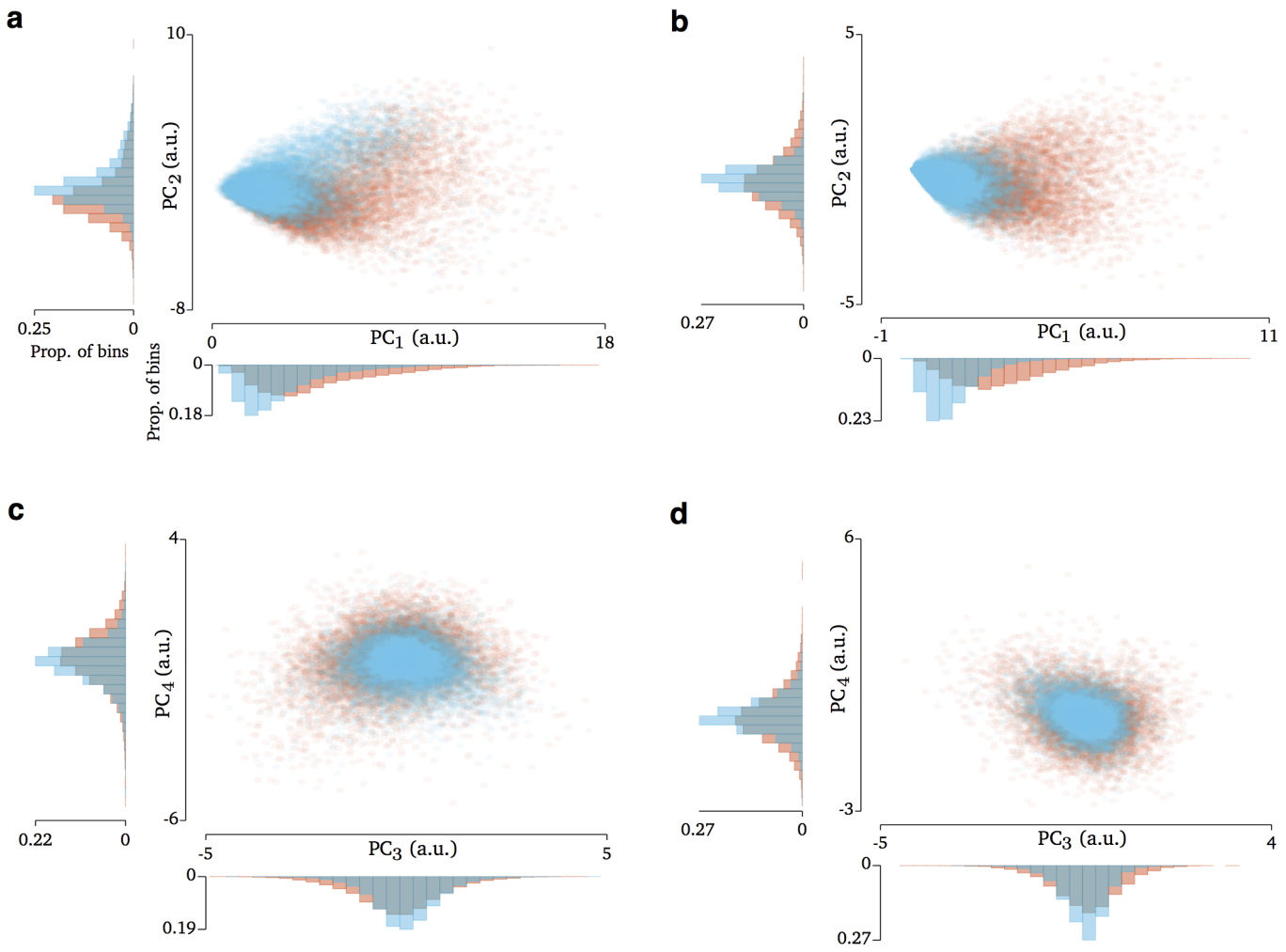

Fig. 3. Low-dimensional projections of neural activity. **a** The neural activity is projected into PC1 and PC2 for Monkey J. Each dot denotes the projected neural activity in a bin. Orange corresponds to neural activity while in the "stop" state while blue corresponds to neural activity while in the "move" state. Histograms of the activity in each PC are also shown. We note that the PCs are not zero-centered since we applied the projector matrix to the non-mean centered neural activity. This was to reduce the number of parameters in the model, as the emissions model incorporates the mean of the neural activity. **b** Same as **a** but for Monkey R. **c** Same as **a** but for PCs 3 and 4 in Monkey J. **d** Same as **a** but for PCs 3 and 4 in Monkey R.

neural observations to the discrete state. If the performance of the two decoders are not significantly different, this indicates that the HMM achieves good performance due to its emissions process rather than the transition model; however, if the HMM performs significantly better than the QD, this indicates that the transition model is crucial to achieving high performance discrete state classification. We compared the performance of the ReFIT-KF + QD to the ReFIT-KF + HMM on the grid task in the same fashion as the first experiment, conducting within-day comparisons where each decoder was evaluated in an experimental block for 200 trials to measure an achieved bitrate.

In the third experiment, we allowed the ReFIT-KF + HMM to be controlled for hours-long experimental sessions, demonstrating that the HMM is robust across an experimental session.

III. Results

The results are organized into three sections. First, we present neural data and offline simulations that characterize the HMM. Second, we present the results of closed-loop experiments demonstrating that the HMM can increase neural prosthetic performance and performs better than an equivalent classifier with no transition model. Third, we present results where the monkeys controlled the ReFIT-KF + HMM for entire experimental sessions, demonstrating that high-performance can be sustained.

