A concurrent brain-machine interface for enhanced motor function

Maryam M. Shanechi[1, 2, 3], Rollin C. Hu[2, 3], Marissa Powers[2], Gregory W. Wornell[1], Emery N. Brown[3, 4], & Ziv M. Williams[2, 3]

[1] Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, USA
[2] Department of Neurosurgery, Massachusetts General Hospital, Boston, MA, USA
[3] Harvard Medical School, Boston, MA, USA
[4] Department of Brain and Cognitive Sciences, Massachusetts Institute of Technology, Cambridge, MA, USA

Brain-machine interfaces (BMIs) largely focus on restoring original motor function. However, a more compelling aim of such research is the development of BMIs that can surpass original motor function by also considering the higher-level goal of the task. Since typical tasks consist of sequential movements, a BMI that can concurrently decode the complete sequence before execution may also be able to reformulate and perform such motor plans more effectively. Using population-wide modeling, we discover two functionally distinct neural subpopulations in the primate premotor cortex that allow two planned targets of a sequential movement to be concurrently held in working memory without degradation. Such surprising stability occurs because one subpopulation always encodes only currently held target information and the other always encodes only newly added target information, irrespective of target locations. Based on these findings, we develop a BMI that can simultaneously decode the full motor sequence in advance and then accurately execute it as desired.

An important motivation to date for the design of brain-machine interfaces (BMIs) has been their potential ability to restore lost motor function in individuals with neurological injury or disease (e.g., due to motor paralysis or stroke). In such cases, the envisioned role of the BMI is to decode intended motor function from neural activity in the relevant areas of the brain, and use this information to control an affected limb, prosthetic, or other device.

The design of such BMIs has received considerable attention in recent years[1-16]. Work to date has largely focused on achieving relatively simple motor goals, such as that associated with the task of moving a cursor on a display to an individual target location. However, in many natural tasks—such as playing a succession of notes on a piano—the goal is more complex, and the motor plan for achieving it can be viewed as a sequence of such simple plan elements to be executed in order.

Our focus is on the design of BMIs that can achieve the goal of these sequential motor plans. For such a BMI, one obvious objective would be to decode and execute the intended plan elements one by one. A BMI designed to meet this objective has a basic *sequential* BMI architecture. Because this objective corresponds to a low-level interpretation of the overall motor goal of the individual, such an architecture is inherently limited. While such a BMI has the potential to restore or match original motor functionality, a more compelling objective is the design of a BMI that may be able to surpass such natural motor functionality by considering the motor goal at a higher-level.

As a basic illustration of the opportunity, consider the task of picking up a key and unlocking a door. While a BMI that reproduces such natural movement would certainly be useful, an even more valuable BMI would be one that simultaneously decodes the full sequence of elements of the motor plan, then analyzes the sequence prior to action to determine ways to perform the task more effectively. For example, the BMI might determine a way to accomplish the task more quickly, or more efficiently (to within any physical constraints that might exist). Alternatively, based on additional sensor inputs, the BMI might determine that the planned sequence of movements would result in an accident, such as the person's hand knocking over an object while reaching for the door, and thus will modify the execution of the task to avoid such an accident.

The realization of BMIs that can enhance natural motor function in this way will obviously require significant technological innovations. But as a key initial step, it requires shifting to a *concurrent* BMI architecture in which the elements of a planned motor task are decoded in parallel (i.e., at once), in contrast to the serial process of a sequential BMI. Hence, the feasibility of such BMIs hinges on the degree to which the elements of a motor plan sequence can, in fact, be decoded concurrently. This is the starting point for our research.

Prior work has demonstrated that individual neurons in the premotor cortex of primates display selective responses to planned single-targeted movements before their initiation, and that such responses often remain sustained during working memory until movement execution[17-23]. Such responses have been successfully exploited in the design of BMIs for single-target tasks[15, 16]. In

comparison, the neural encoding of tasks requiring a sequence of planned targeted movements to be formulated and performed is less well understood, and the design of real-time BMIs that can concurrently decode and then execute such sequential motor plans remains unexplored. Prior work has shown that an individual neuron can display a response that is selective to one or more elements of a sequential motor plan[24-37] (see also Discussion). However, little is known regarding how information about multiple elements of a sequential motor plan (e.g., the planned targets of a sequential movement) is simultaneously distributed across the whole premotor population, and whether these plan elements can be accurately decoded from the neural population in a concurrent manner. More importantly, it is necessary to determine whether and how adding information about the elements of the motor plan, in sequence, to working memory affects the integrity of information about the plan elements that are already held. Finally, it is necessary to assess robustness—whether a BMI limited to recording from only a small number of neurons over time is able to achieve sufficient and consistent decoding accuracy.

Here, we find that sequential motor plans can be decoded simultaneously, accurately, robustly, and in advance of movement from the neural activity in the premotor cortex of monkeys. Additionally our study reveals a surprisingly structured encoding mechanism that is used by the premotor populations for these sequential plans and that, in turn, enables their accurate and concurrent decoding. Based on these findings, we develop and implement a real-time BMI that can concurrently decode a dual sequence of motor targets and then execute them as desired.

RESULTS

In the study, rhesus monkeys were trained to perform a task in which two targets were presented, in sequence, on a computer display. Each of the targets could take on one of four possible spatial locations (“up”, “down”, “left”, or “right”). Repeated locations were precluded, so there were a total of 12 possible combinations (sequences) of two consecutive distinct target locations. After a blank screen variable delay, a “go” cue appeared directing the monkeys to sequentially move a cursor from the center of the screen to each of the two remembered targets, in order (Dual-target task; **Fig. 1a, b**). We define the working memory period as the 500 ms blank-screen interval following presentation of the second target and before the earliest possible “go” cue.

Multiple-unit responses were recorded from the premotor cortex as the primates performed this task. We recorded 281 well-isolated neurons from the supplementary motor area (SMA) and dorsal premotor cortex (PMd) over 11 sessions, for an average of 26 ± 6 cells (mean ± s.d.) per recording session. Inhomogeneous Poisson models were fitted to each neuron's spiking activity using an expectation-maximization algorithm[38] (see Methods and Supplementary Methods). Using these models, we employed a maximum-likelihood decoder to quantify the probabilities that groups of neurons could correctly identify the first and second targets on a trial-by-trial basis during the working memory period (leave-one-out cross-validation; see Methods). We used decoding accuracy as our measure of the amount of information encoded by a population of neurons about each target. Specifically, for an individual (first or second) target, we measured the percentage of trials in which the maximum-likelihood decoder correctly predicted the respective target from that population's activity. Likewise, we measured the amount of information encoded about the full sequence as the percentage of trials in which both targets were correctly decoded.

Accurate and concurrent encoding of the motor sequence by the premotor population

We find that neural population activity within the premotor cortex accurately encoded the location of both targets during the working memory period. During this period, the population correctly encoded the first and second targets on up to 85% and 82% of the trials in a session, respectively. When considering all 12 possible target combinations, the population encoded both targets correctly on up to 72% of the trials in a session (**Fig. 2a; Supplementary Fig. 7**). Across all tested sessions, the population correctly encoded the first and second targets on average on 76 ± 11% and 56 ± 17% of trials, respectively, both of which were significantly above chance (one-sided *Z*-test, $P < 10^{-15}$; **Supplementary Fig. 2**). Also, the population encoded both targets correctly on average on 45 ± 12% of the trials across all sessions, which was also far higher than chance at 1/12≈8% (one-sided *Z*-test, $P < 10^{-15}$; see Supplementary Results for relevant controls). These results were consistent across the two monkeys, i.e., the first target, second target, and sequence decoding accuracies were all significantly above chance in both monkeys ($P < 10^{-15}$; **Supplementary Fig. 8**; see Supplementary Results).

Robustness of the encoding

Only a small number of cells were needed to decode the target sequence with high accuracy. When performing the decoding analysis over all trials, which employed all 12 possible target combinations, only 29% of the population (7.5 cells) was needed on average to achieve higher than 90% of the population sequence accuracy (**Fig. 2b**; see Supplementary Methods). When performing the decoding analysis over subsets of all trials that employed only 4 or 8 target combinations, population sequence accuracies in the best session were as high as 93% and 80%, respectively. In these cases, decoding from only 2 and 4 cells, respectively, was sufficient to achieve higher than 90% of these sequence accuracies.

Real-time concurrent BMI for sequential movement decoding and execution

Motivated by the observation that both targets can be concurrently and accurately decoded from the responses of relatively few neurons in the premotor cortex, we developed a real-time BMI capable of predicting both targets simultaneously prior to monkey's motor response and then executing the targeted movements. In the associated experiments, we recorded a mean of 20 ± 2 cells per session from the premotor cortex of the same monkeys. Here, Poisson models were first fitted to the neural population activity prior to the "go" cue (see Methods and **Supplementary Fig. 4**) as the primates rehearsed a subset of target combinations that included either 4 or 8 possible sequences over an average of 26 ± 2 training trials per sequence (**Fig. 1b**). We chose to use either four or eight sequences in the BMI experiments to obtain sufficient training and real-time trials per session (see Supplementary Methods).

Using the Poisson models, sequence decoding accuracies for the set of 4 and 8 sequences in these training sessions (found using leave-one-out cross-validation) were 79 ± 2% and 80 ± 3% (mean ± s.e.m.; one-sided *Z*-test, $P < 10^{-15}$), respectively. Following training, the primates performed the same task as before, but with the cursor now being sequentially positioned by the BMI on the targets decoded from the recorded neuronal activity during the single preceding working memory period (**Fig. 1c**; see Methods). Here, BMI-generated cursor movements were set to occur immediately following the presentation of the "go" cue and the added delays were selected to match the reaction times that the monkeys normally experienced when moving the

cursor themselves following the "go" cue (obviously, cursor movements could be generated without the added delays if desired).