A. Offline decode using principal components of the neural activity

The emissions process of the HMM models a multivariate distribution on the neural activity. To avoid the "curse of dimensionality" [21] we reduced the dimensionality of the neural activity with PCA, especially as the dimensionality of motor cortex during simple reaching tasks is on the order of 10-20 (e.g., [47], [48]). We found that a substantial proportion

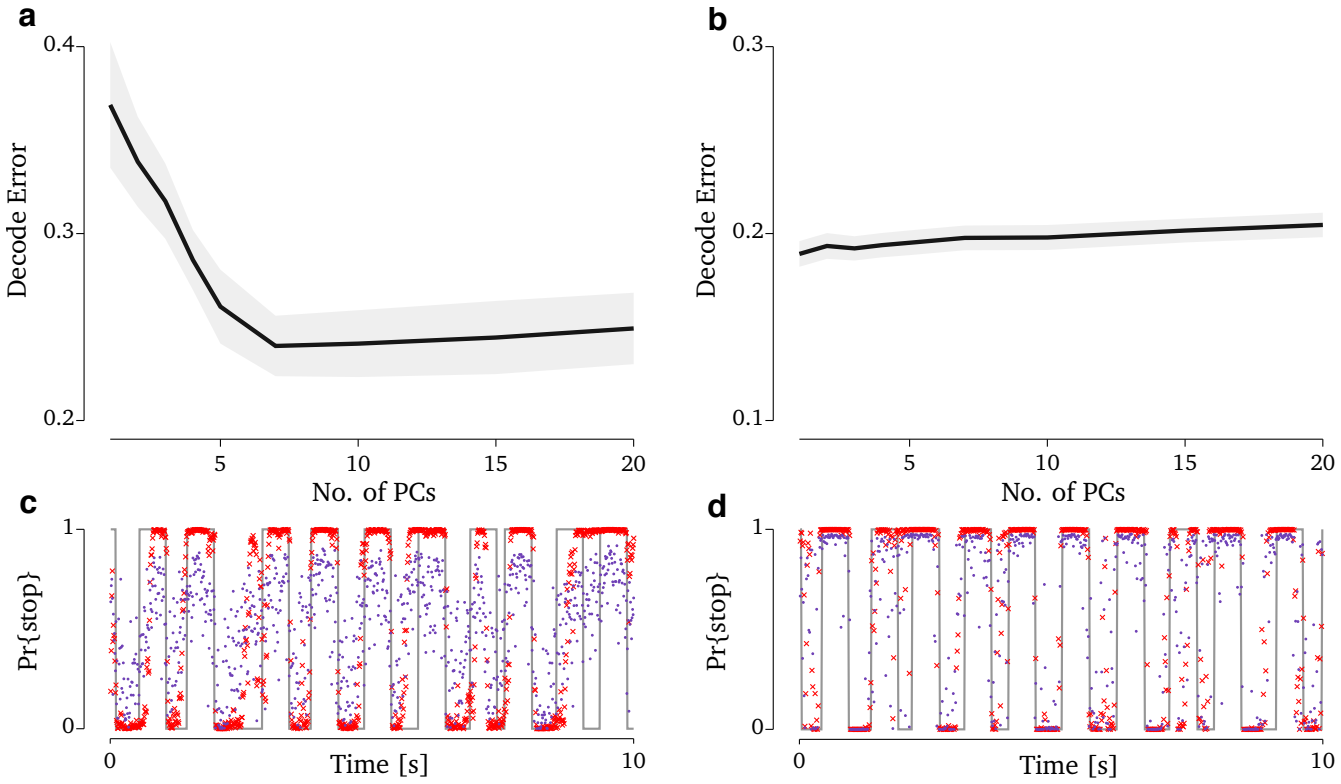

Fig. 4. Offline decode using the HMM. **a** Offline decode error for Monkey J, measured as the proportion of incorrectly predicted states across all time bins. We note that after 5-10 PCs, performance did not increase with more PCs, indicating that a majority of useful variance associated with differentiating between "move" and "stop" are captured by the top principal components of the neural activity. Shading denotes the standard error of the mean. **b** Same as **a** but for Monkey R. Monkey R's results suggest that a majority of the useful variance for decoding "move" versus "stop" is contained in PC 1. **c** Offline decode in Monkey J for hand trials. The gray line represents whether the monkey was in the "move" or "stop" state. The red markers denote the decoded probability of being in the "stop" state per bin using the HMM. The purple marker denotes the probability of being in the "stop" state per bin using the QD classifier. **d** Same as **c** but for Monkey R.

of variance related to the "move" state versus the "stop" state was captured by the leading principal components of the neural activity. Fig 3a-d shows a scatter plot of neural activity during reaching when projected onto the first four principal components (PCs) of the neural activity for Monkeys J and R (for Monkey L, see Supp Fig 1a,b). We note that even in PC 1 the distributions of the "move" and "stop" neural activity, though highly overlapping, are distinguishable. As shown in Fig 4a,b and Supp Fig 1c, we observed that increasing the number of PCs increases offline performance until about 5-10 PCs are used (Monkeys J and L) or did not significantly increase offline performance (Monkey R). This suggests that variance related to discriminating "move" vs "stop" states is largely contained within the top 10 principal components of the data.

We note that although the distributions of neural activity during the "move" and "stop" states are distinguishable, they have a high degree of overlap. This high degree of overlap may be a major reason why achieving high-performance discrete decoding is not straightforward, as discriminating only based on the likelihood of the noisy observations may be unreliable. We tested this idea by comparing the performance of the HMM to the performance of the QD, which is a classifier using only the emissions process of the HMM.

We evaluated the extent to which the transition model of the HMM can help increase discrete state selection accuracy. For 5, 11, and 4 experimental days in Monkeys J, R, and L respectively, we performed an offline decode of 200 cross-validation trials per experimental day using the HMM and QD decoders. Across all of these trials, we obtained a distribution of discrete state transition times per trial, as well as a daily classification error rate for each decoder. We found that the HMM decodes were more accurate than the QD decodes, as shown in Fig 4c,d and Supp Fig 1d, better tracking the true state. The decode error (percentage of bins incorrectly classified) for the HMM was 24%, 20%, and 19% for Monkeys J, R and L respectively while for the QD, the errors were 46%, 24%, and 21%. Therefore, the HMM decoder more accurately decoded the discrete state ($p < 0.01$ for Monkeys J and R, $p = 0.011$ for Monkey L, paired t-test on each experimental day's classification error). The time-series of the offline decode in Fig 4c,d and Supp Fig 1d suggest that the transition model effectively denoises ambiguous neural activity whose distributions are not strong enough to cause transitions in the discrete state. This is supported by our finding that in general, the transition probabilities between states were small, i.e., $p_{\text{move,stop}} < 0.05$ and $p_{\text{stop,move}} < 0.05$, indicating that there was strong inertia to remain in the same state.