Both monkeys performed a total of 459 trials on the four-sequence set, and one monkey performed 110 trials on the eight-sequence set using the real-time BMI. Sequence accuracies for the set of four and eight sequences were 72 ± 2% and 71 ± 4%, respectively, both of which were significantly above chance (mean ± s.e.m., one-sided *Z*-test, $P < 10^{-15}$). Both training and real-time BMI accuracies were similar and significantly above chance across the two monkeys (one-sided *Z*-test, $P < 10^{-15}$; see Supplementary Results). Sequence accuracies using the BMI were also close to the cross-validated sequence accuracies during the training sessions when taking into account the primates' natural error rates during the standard task (**Fig. 3**). In fact the 95% confidence bounds for the two accuracies were overlapping (72 ± 4% vs. 73 ± 3% and 71 ± 8% vs. 66 ± 6%, for the sets of four and eight sequences, respectively; see Supplementary Methods). These results established that the two plan elements, i.e., the two intended sequential targets of movement, could be simultaneously predicted in advance of movement and then executed by a real-time BMI with high accuracy.

Potential for enhancing sequential motor functions

Concurrent decoding of the full sequential plan prior to movement enables the BMI to analyze it before its execution and, therefore, potentially find ways to execute the task more effectively. As a simple but illustrative example of such a prospective capability, in the context of in the context of our experiments, we find that sequence decoding accuracy for the set of four, eight, and twelve sequences reached 90% of the maximum asymptotic accuracy possible, on average, after 488 ms, 561 ms, and 641 ms from the *initial* presentation of the second target, respectively. Even when excluding any time delays included in the task, these decoding times were shorter compared to the mean times it took for the monkeys to react to the two "go" cues and reach the two targets, which was 791 ms (one-sided *t*-test, $P < 0.01$; this is even though the monkeys were already aware of the target locations by the time the "go" cue appeared and all delays including the time required to move between the targets were excluded; **Supplementary Fig. 9**). Indeed, in our experiments, time delays in the BMI-generated movements were introduced only to convey to the monkeys the impression that they were performing the standard task as before. Since

information about the full sequence is decoded in advance, it would therefore be possible to then accurately execute both movements in any desired fashion – even effectively instantaneously if needed (see also Supplementary Results for speed-accuracy tradeoff). While this is one example of the potential for enhancing sequential motor functions, in general, having information about all the plan elements in advance allows the BMI to analyze the higher-level goal of the task and reformulate the plan to perform it more effectively (see Discussion).

Population encoding structure reveals a partitioning mechanism

Observing that both target locations could be accurately and concurrently predicted from the neural population responses, we further examined the spatial and temporal structure of their encoding. In particular, we investigated how neurons within the premotor cortex were able to add new information about the second target to working memory without compromising the integrity of information about the first target that was already being held.

We find, surprisingly, that most cells encoded significant information about only the first (currently held) or the second (newly added) target during the working memory period, and that this partitioning remained stable throughout recordings per day across all tested target locations. Of the 281 neurons recorded in all sessions, 46% had a target accuracy significantly higher than chance for at least one of the two targets during the working memory period (one-sided *Z*-test, $P < 0.01$; see Supplementary Methods). Of these, 68% encoded significant information about only the first currently held target, and 23% encoded significant information about only the second added target (one-sided *Z*-test; $P < 0.01$, **Fig. 4, Supplementary Figs. 10, 11**). The percentage of cells that encoded significant information about both targets was only 9% (one-sided *Z*-test; $P < 0.01$; note that a Bonferroni correction for multiple comparisons was done for *all* comparisons; **Supplementary Fig. 12**) and, even among these, most had target accuracies much closer to one of the two subpopulations of cells that significantly encoded only one target (**Fig. 5** and **Supplementary Figs. 3** and **6**). These results revealed a highly significant divergence in the amount of information encoded by the two subpopulations of neurons about the two targets (random permutation test, $P < 10^{-15}$; see Supplementary Methods and **Supplementary Fig. 6**). Moreover, each of the two subpopulations was only predictive of whether the respective first or

second upcoming movement would be performed correctly or incorrectly by the primates following the "go" cue (see Supplementary Results).

Effect of adding information to working memory on the integrity of existing information

In order to further examine how adding a new target to working memory affected the integrity of the currently held target, we disambiguated the process of holding information in working memory from that of adding information to it. The results were obtained from sessions in which the monkeys performed the standard dual-target trials (as before), but also performed single-target trials in randomly interleaved fashion (see Methods). Unlike dual-target trials, on single-target trials only the first target was presented and the second target presentation period was replaced with a blank screen period of the same duration. The task timing was otherwise unchanged compared to the dual-target task.

We find that adding information about the second target location to working memory did not incur loss of information about the first target location. Of the cells that encoded significant information about the first target during working memory in single-target trials (one-sided *Z*-test, $P < 0.01$), most (74%) provided the same level of accuracy in decoding the first target during working memory in dual-target trials, despite the addition of a second target (χ^2 test, $P > 0.05$). For the population, there was also no significant difference in the first target accuracy during the working memory period when comparing dual-target and single-target trials across sessions (Wilcoxon's signed-rank test, $P = 0.69$; **Fig. 6**). These results demonstrate that the subpopulation encoding the first target and their responses remained largely unchanged when the second target was added to working memory and, therefore, the addition of information about the second target did not comprise the integrity of information already held about the first target.

Stability of the neural encoding structure

While implicit in the preceding results, it should be emphasized that as the pair of presented target locations varied over the hundreds of trials typical of a given day's session, most neurons remained dedicated to encoding only the first (currently held) or only the second (added) target. For the two subpopulations of cells that encoded significant information about the respective first and second targets alone, most (89%) provided substantially the same level of accuracy in

decoding their respective targets in the first and second halves of the recording session (χ^2 test, $P > 0.05$; see Supplementary Methods). Also, the sequence decoding accuracy (across all 12 sequences) of the entire population did not change over time between the first and second halves of the sessions (Wilcoxon's signed-rank test, $P = 0.37$). Inherently, the neural decoding in our BMI exploited this stability of the two constituent subpopulations to achieve sustainable performance.

DISCUSSION

The purpose of the present study was to examine how multiple planned targets of sequential movement are concurrently encoded by premotor neurons during working memory, and to determine whether the activity recorded simultaneously from multiple single-neurons can be used to concurrently and accurately decode the complete motor plan sequence in advance of movement and in real-time. Three methodological approaches were used to investigate these questions. First, we simultaneously recorded the activity of multiple cells across the whole premotor population, rather than one at a time over separate days. Second, we used an interleaved dual-target/single-target task in order to dissociate the dynamic process of maintaining target-related information in working memory from that of adding new information to it. Finally, we employed a population-wide maximum-likelihood decoding approach that allowed us to (i) define an accuracy measure for the amount of information that is concurrently encoded by neuronal activity and, (ii) examine the spatiotemporal distribution of information across the whole population.

Neural partitioning mechanism

Our results reveal a novel functional structure within the premotor cortex that enabled accurate and concurrent decoding of two planned motor targets across multiple spatial locations. We specifically find that the premotor population was partitioned into two largely disjoint subpopulations of cells, one dedicated to always encoding only the first (currently held) target and one to always encoding only the second (added) target. Moreover, while the two target locations changed from trial to trial, the two encoding subpopulations did not. Interestingly, the subpopulation dedicated to encoding the first target and their responses remained largely unchanged when the second target was added to working memory, so that the process of adding

information did not compromise the integrity of existing information. Also remarkably, only a small number of neurons were sufficient to accurately predict the location of both targets, making the decoding of such information highly robust.

Prior studies have shown that individual neurons within areas such as the parietal, premotor and prefrontal cortex can display selective responses to one or more elements of a sequential motor plan[24-37, 39, 40]. Some neurons (often a small fraction) display increased activity for a specific combination of movements (for example a push followed by a pull of a manipulandum) during a preceding delay, suggesting that they encode information about more than one motor goal at a time[25-28, 34, 40]. Other cells have also been found that display selective responses during movement itself with increased activity prior to performing a particular movement (e.g. a push) only when it follows another specific movement (e.g., a pull) in sequence[26, 28, 40], or prior to a movement only if it has a particular order in the sequence[28, 40]. What has remained unclear, however, is how information about individual elements of such sequential plans is *simultaneously* distributed across the whole population, and whether the process of adding *new information* about an element to working memory compromises the integrity of information already held.

It is important to note that certain behavioral contexts may require more than two targets to be held in working memory prior to executing their corresponding movements. In such situations, it is possible that the premotor cortex employs a different encoding strategy, for example, when adding a third target to two concurrently held targets. An interesting finding in the present study is that the added target was encoded by a smaller number of cells than the number encoding the first concurrently held target. It is, therefore, interesting to speculate whether the observation that a progressively smaller number of neurons were recruited for the later target provides a limitation on the number of items of information that may be held during working memory. Indeed, behavioral studies in both human and non-human primates have demonstrated that most subjects can only accurately hold in working memory, on average, three to four items at any given time (ranging between 1.5 to 5)[41]. Future experiments that employ a larger number of targets may provide insight into this question and, in particular, why only a small number of items can be held in working memory in comparison to other forms of memory storage[42].

While the main focus in this study was to examine how information about existing and added targets is encoded by premotor populations during the working memory period, it is important to emphasize that these findings do not necessarily reflect how information is encoded at other times. We particularly find that, consistent with prior studies[26, 28, 40], neurons often dramatically altered the degree to which they encoded information about the two targets across different time-points during the task. Some cells, for example, encoded no information about the second target during the working memory period, but then encoded significant information about the second target during the second movement itself (**Supplementary Fig. 3a**). Therefore, observing that a cell predominantly encodes information about the first target during the working memory period, for example, does not necessarily indicate that it does not encode the second target at any other time during the task. Such shifts in activity may reflect the dynamic role premotor neurons play in processing, maintaining, and then executing motor plans in combination with other motor cortical areas.

Another question that arises from the study is how information encoded by premotor neurons is 'read-out' by the brain in order to perform sequential behavior, or in other words, how this information is related to the later execution of the task. Here, we found that the subpopulation of cells that predominantly encoded information about the first target was only predictive of whether the primates would perform the first upcoming movement correctly or incorrectly. This was similarly true for the other subpopulation that predominantly encoded the second target, together suggesting that this 'partitioning strategy' was ultimately used to direct upcoming sequential behavior. In terms of the small number of cells that encoded information about both targets, it is interesting to speculate whether they may provide an important 'bridge' between distinct motor plan elements or a higher conceptual representation of specific motor combinations not provided by the other subpopulations of neurons.

Potential applications of BMIs with enhanced motor capabilities

We exploited the concurrent decoding and the neural partitioning mechanism observed in these experiments to develop a new BMI that can *concurrently* decode the intended targets of a sequential motor plan in advance of movement in real-time. Prior work[16] has shown that spatial

information about single-targeted movements can be accurately decoded before movement initiation resulting in high-performance single-target BMIs. However, in many tasks, the motor plan consists of a sequence of targeted movements to be executed in order. The present BMI thus takes advantage of the *temporal* encoding of sequential motor behavior and decodes this behavior based on how the brain naturally encodes the concurrently held information about sequential motor plans. This concurrent decoding of a sequence of planned targeted movements is in contrast to the serial application of a BMI that decodes and executes one targeted movement at a time, and in turn, introduces the possibility of a BMI that can both accurately mimic sequential motor function as well as potentially enhance it.

Specifically, because the full motor plan can be decoded at once and in advance of movement, the higher-level goal of the task can, in principle, be analyzed, and the motor plan reformulated accordingly. For instance, this could allow the task to be performed more quickly than possible by natural movement, more flexibly, or more efficiently than originally conceived. As an illustrative example, in the context of our experiments, we demonstrated that the BMI can decode and execute the two planned movements over a time-frame shorter than the time it took the primates to naturally react and execute the movements.

Other potential applications for such a concurrent BMI include altering the order in which the elements of the motor sequence are executed depending on rapid or unpredictable changes in the environment, or correcting the originally planned sequence based on the performance metrics of the task. Examples include changing the planned order of a sequence of movements prior to execution to avoid unanticipated obstacles, or proactively changing a planned sequence of letters or words intended for typing based on known spelling or grammatical rules prior to their usage. Given the present results, it is conceivable that a variety of augmented motor functions could be achieved in the future by such BMIs exploiting concurrent decoding and the neural partitioning revealed in this study. As such, there may be broad implications for use in a variety of clinical and non-clinical applications.

Note: Supplementary information is available on the Nature Neuroscience website.

ACKNOWLEDGEMENTS R.C.U is funded by the NREF, E.N.B is funded by NIH DP1 OD003646, Z.M.W is funded by NIH 5R01-HD059852, PECASE and the Whitehall Foundation.

AUTHOR CONTRIBUTIONS M.M.S. developed the real-time decoder, conceived and performed the computational analysis, assisted with animal recordings, and wrote the manuscript. R.C.H. and M.P. assisted with animal training and recordings. G.W.W. and E.N.B. were involved in the computational methodological development and edited the manuscript. Z.M.W. conceived and designed the study, performed the animal training and recordings and edited the manuscript.

COMPETING FINANCIAL INTERESTS The authors declare no competing financial interests.

Figure Legends

Figure 1 Task design and experimental setup. (**a**) Schematic illustration of a standard dual-target task over a single trial. Task periods and their timings are displayed over a single trial from left to right. The right end of the panel in which the second movement is made is truncated to conserve space. (**b**) Experimental setup for the standard training sessions. (**c**) Experimental setup for the BMI sessions.

Figure 2 Population decoding accuracy for a selected session. (**a**) Population decoding accuracy over time for the first target (red curve), second target (green curve), and the full sequence (blue curve). Each point on the curves indicates the decoding accuracy for the population over the *preceding* 500 ms window. Time at zero is aligned to the start of first target presentation. The red and green vertical bars indicate the times during which the first and second targets were presented, respectively. The first and second dashed black lines indicate the mean times at which the first and second "go" cues were given, respectively. The arrow indicates the time point of decoding for the preceding working memory period (i.e., 0–500 ms from the end of the second target presentation). The dotted lines indicate the 99% chance upper confidence bounds for the first target, second target, and sequence (out of 12 possibilities), with the same respective color scheme used above (see also Supplementary Methods). (**b**) Number of cells sufficient to reach decoding accuracy asymptote during the working memory period for the same session. The first target (red curve), second target (green curve), and sequence (blue curve) accuracies are displayed as a function of the cumulative number of cells, in descending order of single-cell

sequence accuracy. The number of cells needed to reach over 90% of the population accuracy is indicated by the vertical dashed line.

Figure 3 Performances on BMI trials. The gray bars indicate the monkeys' average behavioral accuracy, maximum-likelihood cross-validation accuracy on the training data, and real-time BMI accuracy, with their corresponding s.e.m.. The black bars indicate chance level accuracies. Performances using four sequences are displayed on the *left*, and using eight sequences are displayed on the *right*.

Figure 4 Single-neuron encoding. Examples of spiking activity for a single-neuron selective for the first target during the working memory period are shown under three sample sequences **(**see also **Supplementary Figs. 10–12)**. Here the second target is the same but the first target is different. Each *top* panel corresponds to a different sequence of movements with each row illustrating the spiking activity during a single trial and the black dots indicating the spike times. Each *bottom* panel indicates the corresponding mean firing rate estimates using the expectation-maximization procedure (see Methods). The arrow indicates the working memory period. The vertical bars/lines and their timings follow the same convention as **Fig. 2**.

Figure 5 Distribution of first and second target information across the population. Scatter plot of the first and second target accuracies for the 129 cells that encoded significant information about at least one target during the working memory period (across 12 sequences). Statistical significance of the target accuracies was tested here at the 0.01 level (see Supplementary Methods and also **Supplementary Fig. 6**). Red points indicate cells that encoded significant information about only the first target, green points indicate those that encoded significant information about only the second target, and black points indicate those that encoded significant information about both targets. The *inset* indicates the proportion of cells that encoded significant information about only the first, only the second or both targets during the working memory period with the same coloring scheme, above.

Figure 6 The effect of adding information to working memory. In an interleaved session, population decoding accuracy for the first target on dual-target trials is shown in red whereas

population decoding accuracy for the first target on single-target trials is shown in magenta. Each point on the curves indicates the decoding accuracy over the *preceding* 500 ms window. Dotted lines indicate the 95% confidence bounds for accuracy of each curve (rather than chance level). The vertical red bar indicates the time during which the first target was presented. The vertical green bar indicates the time during which the second target was or was not shown depending on the trial type. The arrow indicates the time point corresponding to the decoding accuracy of the preceding working memory period (same convention as in **Fig. 2**).

METHODS

Behavioral task. Two adult rhesus monkeys (*Macaca Mulatta*) were trained to perform a sequential delayed motor task in which they were first sequentially presented with two distinct target locations on a screen and then had to move a cursor to each in order by using a joystick (Dual-target task; **Fig. 1a**). After initial presentation of a blank screen, two targets were sequentially presented each of which could take on one of four possible spatial locations: "up" (U), "down" (D), "right" (R) or "left" (L). To ensure that the two target locations were distinct, the motor sequence was chosen at random from a total of 12 possible sequences, i.e., all possible combinations of the two target locations excluding the ones with repeated locations. Targets were displayed for 500 ms each and were interleaved by a 300 ms interval during which a blank screen was shown. Following the end of second target presentation, there was an additional blank screen variable delay of 500–800 ms following which the first "go" cue signal appeared. After this, the monkeys were required to move a cursor from the center of the screen to the first remembered target. After reaching the target, they were required to return the joystick to the center and then wait for a second "go" cue to appear after an additional 500 ms delay interval. Once the second "go" cue appeared, they were allowed to move the cursor from the center of the screen to the second remembered target. The monkeys received a juice reward if they correctly moved to the two instructed targets.

Dual-target vs. single-target task. To examine the effect of adding information about a new target to working memory, it was necessary to disambiguate the process of holding information in working memory from that of adding information to it. To do this, primates performed randomly interleaved dual-target and single-target trials in a subset of sessions. On dual-target

trials, described above, the primates were sequentially presented with two targets and then a blank screen delay. The time delay from the end of the first target presentation to the first "go" cue was therefore 1300–1600 ms. On single-target trials, in comparison, the primates were presented by only the first target, and had to keep this single target in working memory for the same total 1300–1600 ms time duration as in dual-target trials. However, here, they were not presented by a second target and were only shown a blank screen until the "go" cue.