We also evaluated the time it took to correctly transition from the "move" to "stop" state to determine if the HMM

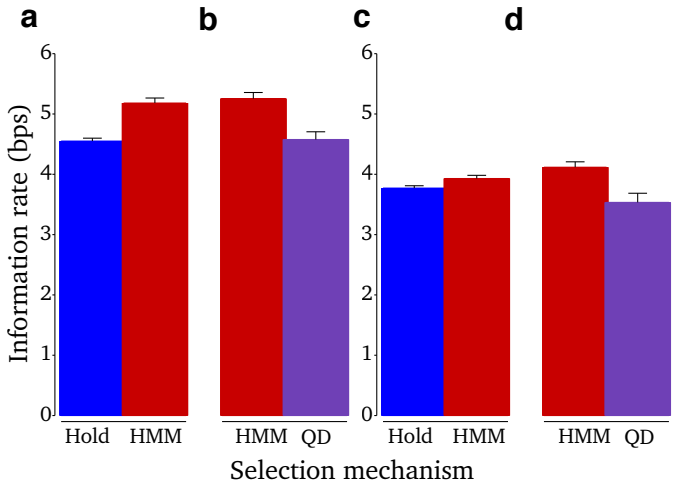

Fig. 5. Direct comparison of ReFIT-KF and ReFIT-KF + QD vs. ReFIT-KF + HMM. **a** In Monkey J, we compared the performance of the ReFIT-KF (using a mandatory hold selection mechanism, blue) vs the ReFIT-KF + HMM (red) in within-day blocks. Error bars denote the standard error of the mean. The ReFIT-KF + HMM significantly achieved higher performance than the ReFIT-KF alone. Datasets **J_2015-12-03**, **J_2015-12-04**, and **J_2015-12-07** comprising $7,075$ total trials and 17 comparison blocks. **b** In Monkey J, we compared the performance of the ReFIT-KF + HMM vs the ReFIT-KF + QD. The ReFIT-KF + HMM significantly achieved higher performance than the ReFIT-KF + QD, indicating that the transition model of the HMM is important for achieving high performance discrete state selection. Datasets **J_2015-12-08** and **J_2015-12-09** comprising $4,803$ total trials and 12 comparison blocks. **c** Same as (a) but for Monkey R. Datasets **R_2015-12-14**, **R_2015-12-15**, **R_2015-12-16**, and **R_2015-12-17** comprising $6,117$ total trials and 15 comparison blocks. **d** Same as (b) but for Monkey R. Datasets **R_2015-12-16** and **R_2015-12-17** comprising $4,152$ total trials and 10 commparison blocks.

could transition as quickly as an QD decoder. Importantly, this transition time will be a function of the probability threshold, t_{stop}, used to detect the "stop" state, as shown in the offline simulations of Supp Fig 3. At lower thresholds, the QD transitions more quickly than the HMM, as may be expected due to the effective "momentum" imparted by the HMM transition model. However, offline decode performance is also poorer at lower thresholds. Opposedly, we found that at high enough thresholds, the HMM transitions more quickly than the QD because the transition model enables the HMM to more quickly achieve high confidence in a certain state. In particular, we found that at the probability threshold used for closed-loop experiments ($t_{\text{stop}} = 0.8$), the HMM transitioned 159 ms and 23 ms more quickly than the QD model in Monkeys J and L ($p < 0.01$, two-sample t-test across transition times for every trial across all experimental days) while for Monkey R, the transition times were not significantly different ($p = 0.81$). These results suggest that at thresholds useful for closed-loop control, the HMM transition model does not slow transition time.

B. Performance comparisons of ReFIT-KF and ReFIT-KF + QD vs ReFIT-KF + HMM

Quantifying neural prosthetic performance in closed-loop experiments is crucial for assessing clinical viability and utility, as closed-loop results may differ from offline simulation [36], [45]. To evaluate the utility and performance of the HMM in a closed-loop neural prosthesis system, we performed experiments where monkeys controlled the HMM in parallel with a continuous decoder. In this section, we present two closed-loop experiments. First, to test if incorporating the HMM into a neural prosthesis could improve state-of-the-art performance, we compared the performance of the ReFIT-KF using a mandatory hold selection mechanism to the ReFIT-KF + HMM. Second, to test the importance of the transition model of the HMM, we compared the performance of the ReFIT-KF + QD to the ReFIT-KF + HMM.

We found that the performance of the ReFIT-KF + HMM was significantly higher than that of the ReFIT-KF using a mandatory hold selection mechanism. We evaluated the performance of the ReFIT-KF and ReFIT-KF + HMM in a blocked fashion (see Methods). We repeated these experiments across three experimental days in Monkey J (comprising $7,075$ trials) and four experimental days in Monkey R (comprising $6,117$ trials). We found that the ReFIT-KF + HMM increased the achieved bitrate of the ReFIT-KF by 13.9% and 4.2% in Monkeys J and R ($p < 0.01$, paired t-test of bitrates of all experimental blocks; also significant under Wilcoxon signed-rank test) as shown in Fig 5a,c. These results demonstrate that incorporating an HMM into a state-of-the-art decoder can significantly increase neural prosthetic performance, providing intuitive, fast, and accurate discrete state selection.