Neurophysiologic recordings. All procedures were performed under IACUC-approved guidelines. Prior to recordings, multiple silicone multi-electrode arrays (NeuroNexus Technologies Inc., MI) were surgically implanted in each monkey. A craniotomy was placed over the premotor cortex under stereotactic guidance (David Kopf Instruments, CA). The multi-electrode arrays were separately implanted into the dorsal premotor (PMd) and the supplementary motor (SMA) areas (**Supplementary Fig. 1**). The electrode lead of each array was secured to the skull and attached to female connectors with the aid of titanium miniscrews and dental acrylic. Confirmation of electrode positions was done in both monkeys by direct visual inspection of the sulci and gyral pattern through the craniotomy. Additional post-mortem confirmation of electrode positions was made in one monkey (the second monkey is still performing experiments). Recordings began two weeks following surgical recovery. A Plexon multichannel acquisition processor was used to amplify and band-pass filter the neuronal signals (150 Hz – 8 kHz; 1 pole low-cut and 3 pole high-cut with 1000x gain; Plexon Inc., TX). Shielded cabling carried the signals from the electrode array to a set of six 16-channel amplifiers. Signals were then digitized at 40 kHz and processed to extract action potentials in real time by the Plexon workstation. Classification of the action potential waveforms was performed using template matching and principle component analysis based on waveform parameters. Only single, well-isolated units with identifiable waveform shapes and adequate refractory periods (less than 1% of spikes within a 1 millisecond interval) were used for the online experiments and offline analysis. No multiunit activity was used.

Model construction. For the analysis of standard recording sessions, we model the activity of each neuron under any given sequence as an inhomogeneous Poisson process whose likelihood function is given by[43, 44]

$$p(N^{c}_{1:K} \mid S_i) = \prod_{k=1}^{K} (\lambda_c(k \mid S_i)\Delta)^{N^{c}_{k}} \exp(-\lambda_c(k \mid S_i)\Delta) \qquad i = 1:12\,, \qquad (1)$$

where Δ is the time increment taken to be small enough to contains at most one spike, N^{c}_{k} is the binary spike event of the c'th neuron in the time interval $[(k-1)\Delta, k\Delta]$, $\lambda_c(k \mid S_i)$ is its instantaneous firing rate in that interval, S_i is the i'th sequence, and K is the total number of bins in a duration $K\Delta$. We take $\Delta = 5$ ms as the bin width of the spikes. By building the neuronal models under each sequence separately in the dual-target task we avoid making any a priori assumptions about whether the cells encode individual targets or combined sequences. For each sequence and neuron, we need to estimate the firing rate $\lambda_c(k \mid S_i)$ using the neuronal data observed. To do so, we use a state-space approach using the expectation-maximization (EM) algorithm[38, 45, 46] (see Supplementary Methods). After fitting the models, we validated them using the χ^2 goodness-of-fit test on the data[38] and confirmed that they fitted the data well ($P > 0.7$ for all cells in all sessions).

Maximum-likelihood decoder. Once models are fitted, a maximum-likelihood decoder is used to decode the intended sequence based on the neuronal activity in any period of interest. A maximum-likelihood decoder is the optimal decoder in the sense of maximizing accuracy, i.e., the percentage of trials in which the combined sequence is decoded correctly, when the sequences are equally likely to be presented as is the case in our experiments. The decoder finds the likelihood of observing the population neuronal data under each sequence and selects the sequence with the highest likelihood as its prediction. Using the likelihood model in (1) and assuming that neurons are conditionally independent given the sequence, the population likelihood under any sequence is given by

$$p(N^{1:C}_{1:K} \mid S_i) = \prod_{c=1}^{C} \prod_{k=1}^{K} (\lambda_c(k \mid S_i)\Delta)^{N^{c}_{k}} \exp(-\lambda_c(k \mid S_i)\Delta) \qquad i = 1:12$$

where K is the total number of bins in any period of interest during the trial, C is the total number of neurons, and $\lambda_c(k \mid S_i)$ for $k = 1, \cdots, K$ and $c = 1, \cdots, C$ is the estimate of the firing rate. The predicted sequence, $\hat{S}$, is thus given by

$$\hat{S} = \arg\max_{S_i} p(N^{1:C}_{1:K} \mid S_i)\,.$$

To find the sequence decoding accuracy of a single cell, the maximum-likelihood decoder uses only that cell's spiking activity to decode the sequence (**Fig. 4 and Supplementary Fig. 3**). The decoder also outputs the posterior probability of each sequence, which is the probability that it is the correct one after the neuronal observations, i.e.,

$$p(S_i \mid N_{1:K}^{1:C}) = \frac{p(N_{1:K}^{1:C} \mid S_i)}{\sum_i p(N_{1:K}^{1:C} \mid S_i)} \qquad i = 1, \cdots, 12 .$$

To dissociate the decoding accuracy of the first and second targets, denoted by T_1 and T_2, the decoder also outputs their predictions based on the neuronal activity. To do so, the decoder finds their posterior probabilities, i.e., $p(T_1 = l_1 \mid N_{1:K}^{1:C})$ and $p(T_2 = l_2 \mid N_{1:K}^{1:C})$, for all possible spatial locations, l_1 and l_2, by summing over the posterior probability of the sequences that have these spatial locations as their first or second targets. The decoder then picks the spatial location with the highest first target (second target) posterior, or equivalently likelihood in our design, as its first (second) target prediction.

Comparison of the first target decoding accuracies in the single-target and dual-target tasks. To find the first target decoding accuracy of the single-target task we modeled the activity of each neuron under any given single target location as an inhomogeneous Poisson process, which was fitted using the EM procedure. We then performed the maximum-likelihood decoding analysis using leave-one-out cross-validation on the single-target trials. To make the comparison, for the dual-target task we constructed two models one for the first target and one for the second target, and then performed the decoding analysis for each target separately.

BMI model training. In each BMI recording session, the monkeys first performed the dual-target task using a joystick (training session) during which models were constructed for the neuronal activity during an 800 ms time window prior to presentation of the "go" cue. This window length was chosen because in the standard dual-target sessions, it was sufficient to achieve better than 95% of the (maximum) sequence accuracy possible when using the entire window starting from second target presentation until the "go" cue (**Supplementary Fig. 4**). We modeled the activity of each neuron in this window under any sequence as a homogeneous Poisson process (point process with constant rate), instead of an inhomogeneous one, to make the

model construction faster for the BMI experiments. Hence using (1) the likelihood function for the spiking activity of neuron c under any of the sequences, S_i, was modeled as[43, 44]

$$p(N_{1:K}^c \mid S_i) = \prod_{k=1}^{K} (\lambda_c(S_i)\Delta)^{N_k^c} \exp(-\lambda_c(S_i)\Delta),$$

where $\lambda_c(S_i)$ denotes the fitted firing rate of that neuron in the 800 ms window for sequence S_i and $K = 800/\Delta$ is the total number of bins in this period with bin width $\Delta = 5$ ms. The firing rates were fitted using maximum likelihood parameter estimation. The training sessions were followed by the real-time BMI sessions in which these trained Poisson models were used to predict the sequence using the maximum-likelihood decoder.

Concurrent online predictions and movement execution in the BMI. After the training sessions, the monkeys performed the same task as before. However, this time, cursor position was controlled by target predictions made by the maximum-likelihood decoder rather than the joystick. During the real-time BMI experiments, individual spike timings of all cells within the population were continuously recorded at a 40 kHz resolution by the Plexon multi-channel acquisition processor. Each recorded spike was then transmitted via an Ethernet port to a separate computer running a Matlab routine in real time. For each real-time trial, the Matlab routine then used the maximum-likelihood decoder to calculate the likelihood of the population spiking activity during the 800 ms time window prior to the "go" cue, i.e., $N_{1:K}^{1:C}$, under each sequence, S_i. This likelihood was calculated based on the trained Poisson models and assuming neurons were independent conditioned on the sequence. Hence the population likelihood for each sequence was found as

$$p(N_{1:K}^{1:C} \mid S_i) = \prod_{c=1}^{C} \prod_{k=1}^{K} (\lambda_c(S_i)\Delta)^{N_k^c} \exp(-\lambda_c(S_i)\Delta)$$

The maximum likelihood decoder then outputted the sequence under which the population likelihood was maximized as the decoded sequence.

Based on the sequence decoded, a second Matlab routine running on the same computer then activated an analog output channel on the NI DAQ I/O interface to go from 0V to either +5V or -5V for 500 ms. The voltage line was, in turn, connected to a second NI DAQ I/O input channel

located on a third computer running the behavioral program. Depending on the voltage received, the cursor displayed in the middle of the screen moved in a straight line to one of the four possible target locations (e.g., +5V in I/O channel 1 corresponded to a cursor location within the top target). This process then repeated for the second decoded target after another artificially introduced time delay. Here, the time delays in the two generated movements were selected to be similar to those that the monkeys normally experienced when performing the standard task using a joystick. However, the NI DAQ could in principle generate the two movements in as little as 1 ms apart.

References

1. Chapin, J. K., Moxon, K. A., Markowitz, R. S. & Nicolelis, M. A. L. Real-time control of a robot arm using simultaneously recorded neurons in the motor cortex. *Nat. Neurosci.* **2**, 664–670 (1999).

2. Wessberg, J. et al. Real-time prediction of hand trajectory by ensembles of cortical neurons in primates. *Nature* **408**, 361–365 (2000).