We also sought to understand the importance of the Markovian transition model of the HMM in achieving this performance increase. To this end, we evaluated the performance of the ReFIT-KF + QD, which is a quadratic discriminator using the emissions process of the HMM but crucially has no transition model. This performance comparison would therefore quantify the benefit of incorporating a transition model into a discrete state classifier. We performed experiments in the same blocked-fashion as for the ReFIT-KF vs. ReFIT-KF + HMM experiment (see Methods). We repeated these experiments across two experimental days in Monkey J (comprising $4,803$ trials) and two experimental days in Monkey R (comprising $4,152$ trials). We found that the ReFIT-KF + HMM achieved a higher bitrate than the ReFIT-KF + QD by 14.8% and 16.5% in Monkeys J and R ($p < 0.01$, paired t-test of bitrates of all experimental blocks; also significant under Wilcoxon signed-rank test) as shown in Fig 5b,d. These results demonstrate that the transition model of the HMM is crucial to achieving high performance. Indeed, the ReFIT-KF + QD achieved mean bitrates that were either comparable to (Monkey J) or worse (Monkey R) than simply using a mandatory hold time selection. This result is consistent with a report that, in humans, discrete state classification with an LDA (having no transition model) was slower than selection than with a mandatory hold time [20].

Together, these results indicate that an HMM provides fast and accurate discrete state control, and can be used to improve state-of-the-art continuous decoders. Further, these results demonstrate that the HMM transition model is a crucial component for achieving this performance improvement.

C. Long run performance of a ReFIT-KF + HMM

We further evaluated if the monkeys could control the HMM for entire experimental sessions. Across five experimental

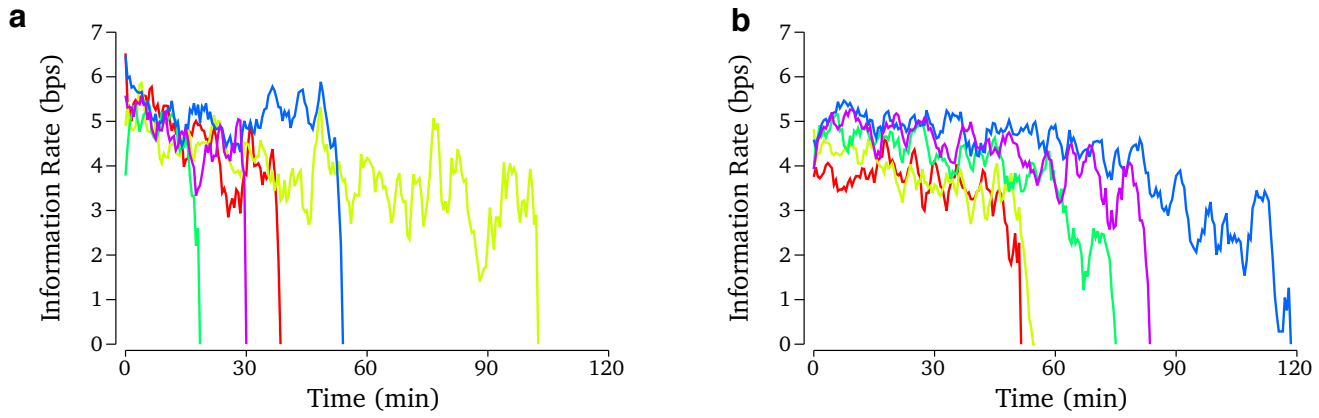

Fig. 6. Long-run performance of the ReFIT-KF + HMM. **a** Performance of the ReFIT-KF + HMM over entire experimental sessions for Monkey J. Each trace corresponds to one experimental day. The monkey sustained performance until he lost interest in the task, a behavior also observed when the monkey reaches with his native arm [15]. Datasets **J_2012-03-26**, **J_2012-03-27**, **J_2012-03-30**, **J_2012-04-05**, and **J_2012-04-06** comprising $18,825$ total trials. **b** Same as **a** but for Monkey R. Datasets **R_2015-12-20**, **R_2016-01-04**, **R_2016-01-05**, **R_2016-01-06** and **R_2016-01-07** comprising $29,406$ trials.

days in Monkey J (comprising $18,825$ total trials), five experimental days in Monkey R (comprising $29,406$ trials) and four experimental days in Monkey L (comprising $18,791$ total trials), we evaluated the long-run performance of the ReFIT-KF + HMM. As shown in Fig 6 and Supp Fig 2, we found that Monkeys J, R, and L were able to control the ReFIT-KF + HMM for entire experimental sessions. The ReFIT-KF + HMM was capable of being controlled at state-of-the-art levels of performance through the duration of the task, although performance sometimes declined over the course of the session. As further discussed below, this reduction in performance is potentially due to a decline in behavior resulting from fatigue. In addition to this, the monkeys performed the task until they lost motivation, reflected by a sharp drop in performance at the end of the experimental session. This sharp drop in performance is also observed when the monkey controls a cursor with his arm or with the ReFIT-KF alone [15] and does not reflect a drop in performance of the HMM.

During these sessions, we measured the *peak* achieved bitrate of the ReFIT-KF + HMM. We defined the peak bitrate as the maximum achieved bitrate that was sustainable for 60 seconds. This number is not representative of the performance of the decoder on average, but characterizes the upper limits of decoder performance. A decoder that is able to achieve a higher peak bitrate has the capacity to achieve higher communication rates.

For Monkeys J, R, and L, we found that the ReFIT-KF + HMM achieved peak bitrates of 6.49 bps, 5.71 bps, and 4.74 bps respectively. Across years-worth of experimental sessions we performed with Monkeys J, R, and L, we never observed the ReFIT-KF achieve a higher peak bitrate than ReFIT-KF + HMM. Specifically, across $899,024$ trials in Monkey J (spanning 298 experimental sessions across 5.2 years), $26,449$ trials in Monkey R (spanning 22 experimental sessionsa cross 3.3 years), and $386,563$ trials in Monkey L (spanning 158 experimental sessions across 5.3 years), the highest bitrates achieved by the ReFIT-KF (or a slightly modified variant of the ReFIT-KF) were 5.27 bps, 4.43 bps, and 4.45 bps. Thus, the peak bitrate achieved by ReFIT-KF + HMM in the course of a few experimental sessions exceeds the highest achieved peak performance of the ReFIT-KF across years of experiments.

We also note that, towards the end of long-run experiments, the performance of the ReFIT-KF + HMM tended to degrade. This was likely due to fatigue, as the monkey had to sustain higher selection rates and thus, higher average neural prosthesis velocities than with a ReFIT-KF having a mandatory hold time. Towards the end of the session, the monkey did not as reliably "dial-in" to the target, which led to incorrect clicks on adjacent targets. We did not observe such levels of degradation in performance in the direct within-day comparisons, where the monkeys were allowed brief pauses while the decoders were changed. Hence, the average performance of long-run sessions may be confounded by fatigue and as such, do not adequately reflect the average performance of the ReFIT-KF + HMM during normal patient use, where a patient would be capable of taking breaks as desired.

Supp Movies 1, 2 and 3 demonstrate near peak performance of the ReFIT-KF + HMM decoder for Monkeys J, R, and L respectively, while Supp Movies 4, 5 and 6 demonstrate performance approximately an hour into the experimental session.

IV. Discussion

Many communications and motor tasks incorporate discrete state selection (e.g., clicking an icon with a computer cursor). Achieving high-performance discrete state control therefore has the potential to considerably increase both the ease-of-use and performance of neural prostheses. Here we demonstrated that discrete decoding using an HMM could increase the performance of a neural prosthesis on a virtual keyboard communication task. This study represents the highest-reported peak and average bitrate achieved on a communication task of any neural prosthesis under any recording modality in non-human primates. Further, we demonstrated that the HMM is robust and can be used for entire experimental sessions. Together, these results demonstrate that high-performance parallel continuous and discrete decoding is possible. This

should enable the high-performance use of intuitive devices incorporating both analog and discrete components, such as a computer mouse.

An important component in improving the speed and accuracy of discrete decoding is the use of a transition model. Although neural activity is very noisy on single trials at relatively small bin widths, the incorporation of a transition model increased decoding accuracy in both offline (Fig 4c,d and Supp Fig 1d) and online comparisons (Fig 5b,d). This is likely because the transition model, which uses prior information about the frequency of state-transitions, has a tendency to stay in the same state until strong evidence from the emissions process causes a state transition. A potential drawback of using a transition model, however, is that it may provide too much "inertia" to the discrete state, causing discrete transitions to occur too slowly. We observed that this was not the case for appropriate thresholds, where correct state transitions in an HMM were as quick, if not quicker than state transitions in a QD.

We also found that the transition model could be modified to accomodate differing behaviors or to potentially increase performance. Importantly in the HMM, the transition model could be flexibly designed to incorporate an arbitrary number of discrete states and their transition rules. In Monkey L, who had poorer quality arrays and motivation, we were able to successfully decode an "idle" state when the monkey was not engaged in performing the task (Fig 2c). Further, we employed a transition model that required the monkey to transition through three phases of a reach, move "slow" to move "fast" and then again to move "slow" before entering the "stop" state. As HMMs are flexible in design, it is possible that our results could be further improved by optimizing the discrete state transition model.

We note that a previous online study [18] did also use a transition model, which took into account the probability $\mathbf{Pr}(c_k = c|c_{k-1} = c')$. However, this study did not propagate the probability vector forward in time. That is, at each time point k, they chose c_k to be deterministically in one state. Therefore, in calculating the distribution of c_{k+1}, the only information used from the previous time step is the most likely state of c_k rather than the entire distribution of c_k. We note that, in human clinical trials, this approach to discrete decoding was at least two times slower than selecting a target with a mandatory hold time [20]. In our study, we decoded with the forward algorithm which uses the distribution of c_k to estimate the distribution of c_{k+1}. We believe this transition model is a crucial step towards achieving high-performance discrete decoding. Our study demonstrates, for the first time, an improvement in closed-loop performance by using discrete state selection instead of selecting targets with a mandatory hold period.

While we decoded discrete states corresponding to distinct task actions (i.e., "move" and "stop") we note that decoding discrete states may also extend beyond task actions. It may be possible to further increase performance by allowing the discrete and continuous decoders to interact. For example, decoding discrete states may be used to modify a continuous decoder (e.g., [31]). In a simple example, one might have two continuous decoders, one used to make "ballistic" movements and the other used to make "fine" movements. The continuous decoder under "ballistic" control may be optimized for making quick movements across the workspace, while the continuous decoder under "fine" control would be optimized for making small and precise movements. A discrete decoder would determine the probabilities of being in the "ballistic" or "fine" control states, which could be used to choose (or mix) the corresponding continuous decoders. In this manner, the decoder would no longer be time-invariant, but would instead be a piecewise composition of decoders. The resulting nonlinear decoder, whose parameters change based on the decoded discrete state distribution, may be able to improve control during different task epochs. Existing offline simulations along these lines suggest that this may lead to increased neural prosthetic performance (e.g., [31]–[33]). Future work may explore the extent to which such interactions may increase closed-loop neural prosthetic performance.

We observed that neural activity was very noisy on single trials so that the distribution of neural activity during the "move" and "stop" state were highly overlapping. In light of this, there are at least two possible methods by which HMM performance might be further improved. First, in our study, we decoded the "stop" state by modeling neural activity when the monkey was holding a target. Therefore, our "stop" state corresponded to the intention to hold the cursor still over a target. However, in clinical trials with human participants, other cues, such as "squeeze" or "hand open" might be used to signal a "click" [18]. If these imagined movements have modulation that is distinct from controlling the cursor, performance may substantially improve because the neural activity would be more separable. Second, it might be possible to incorporate neural dynamical estimation to more robustly distinguish "move" and "stop" activity. A recent report demonstrates that when modeling the dynamics of the neural population activity, the neural activity during the "stop" state is closer to a fixed point than the "move" state in neural state space [11]. These dynamics accentuate the differences in neural activity between the "move" and "stop" state, effectively de-noising the observations. We predict that incorporating this technique into an HMM would improve decoding performance.