3. Serruya, M. D., Hatsopoulos, N. G., Paninski, L., Fellows, M. R. & Donoghue, J. P. Instant neural control of a movement signal. *Nature* **416**, 141–142 (2002).

4. Hochberg, L. R. et al. Neuronal ensemble control of prosthetic devices by a human with tetraplegia. *Nature* **442**, 164–171 (2006).

5. Carmena, J. M. et al. Learning to control a brain-machine interface for reaching and grasping by primates. *PLoS Biol.* **1**, 193–208 (2003).

6. Taylor, D. M., Tillery, S. I. H. & Schwartz, A. B. Direct cortical control of 3D neuroprosthetic devices. *Science* **296**, 1829–1832 (2002).

7. Ganguly, K. & Carmena, J. M. Emergence of a stable cortical map for neuroprosthetic control. *PLoS Biol.* **7**, 1–13 (2009).

8. Wolpaw, J. R. & McFarland, D. J. Control of a two-dimensional movement signal by a noninvasive brain-computer interface in humans. *Proc. Natl. Acad. Sci. USA* **101**, 17849–17854 (2004).

9. Velliste, M., Perel, S., Spalding, M. C., Whitford, A. S. & Schwartz, A. B. Cortical control of a prosthetic arm for self-feeding. *Nature* **453**, 1098–1101 (2008).

10. Moritz, C. T., Perlmutter, S. I. & Fetz, E. E. Direct control of paralysed muscles by cortical neurons. *Nature* **456**, 639–643 (2008).

11. Mulliken, G. H., Musallam, S. & Andersen, R. A. Decoding trajectories from posterior parietal cortex ensembles. *J. Neurosci.* **28**, 12913–12926 (2008).

12. Kim, S.-P., Simeral, J. D., Hochberg, L. R., Donoghue, J. P. & Black, M. J. Neural control of computer cursor velocity by decoding motor cortical spiking activity in humans with tetraplegia. *J. Neural Eng.* **5**, 455–476 (2008).

13. Li, Z. et al. Unscented Kalman filter for brain-machine interfaces. *PLoS ONE* **4**, 1–18 (2009).

14. Chase, S. M., Schwartz, A. B. & Kass, R. E. Bias, optimal linear estimation, and the differences between open-loop simulation and closed-loop performance of spiking-based brain-computer interface algorithms. *Neural Networks* **22**, 1203–1213 (2009).

15. Musallam, S., Corneil, B. D., Greger, B., Scherberger, H. & Andersen, R. A. Cognitive control signals for neural prosthetics. *Science* **305**, 258–262 (2004).

16. Santhanam, G., Ryu, S. I., Yu, B. M., Afshar, A. & Shenoy, K. V. A high-performance brain-computer interface. *Nature* **442**, 195–198 (2006).

17. Kurata, K. Premotor cortex of monkeys: Set- and movement-related activity reflecting amplitude and direction of wrist movements. *J. Neurophysiol.* **69**, 187–200 (1993).

18. Messier, J. & Kalaska, J. F. Covariation of primate dorsal premotor cell activity with direction and amplitude during a memorized-delay reaching task. *J. Neurophysiol.* **84**, 152–165 (2000).

19. Crammond, D. J. & Kalaska, J. F. Modulation of preparatory neuronal activity in dorsal premotor cortex due to stimulus-response compatibility. *J. Neurophysiol.* **71**, 1281–1284 (1994).

20. Boussaoud, D. & Bremmer, F. Gaze effects in the cerebral cortex: reference frames for space coding and action. *Exp. Brain Res.* **128**, 170–180 (1999).

21. Crammond, D. J. & Kalaska, J. F. Differential relation of discharge in primary motor cortex and premotor cortex to movements versus actively maintained postures during a reaching task. *Exp. Brain Res.* **108**, 45–61 (1996).

22. Crammond, D. J. & Kalaska, J. F. Prior information in motor and premotor cortex: Activity during the delay period and effect on pre-movement activity. *J. Neurophysiol.* **84**, 986–1005 (2000).

23. Crutcher, M. D., Russo, G. S., Ye, S. & Backus, D. A. Target-, limb-, and context-dependent neural activity in the cingulate and supplementary motor areas of the monkey. *Exp. Brain Res.* **158**, 278–288 (2004).

24. Batista, A. P. & Andersen, R. A. The parietal reach region codes the next planned movement in a sequential reach task. *J. Neurophysiol.* **85**, 539–544 (2001).

25. Ninokura, Y., Mushiake, H. & Tanji, J. Representation of the temporal order of visual objects in the primate lateral prefrontal cortex. *J. Neurophysiol.* **89**, 2868–2873 (2003).

26. Tanji, J. & Shima, K. Role for supplementary motor area cells in planning several movements ahead. *Nature* **371**, 413–416 (1994).

27. Shima, K., Isoda, M., Mushiake, H. & Tanji, J. Categorization of behavioural sequences in the prefrontal cortex. *Nature* **445**, 315–318 (2007).

28. Shima, K. & Tanji, J. Neuronal activity in the supplementary and presupplementary motor areas for temporal organization of multiple movements. *J. Neurophysiol.* **84**, 2148–2160 (2000).

29. Baldauf, D., Cui, H. & Andersen, R. A. The posterior parietal cortex encodes in parallel both goals for double-reach sequences. *J. Neurosci.* **28**, 10081–10089 (2008).

30. Averbeck, B. B., Sohn, J.-W. & Lee, D. Activity in prefrontal cortex during dynamic selection of action sequences. *Nat. Neurosci.* **9**, 276–282 (2006).

31. Mushiake, H., Saito, N., Sakamoto, K., Itoyama, Y. & Tanji, J. Activity in the lateral prefrontal cortex reflects multiple steps of future events in action plans. *Neuron* **50**, 631–641 (2006).

32. Ohbayashi, M., Ohki, K. & Miyashita, Y. Conversion of working memory to motor sequence in the monkey premotor cortex. *Science* **301**, 233–236 (2003).

33. Kettner, R. E., Marcario, J. K. & Port, N. L. Control of remembered reaching sequences in monkey. II. Storage and preparation before movement in motor and premotor cortex. *Exp. Brain Res.* **112**, 347–358 (1996).

34. Lu, X. & Ashe, J. Anticipatory activity in primary motor cortex codes memorized movement sequences. *Neuron* **45**, 967–973 (2005).

35. Nakajima, T., Hosaka, R., Mushiake, H. & Tanji, J. Covert representation of second-next movement in the pre-supplementary motor area of monkeys. *J. Neurophysiol.* **101**, 1883–1889 (2009).

36. Averbeck, B. B., Chafee, M. V., Crowe, D. A. & Georgopoulos, A. P. Parallel processing of serial movements in prefrontal cortex. *Proc. Natl. Acad. Sci. USA* **99**, 13172–13177 (2002).

37. Saito, N., Mushiake, H., Sakamoto, K., Itoyama, Y. & Tanji, J. Representation of immediate and final behavioral goals in the monkey prefrontal cortex during an instructed delay period. *Cereb. Cortex* **15**, 1535–1546 (2005).

38. Smith, A. C. et al. State-space algorithms for estimating spike rate functions. *Computational Intelligence and Neuroscience* **2010** (2010).

39. Shima, K. & Tanji, J. Both supplementary and presupplementary motor areas are crucial for the temporal organization of multiple movements. *J. Neurophysiol.* **80**, 3247–3260 (1998).

40. Tanji, J. Sequential organization of multiple movements: Involvement of cortical motor areas. *Annu. Rev. Neurosci.* **24**, 631–651 (2001).

41. Vogel, E. K. & Machizawa, M. G. Neural activity predicts individual differences in visual working memory capacity. *Nature* **428**, 748–751 (2004).

42. Halsband, U. & Freund, H.-J. Motor learning. *Curr. Opin. Neurobiol.* **3**, 940–949 (1993).

43. Brown, E. N., Barbieri, R., Eden, U. T. & Frank, L. M. Likelihood methods for neural data analysis. In*: Computational Neuroscience: A Comprehensive Approach*, (ed. Feng, J.) 253–286 (CRC Press, Boca Raton, 2003)

44. Truccolo, W., Eden, U. T., Fellows, M. R., Donoghue, J. P. & Brown, E. N. A point process framework for relating neural spiking activity to spiking history, neural ensemble, and extrinsic covariate effects. *J. Neurophysiol.* **93**, 1074–1089 (2005).

45. Smith, A. C. & Brown, E. N. Estimating a state-space model from point process observations. *Neural Comput.* **15**, 965–991 (2003).

46. Dempster, A. P., Laird, N. M. & Rubin, D. B. Maximum likelihood from incomplete data via the EM algorithm. *J. Roy. Statist. Soc. B* **39**, 1–38 (1977).

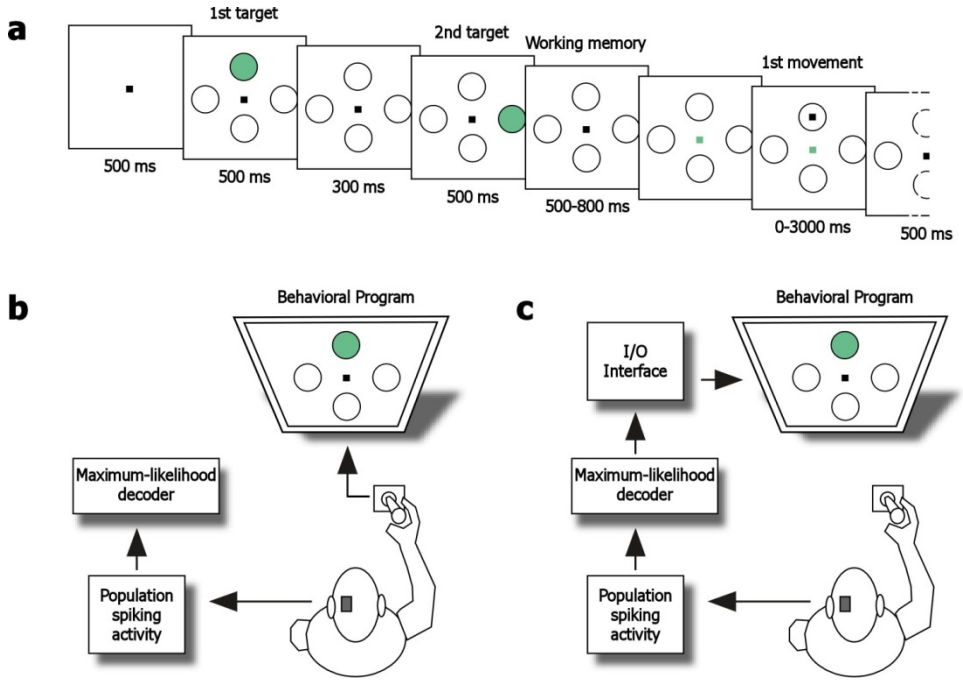

a
1st target
2nd target
Working memory
1st movement
500 ms
500 ms
300 ms
500 ms
500-800 ms
0-3000 ms
500 ms
b
Behavioral Program
Maximum-likelihood decoder
Population spiking activity
c
Behavioral Program
I/O Interface
Maximum-likelihood decoder
Population spiking activity

Figure 1

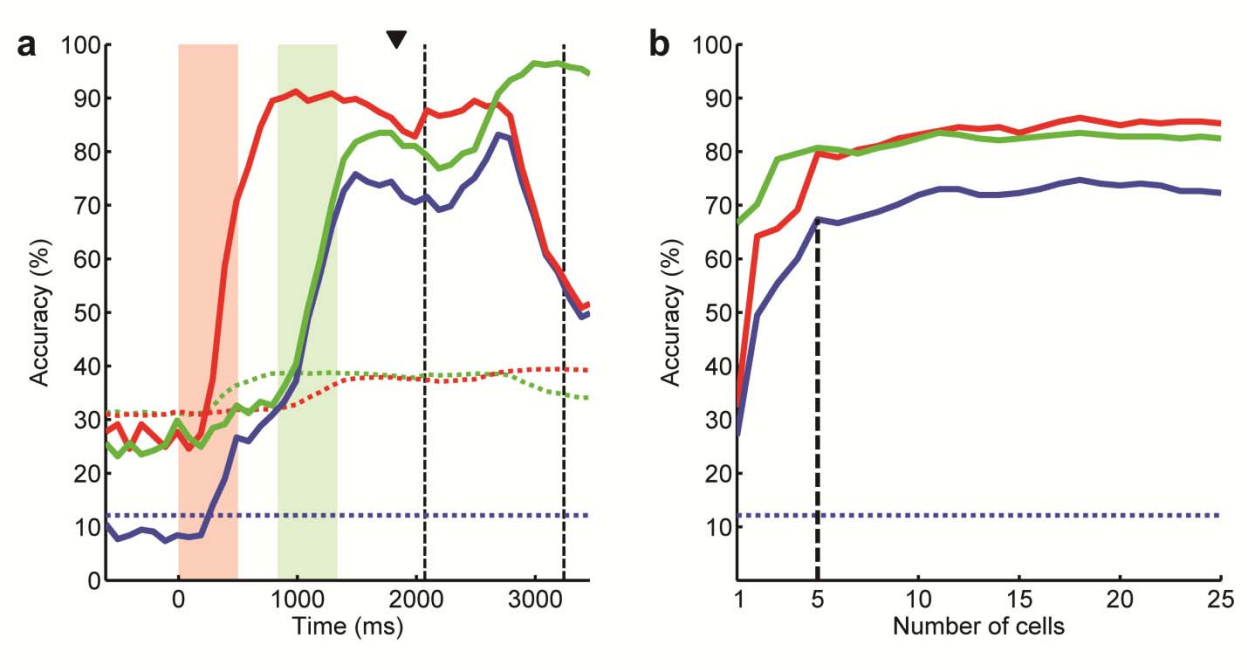

a
b
Accuracy (%)
Time (ms)
Number of cells
0
1000
2000
3000
1
5
10
15
20
25
10
20
30
40
50
60
70
80
90
100

Figure 2

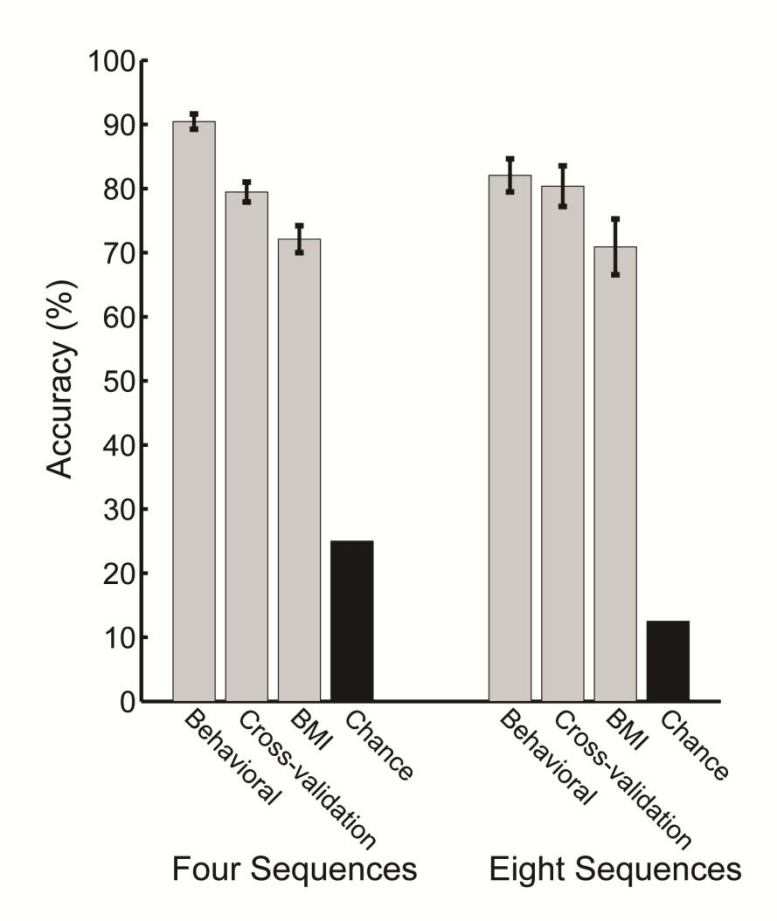

100
90
80
70
60
50
40
30
20
10
0
Accuracy (%)
Behavioral
Cross-validation
BMI
Chance
Behavioral
Cross-validation
BMI
Chance
Four Sequences
Eight Sequences