V. Conclusion

As both analog and discrete actions are a part of everyday motor control tasks, it is important that neural prostheses be capable of decoding both signals at high-levels of performance. We report that an HMM discrete decoder can significantly increase neural prosthetic performance by running in parallel to a continuous decoder. Our results suggest that existing neural prostheses using only continuous decoders can be further improved by incorporating a parallel HMM decoder. These advances are important for further increasing the utility, usability, and clinical viability of intracortical neural prostheses.

References

[1] D. M. Taylor, S. I. H. Tillery, and A. B. Schwartz, "Direct cortical control of 3D neuroprosthetic devices," *Science*, vol. 296, no. 5574, pp. 1829–32, Jun 2002.

[2] J. M. Carmena, M. A. Lebedev, R. E. Crist, J. E. O'Doherty, D. M. Santucci, D. F. Dimitrov, P. G. Patil, C. S. Henriquez, and M. A. L. Nicolelis, "Learning to control a brain-machine interface for reaching and grasping by primates," *PLoS Biology*, vol. 1, no. 2, p. E42, Nov 2003.

[3] M. Velliste, S. Perel, M. C. Spalding, A. S. Whitford, and A. B. Schwartz, "Cortical control of a prosthetic arm for self-feeding," *Nature*, vol. 453, no. 7198, pp. 1098–101, Jun 2008.

[4] S.-P. Kim, J. D. Simeral, L. R. Hochberg, J. P. Donoghue, and M. J. Black, "Neural control of computer cursor velocity by decoding motor cortical spiking activity in humans with tetraplegia," *Journal of Neural Engineering*, vol. 5, no. 4, pp. 455–76, Dec 2008.

[5] V. Gilja, P. Nuyujukian, C. A. Chestek, J. P. Cunningham, B. M. Yu, J. M. Fan, M. M. Churchland, M. T. Kaufman, J. C. Kao, S. I. Ryu, and K. V. Shenoy, "A high-performance neural prosthesis enabled by control algorithm design," *Nature Neuroscience*, vol. 15, no. 12, pp. 1752–7, Nov 2012.

[6] L. R. Hochberg, D. Bacher, B. Jarosiewicz, N. Y. Masse, J. D. Simeral, J. Vogel, S. Haddadin, J. Liu, S. S. Cash, P. van der Smagt, and J. P. Donoghue, "Reach and grasp by people with tetraplegia using a neurally controlled robotic arm," *Nature*, vol. 485, no. 7398, pp. 372–5, May 2012.

[7] J. L. Collinger, B. Wodlinger, J. E. Downey, W. Wang, E. C. Tyler-Kabara, D. J. Weber, A. J. C. McMorland, M. Velliste, M. L. Boninger, and A. B. Schwartz, "High-performance neuroprosthetic control by an individual with tetraplegia," *Lancet*, vol. 381, no. 9866, pp. 557–64, Feb 2013.

[8] B. Wodlinger, J. E. Downey, E. C. Tyler-Kabara, A. B. Schwartz, M. L. Boninger, and J. L. Collinger, "Ten-dimensional anthropomorphic arm control in a human brain-machine interface: difficulties, solutions, and limitations." *Journal of Neural Engineering*, vol. 12, no. 1, p. 016011, Dec 2014.

[9] S. Musallam, B. D. Corneil, B. Greger, H. Scherberger, and R. A. Andersen, "Cognitive control signals for neural prosthetics," *Science*, vol. 305, no. 5681, pp. 258–262, 2004.

[10] G. Santhanam, S. I. Ryu, B. M. Yu, A. Afshar, and K. V. Shenoy, "A high-performance brain-computer interface," *Nature*, vol. 442, pp. 195–198, Jul 2006.

[11] J. C. Kao, P. Nuyujukian, S. I. Ryu, M. M. Churchland, and J. P. Cunningham, "Single-trial dynamics of motor cortex and their applications to brain-machine interfaces," *Nature Communications*, vol. 6, no. May, pp. 1–12, 2015.

[12] S. I. Ryu and K. V. Shenoy, "Human cortical prostheses: lost in translation?" *Neurosurgical Focus*, vol. 27, p. E5, 2009.

[13] V. Gilja, C. A. Chestek, I. Diester, J. M. Henderson, and K. V. Shenoy, "Challenges and opportunities for next-generation intracortically based neural prostheses," *IEEE Transactions on Biomedical Engineering*, vol. 58, no. 7, pp. 1891–1899, 2011.

[14] J. C. Kao, S. D. Stavisky, D. Sussillo, P. Nuyujukian, and K. V. Shenoy, "Information systems opportunities in brain-machine interface decoders," *Proceedings of the IEEE*, vol. 102, no. 5, pp. 666–682, 2014.

[15] P. Nuyujukian, J. M. Fan, J. C. Kao, S. I. Ryu, and K. V. Shenoy, "A high-performance keyboard neural prosthesis enabled by task optimization," *IEEE Transactions on Biomedical Engineering*, vol. 62, no. 1, pp. 21–29, 2015.

[16] P. Nuyujukian, J. C. Kao, J. M. Fan, S. D. Stavisky, S. I. Ryu, and K. V. Shenoy, "Performance sustaining intracortical neural prostheses," *Journal of Neural Engineering*, vol. 11, no. 6, p. 066003, 2014.

[17] D. Sussillo, P. Nuyujukian, J. M. Fan, J. C. Kao, S. D. Stavisky, S. I. Ryu, and K. V. Shenoy, "A recurrent neural network for closed-loop intracortical brain-machine interface decoders," *Journal of Neural Engineering*, vol. 9, no. 2, p. 026027, Apr 2012.

[18] S.-P. Kim, J. D. Simeral, L. R. Hochberg, J. P. Donoghue, G. M. Friehs, and M. J. Black, "Point-and-click cursor control with an intracortical neural interface system by humans with tetraplegia," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 19, no. 2, pp. 193–203, Apr 2011.