Figure 3

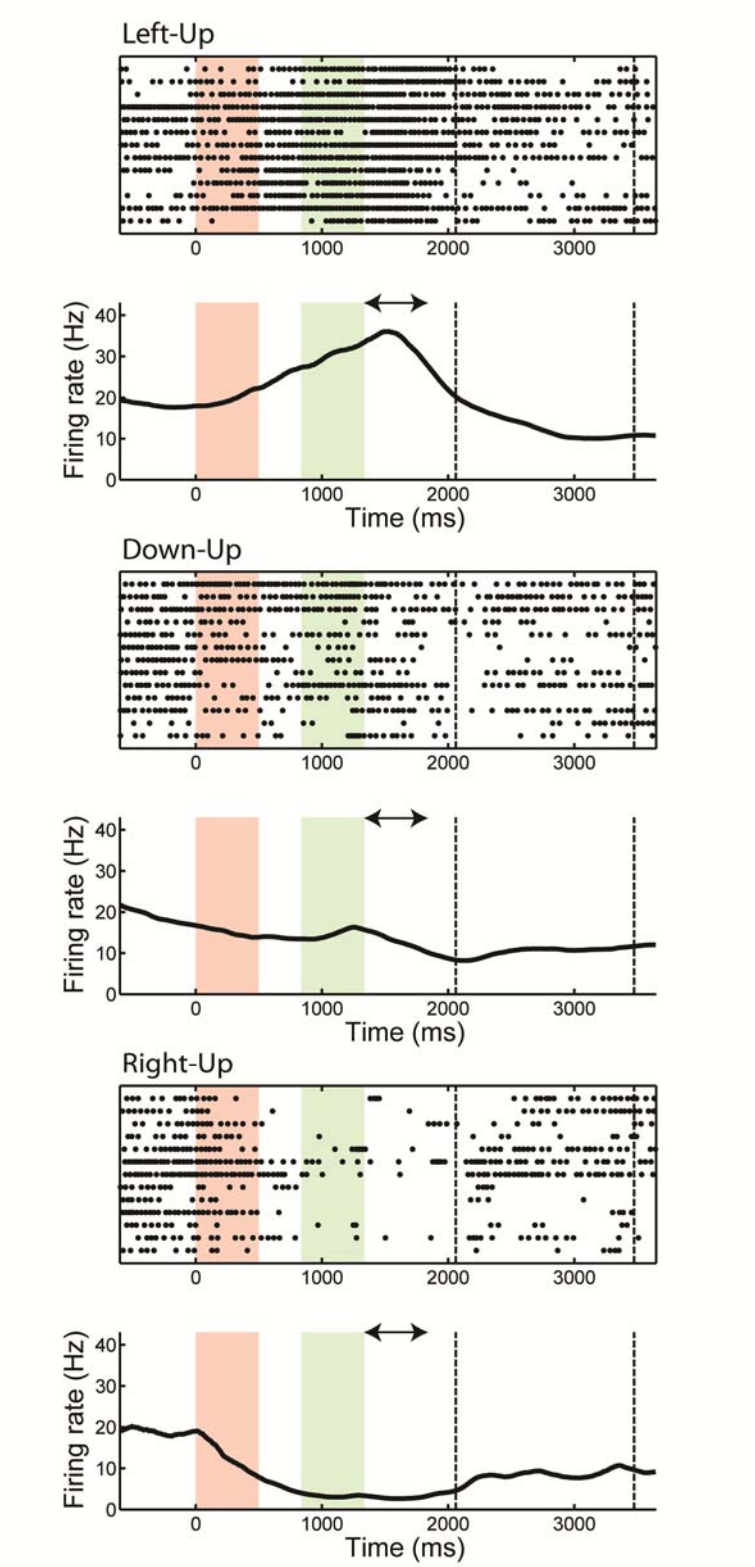

Left-Up
0
1000
2000
3000
Firing rate (Hz)
40
30
20
10
0
Time (ms)
Down-Up
Right-Up

Figure 4

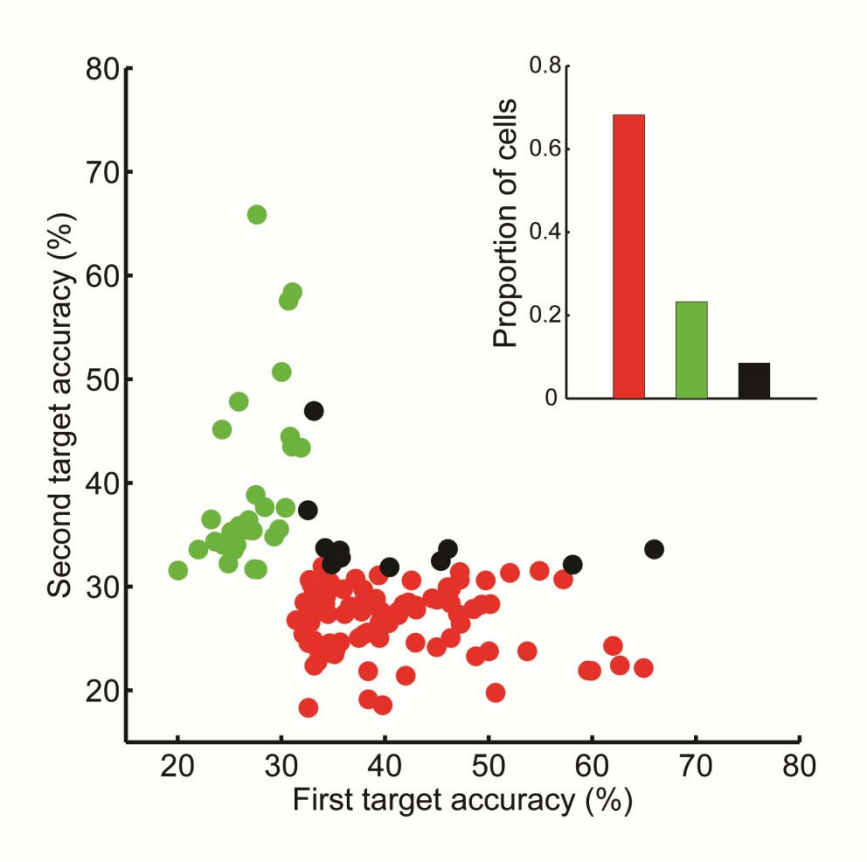

Second target accuracy (%)
First target accuracy (%)
Proportion of cells
80
70
60
50
40
30
20
0.8
0.6
0.4
0.2
0

Figure 5

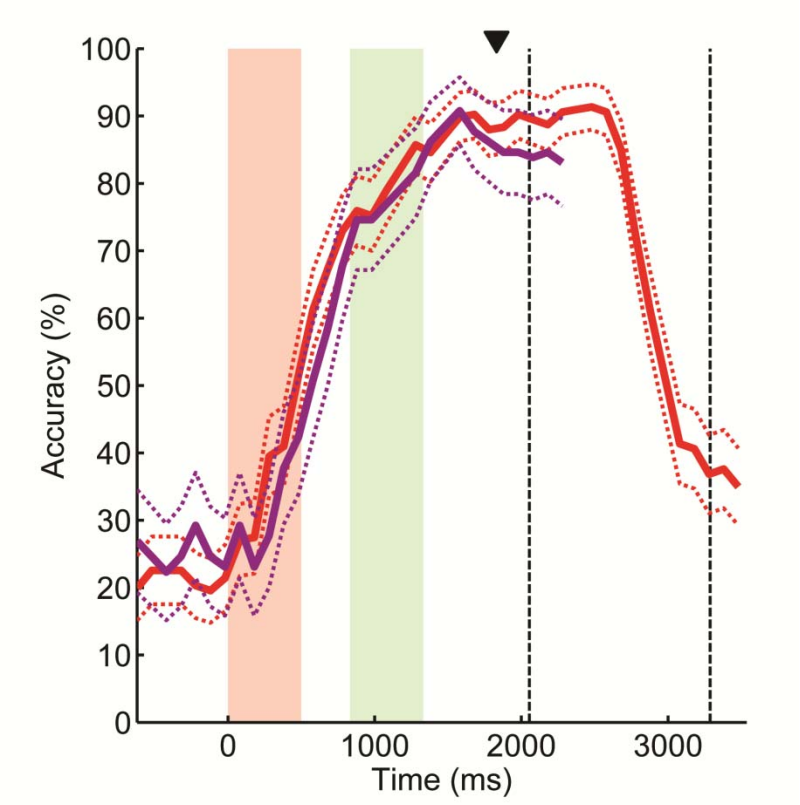


Figure 6

Supplementary Materials

A concurrent brain-machine interface for enhanced motor function

Maryam M. Shanechi, Rollin C. Hu, Marissa Powers, Gregory W. Wornell, Emery N. Brown & Ziv M. Williams

Supplementary Methods

Model construction using the expectation-maximization (EM) algorithm

To analyze the standard recording sessions, we model the activity of each neuron under any given sequence as an inhomogeneous Poisson process whose likelihood function (using the theory of point processes) is given by[43, 44]

$$p(N^c_{1:K} \mid S_i) = \prod_{k=1}^{K} (\lambda_c(k \mid S_i)\Delta)^{N^c_k} \exp(-\lambda_c(k \mid S_i)\Delta) \qquad i = 1{:}12\,, \qquad (1)$$

where Δ is the time increment taken to be small enough to contains at most one spike, N^c_k is the binary spike event of the c'th neuron in the time interval $[(k-1)\Delta, k\Delta]$, $\lambda_c(k \mid S_i)$ is its instantaneous firing rate in that interval, S_i is the i'th sequence, and K is the total number of bins in a duration $K\Delta$.

For each sequence and neuron, we need to estimate the firing rate $\lambda_c(k \mid S_i)$ using the neuronal data observed. One way to do so is to bin the data into non-overlapping windows of fixed length during which the firing rate is assumed to be constant and estimated using maximum likelihood techniques. This method is equivalent to finding the peristimulus time histogram (PSTH) that simply averages the number of spikes over any given window. The main drawback of this technique is that unless there are a large number of training trials under each sequence, to get a good estimate one has to pick relatively large windows. This in turn masks the fine-scaled evolution of the firing rate. Also, there is no principled way for selecting a window size, which the analysis is dependent on. One way to avoid these problems in estimating the spike rate function is to use a state-space approach[38, 45]. This approach is used in many applications to

estimate an unobservable state process and consists of two models: A prior or state model that in general enforces any prior information available about the unobservable states—such as a simple continuity condition—and an observation model that relates the neuronal observations to these states. In the case of estimating the spike rate function, and since it is a non-negative quantity, similar to previous work[38, 45] we take the state at time increment k, x_k, to be the logarithm of the firing rate, i.e., $x_k = \log(\lambda_c(k \mid S_i))$, or equivalently

$$\lambda_c(k \mid S_i) = \exp(x_k), \tag{2}$$

and enforce a continuity condition on it by assuming that it evolves according to a linear first-order Gaussian model[38, 45],

$$x_k = x_{k-1} + \epsilon_k,$$

where ϵ_k is the zero-mean white Gaussian noise with variance σ_ϵ^2. The observation model is in turn given by substituting (2) in (1). Here, $\theta = \sigma_\epsilon^2$ is an unknown parameter of the model and should be estimated jointly with the state. Hence we use the expectation-maximization (EM) iterative algorithm to find the maximum likelihood estimate of θ and in turn estimate the firing rate[38, 45, 46]. Denoting the estimate of θ in the i'th iteration by $\theta^{(i)}$, its estimate in the $i+1$'th iteration after the maximization step is given by

$$\theta^{(i+1)} = \frac{1}{K}\left(\sum_{k=1}^{K} W_k + W_{k-1} - 2W_{k,k-1}\right), \tag{3}$$