[19] D. Bacher, B. Jarosiewicz, N. Y. Masse, S. D. Stavisky, J. D. Simeral, K. Newell, E. M. Oakley, S. S. Cash, G. Friehs, and L. R. Hochberg, "Neural point-and-click communication by a person with incomplete locked-In syndrome," *Neurorehabilitation and Neural Repair*, 2014.

[20] V. Gilja, C. Pandarinath, C. H. Blabe, P. Nuyujukian, J. D. Simeral, A. A. Sarma, B. L. Sorice, J. A. Perge, B. Jarosiewicz, L. R. Hochberg, K. V. Shenoy, and J. M. Henderson, "Clinical translation of a high-performance neural prosthesis," *Nature Medicine*, vol. 21, no. 10, pp. 1142–1145, 2015.

[21] C. M. Bishop, *Pattern recognition and machine learning*. Springer, 2006.

[22] D. Koller and N. Friedman, *Probabilistic graphical models*. Cambridge, UK: MIT Press, 2009.

[23] C. Kemere, G. Santhanam, B. M. Yu, A. Afshar, S. I. Ryu, T. H. Meng, and K. V. Shenoy, "Detecting neural-state transitions using hidden Markov models for motor cortical prostheses," *Journal of Neurophysiology*, vol. 100, pp. 2441–2452, 2008.

[24] K. V. Shenoy, D. Meeker, S. Cao, S. A. Kureshi, B. Pesaran, C. A. Buneo, A. P. Batista, P. P. Mitra, J. W. Burdick, and R. A. Andersen, "Neural prosthetic control signals from plan activity," *NeuroReport*, vol. 14, no. 4, pp. 591–6, 2003.

[25] N. Achtman, A. Afshar, G. Santhanam, B. M. Yu, S. I. Ryu, and K. V. Shenoy, "Free-paced high-performance braincomputer interfaces," *Journal of Neural Engineering*, vol. 4, no. 3, pp. 336–347, 2007.

[26] K. L. Brigmann, H. D. I. Abarbanel, and W. B. K. Jr., "Optical imaging of neuronal populations during decision-making," *Science*, vol. 307, pp. 896–901, 2005.

[27] D. Durstewitz, N. M. Vittoz, S. B. Floresco, and J. K. Seamans, "Abrupt transitions between prefrontal neural ensemble states accompany behavioral transitions during rule learning," *Neuron*, vol. 66, no. 3, pp. 438–448, 2010.

[28] V. Aggarwal, M. Mollazadeh, A. G. Davidson, M. H. Schieber, and N. V. Thakor, "State-based decoding of hand and finger kinematics using neuronal ensemble and LFP activity during dexterous reach-to-grasp movements," *Journal of Neurophysiology*, vol. 109, no. 12, pp. 3067–81, 2013.

[29] M. T. Kaufman, M. M. Churchland, S. I. Ryu, and K. V. Shenoy, "Vacillation, indecision and hesitation in moment-by-moment decoding of monkey motor cortex," *eLife*, vol. 4, pp. 1–21, 2015.

[30] E. Stark and M. Abeles, "Predicting movement from multiunit activity," *The Journal of Neuroscience*, vol. 27, no. 31, pp. 8387–94, Aug 2007.

[31] B. M. Yu, C. Kemere, G. Santhanam, A. Afshar, S. I. Ryu, T. H. Meng, M. Sahani, and K. V. Shenoy, "Mixture of trajectory models for neural decoding of goal-directed movements," *Journal of Neurophysiology*, vol. 97, no. 5, pp. 3763–80, May 2007.

[32] X. Kang, M. Schieber, and N. Thakor, "Decoding of finger, hand and arm kinematics using switching linear dynamical systems with pre-motor cortical ensembles," in *Proceedings of the 34th Annual International Conference of the IEEE EMBS*, 2012, pp. 1732–1735.

[33] W. Wu, M. J. Black, D. Mumford, Y. Gao, E. Bienenstock, and J. P. Donoghue, "A switching Kalman filter model for the motor cortical coding of hand motion," in *Proceedings of the 25th Annual International Conference of the IEEE EMBS*. Ieee, 2003, pp. 2083–2086.

[34] L. Srinivasan, U. T. Eden, S. K. Mitter, and E. N. Brown, "General-purpose filter design for neural prosthetic devices," *Journal of Neurophysiology*, vol. 98, no. 4, pp. 2456–2475, Aug 2007.

[35] M. M. Shanechi, Z. M. Williams, G. W. Wornell, R. C. Hu, M. Powers, and E. N. Brown, "A real-time brain-machine interface combining motor target and trajectory intent using an optimal feedback control design," *PLoS ONE*, vol. 8, no. 4, pp. 23–32, 2013.

[36] J. P. Cunningham, P. Nuyujukian, V. Gilja, C. A. Chestek, S. I. Ryu, and K. V. Shenoy, "A closed-loop human simulator for investigating the role of feedback control in brain-machine interfaces," *Journal of Neurophysiology*, vol. 105, pp. 1932–1949, 2011.

[37] J. M. Fan, P. Nuyujukian, J. C. Kao, C. A. Chestek, S. I. Ryu, and K. V. Shenoy, "Intention estimation in brain machine interfaces," *Journal of Neuroengineering*, vol. 11, no. 1, p. 016004, 2014.

[38] R. Davoodi and G. E. Loeb, "Real-time animation software for customized training to use motor prosthetic systems," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 20, no. 2, pp. 134–142, 2012.

[39] S. D. Stavisky, J. C. Kao, P. Nuyujukian, S. I. Ryu, and K. V. Shenoy, "A high performing brainmachine interface driven by low-frequency local field potentials alone and together with spikes," *Journal of Neural Engineering*, vol. 12, no. 3, p. 036009, 2015.

[40] P. Nuyujukian, J. M. Fan, V. Gilja, P. S. Kalanithi, C. A. Chestek, and K. V. Shenoy, "Monkey models for brain-machine interfaces: the need for maintaining diversity," in *Proceedings of the 33rd Annual Conference of the IEEE EMBS*, vol. 2011, Jan 2011, pp. 1301–5.

[41] A. J. Suminski, D. C. Tkach, A. H. Fagg, and N. G. Hatsopoulos, "Incorporating feedback from multiple sensory modalities enhances brain-machine interface control," *Journal of Neuroscience*, vol. 30, no. 50, pp. 16 777–16 787, 2010.

[42] M. T. Kaufman, M. M. Churchland, S. I. Ryu, and K. V. Shenoy, "Cortical activity in the null space: permitting preparation without movement," *Nature Neuroscience*, vol. 17, no. 3, pp. 440–8, Mar 2014.

[43] M. M. Churchland, J. P. Cunningham, M. T. Kaufman, J. D. Foster, P. Nuyujukian, S. I. Ryu, and K. V. Shenoy, "Neural population dynamics during reaching," *Nature*, vol. 487, no. 7405, pp. 51–6, Jul 2012.
[44] J. C. Kao, P. Nuyujukian, S. D. Stavisky, S. I. Ryu, S. Ganguli, and K. V. Shenoy, "Investigating the role of firing-rate normalization and dimensionality reduction in brain-machine interface robustness," in *Proceedings of the 35th Annual Conference of the IEEE EMBS*, vol. 2010, 2013, pp. 3–7.
[45] S. Koyama, S. M. Chase, A. S. Whitford, M. Velliste, A. B. Schwartz, and R. E. Kass, "Comparison of brain-computer interface decoding algorithms in open-loop and closed-loop control," *Journal of Computational Neuroscience*, vol. 29, no. 1-2, pp. 73–87, Aug 2010.
[46] S. M. Chase, A. B. Schwartz, and R. E. Kass, "Bias, optimal linear estimation, and the differences between open-loop simulation and closed-loop performance of spiking-based brain-computer interface algorithms," *Neural Networks*, vol. 22, no. 9, pp. 1203–1213, 2009.
[47] B. M. Yu, J. P. Cunningham, G. Santhanam, S. I. Ryu, and K. V. Shenoy, "Gaussian-process factor analysis for low-dimensional single-trial analysis of neural population activity," *Journal of Neurophysiology*, vol. 102, pp. 612–635, 2009.
[48] J. P. Cunningham and B. M. Yu, "Dimensionality reduction for large-scale neural recordings," *Nature Neuroscience*, vol. 17, no. 11, pp. 1500–1509, Aug 2014.

VI. Acknowledgments

We thank Mackenzie Risch, John Aguayo, Clare Sherman and Erica Morgan for surgical assistance and expert veterinary care; Sandy Eisensee, Evelyn Castenada, and Beverly Davis for administrative support; Boris Oskotsky for information technology support.

VII. Author contributions

JCK and PN were responsible for designing and conducting experiments, algorithm development and data analysis. JCK was responsible for manuscript writeup. PN assisted in manuscript review. SIR was responsible for surgical implantation and assisted in manuscript review. KVS was involved in all aspects of experimentation, data review, and manuscript writeup.

Jonathan C. Kao (S13) received the B.S. and M.S. degree in electrical engineering from Stanford University in 2010. He is currently pursuing his Ph.D. degree in electrical engineering at Stanford University. His research interests include algorithms for neural prosthetic control, neural dynamical systems modeling, and the development of clinically viable neural prostheses.

Paul Nuyujukian (S05-M13) received the B.S. degree in cybernetics from the University of California, Los Angeles, in 2006. He received an M.S. and Ph.D. degrees in bioengineering and M.D degree from Stanford University in 2011, 2012, and 2014, respectively. He is currently a postdoctoral scholar in the department of Neurosurgery at Stanford University. His research interests include the development and clinical translation of neural prostheses.

Stephen I. Ryu received the B.S. and M.S. degree in electrical engineering from Stanford University, Stanford, CA, in 1994 and 1995, respectively, and the M.D. degree from the University of California at San Diego, La Jolla, in 1999. He was a postdoctoral fellow at Stanford in Neurobiology and Electrical Engineering from 2002-2006. He completed neurosurgical residency and fellowship training at Stanford University in 2006. He was on faculty as an Assistant Professor of Neurosurgery at Stanford University until 2009. Since 2009, he has been a Consulting Professor of Electrical Engineering at Stanford University. He now practices at the Palo Alto Medical Foundation in Palo Alto, CA. His research interests include brainmachine interfaces, neural prosthetics, minimally invasive neurosurgery, and stereotactic radiosurgery.

Krishna V. Shenoy (S87–M01–SM06) received the B.S. degree in electrical engineering from U.C. Irvine in 1990, and the M.S. and Ph.D. degrees in electrical engineering from MIT, Cambridge, in 1992 and 1995, respectively. He was a Neurobiology Postdoctoral Fellow at Caltech from 1995 to 2001 and then joined Stanford University where he is currently a Professor in the Departments of Electrical Engineering, Bioengineering, and Neurobiology, and in the Bio-X and Neurosciences Programs. He is also with the Stanford Neurosciences Institute and is a Howard Hughes Medical Institute Investigator. His research interests include computational motor neurophysiology and neural prosthetic system design. He is the director of the Neural Prosthetic Systems Laboratory and co-director of the Neural Prosthetics Translational Laboratory at Stanford University. Dr. Shenoy was a recipient of the 1996 Hertz Foundation Doctoral Thesis Prize, a Burroughs Wellcome Fund Career Award in the Biomedical Sciences, an Alfred P. Sloan Research Fellowship, a McKnight Endowment Fund in Neuroscience Technological Innovations in Neurosciences Award, a 2009 National Institutes of Health Director's Pioneer Award, the 2010 Stanford University Postdoctoral Mentoring Award, and the 2013 Distinguished Alumnus Award from the Henry Samueli School of Engineering at U.C. Irvine.