where $W_k = E[x_k^2 \mid N_{1:K}^c; \theta^{(i)}]$ and $W_{k,k-1} = E[x_{k-1}x_k \mid N_{1:K}^c; \theta^{(i)}]$ are found from the forward filter, fixed-interval smoothing, and covariance recursive algorithms in the expectation step as follows. Assuming that there are J total trials and denoting the causal filter state estimate by $x_{k|k} = E[x_k \mid N_{1:k}^c; \theta^{(i)}]$ and its variance by $w_{k|k}$, and the smoothed state estimate by $x_{k|K} = E[x_k \mid N_{1:K}^c; \theta^{(i)}]$ and its variance by $w_{k|K}$, the recursions in the E-step are given by the forward filter recursions[38, 45, 47],

$$w_{k|k-1} = w_{k-1|k-1} + \theta^{(i)}$$
$$x_{k|k-1} = x_{k-1|k-1}$$
$$w_{k|k} = (w_{k|k-1}^{-1} + J\exp(x_{k|k-1})\Delta)^{-1}$$
$$x_{k|k} = x_{k|k-1} + w_{k|k}\sum_{j=1}^{J}(N_k^c(j) - \exp(x_{k|k-1})\Delta),$$

for $k = 1,\cdots,K$, where $N_k^c(j)$ is the spike event in trial j, and by the fixed interval smoothing recursions[38, 45, 48],

$$A_k = w_{k|k}w_{k+1|k}^{-1}$$
$$x_{k|K} = x_{k|k} + A_k(x_{k+1|K} - x_{k+1|k})$$
$$w_{k|K} = w_{k|k} + A_k^2(w_{k+1|K} - w_{k+1|k}),$$

for $k = K-1,\cdots,0$ and with initial condition $x_{K|K}$ and $w_{K|K}$ from the filter recursions. We pick the initial conditions for the forward filter at each iteration of the EM algorithm as $x_{0|0}^{(i+1)} = x_{0|K}^{(i)}$ and $w_{0|0}^{(i+1)} = w_{0|K}^{(i)}$. Finally the state-space covariance algorithm gives all the terms needed for the M-step to find $\theta^{(i+1)}$ in (3) using these recursions[38, 45, 49],

$$W_k = w_{k|K} + x_{k|K}^2,$$

for $k = 0,\cdots,K$ and

$$W_{k+1,k} = A_k w_{k+1|K} + x_{k|K}x_{k+1|K},$$

for $k = 0,\cdots,K-1$. The iterations of the EM algorithm are run until convergence. The estimated firing rate at any time bin $k = 1,\cdots,K$ is in turn the smoothed estimate, $\hat{\lambda}_c(k \mid S_i) = \exp(x_{k|K})$ evaluated at the estimate of θ in the final iteration.

Repeating this procedure for all neurons under each sequence and fitting the inhomogeneous Poisson models results in a continuous smoothed estimate of the rate function for each neuron under any given sequence and over the entire length of a trial. Our implementation of the EM algorithm is similar to prior work[38, 45] (see also Supplementary Results for a comparison with PSTH and **Supplementary Figs. 10–12**).

Testing the decoding performance

Chance level accuracy

For the sequence, the chance level accuracy is simply $1/S$, where S is the number of sequences used. For targets, however, one has to take into account the correlation between the first and second targets when calculating the chance level accuracy. This is because depending on the number of sequences used in the decoding analysis, the first and second decoded targets may not be independent. In the case of 12 sequences, for example, since both targets cannot be at the same location, information about one also implies some information about the other. These correlation effects must therefore be taken into account when calculating the chance level accuracy of the targets.

We define indicator functions for the first and second targets, denoted by I_1 and I_2, that are 1 if the corresponding targets are decoded correctly and 0 otherwise. We show this analysis for the case when 12 sequences are used. In all other cases they can be found similarly. For 12 sequences, using the total law of probability, the probability that the second target is decoded correctly is given by,

$$p(I_2 = 1) = p(I_2 = 1 \mid I_1 = 1)\, p(I_1 = 1) + p(I_2 = 1 \mid I_1 = 0)\, p(I_1 = 0)\,.$$

Now if the first target is correctly decoded, the second target could be at one of three possible locations as the two targets cannot be at the same location. Hence the chance level accuracy in this case is given by $p(I_2 = 1 \mid I_1 = 1) = 1/3$. By a similar argument, if the first target is decoded incorrectly, the chance level accuracy of the second target is $p(I_2 = 1 \mid I_1 = 0) = 2/9$. Hence the chance level accuracy of the second target is given by

$$p_2^{chance} = \frac{2}{9} + \frac{1}{9}\, p(I_1 = 1)\,, \qquad (4)$$

and vice versa for the first target as the two targets are selected symmetrically in the choice of sequences. For example, in a session where we observe a first target accuracy of p^* in our decoding analysis, to assess the significance of the second target accuracy we should compare to $p_2^{chance} = 2/9 + p^*/9$ as opposed to simply $1/4$, using a one-sided *Z*-test for a Bernoulli process.

Note that if $p^* = 1/4$, i.e., at chance level, then $p_2^{chance} = 1/4$ and also at chance level as expected.

Testing the significance of first and second target encoding for each neuron

To test whether a given neuron encoded significant information about only the first target, only the second target, or both targets during the working memory period, we examined its decoding accuracy in this period for each target. Here, a neuron was considered to encode significant information about a target if its accuracy for the target was above the chance upper confidence bound. Since this involved two comparisons, one for the first and one for the second target, we used a Bonferroni correction for multiple comparisons reflected in calculating the chance upper confidence bounds.

Random permutation test: Testing significance for the divergence in the amount of information held by each cell about the two targets

To determine whether a significant divergence exists in the amount of information held by the premotor neurons about the two targets, we need to show that the absolute difference between the two target accuracies of each cell averaged across the population is significantly larger than that of a population with the same target accuracy values but with no structured relationship between each cell's target accuracies. To find the distribution of this average absolute difference in such a population with no structure, we keep the accuracy values the same but randomly permute them within the population, and repeat this process 100,000 times. This removes any possible structure between the target accuracies of each cell and hence creates a null hypothesis distribution. We can then establish the significance of the divergence by comparing the average absolute difference of the target accuracies of the premotor neurons against this null distribution and calculate a *P*-value.

To do so, we first correct for the correlation effect between the first and second target accuracy values, which is the byproduct of the choice of sequences used in the experimental design. Since the first and second targets cannot be at the same location within the set of 12 sequences, the

accuracy of one target has a contribution (even though fairly small) to the accuracy of the other even if the neuronal activity is not encoding that target *per se*. Hence we also need to remove this effect to see the true representation of a target by the neuronal activity, just as we take it into account in calculating the chance level accuracies. This means that we subtract from the first target accuracy value of each cell, the chance level contribution of that cell's second target accuracy, or $P_1^{chance} - 1/4$, and vice versa. We then randomly permute these first target accuracy values among the cells while keeping their second target accuracy values the same, and repeat this process many times. This generates a new population each time with the same accuracy values but no pair-wise structure between the first and second target accuracies of each cell. For each new population, we compute the average difference and after repeating this many times, find the distribution of this average difference. We then use this distribution to find whether the difference of the first and second target accuracies of the cells averaged over the premotor population is significantly different from a population with no structure (We find that the average divergence in the premotor population is significantly different from that of a population with no structure, with or without applying the correlation correction, $P < 10^{-15}$).

We corrected for the fairly small correlation effect between the first and second target accuracy values in the scatter plots of **Fig. 5** and **Supplementary Fig. 6** as explained for the random permutation test above.

Determining the number of cells required to achieve the population accuracy

We found the number of cells required to achieve a given percentage of the population accuracy by first sorting them in each session based on their single neuron sequence accuracies and then performing the decoding analysis in that session for different number of cells in descending order. We hence found the sequence accuracy as a function of the number of cells in descending order included in the decoding.

Determining stability of neuronal encoding selectivity for the first vs. second target over time

Since most cells encoded significant information about only one of the two targets, we were interested in examining whether this selectivity was 'hard-coded' to individual cells or whether cells altered their encoding selectivity over time (i.e., over multiple consecutive trials). To do so, we divided all sessions into halves and performed the model construction, leave-one-out cross-validation, and single neuron decoding analyses independently for each half.

BMI for predicting and executing two upcoming movements in real time

Selection of four and eight sequences

Each training session consisted of 26 ± 2 trials per sequence on average. Here, the task involved either four (both monkeys) or eight (monkey P) sequences. The four-sequence task consisted of either U-R, U-L, D-R, D-L or L-U, L-D, R-U, R-D. The eight-sequence task consisted of the union of the sequences in the two four-sequence tasks.

Although very high leave-one-out cross-validation accuracies were obtained for twelve sequences in the offline analysis on standard dual-target experiments, there were two limitations that led us to focus on four and eight sequences for the online BMI sessions. First, far more trials were required to train the models on all possible twelve sequences. On the test sessions in which all twelve sequences were attempted, the primates became satiated from juice reward and unwilling to continue performing experiments at the end of training. Second, the primates were sensitive to unexpected errors in trial performance. We specifically noticed that in standard experiments, the monkeys would often stop working when they got several trials incorrectly in a row. Therefore, even though their baseline performance on standard twelve sequence sessions was high, their performances on four or eight sequence sessions were higher. Since we expected BMI prediction accuracy to be somewhat lower than actual natural performance, especially for 12 sequences, we wanted to ensure that the monkeys do not notice a substantive difference between the normal task and BMI controlled tasks, and hence consistently perform the task.

Behavior versus prediction errors

Since the primates did not perform the dual-target task with 100% behavioral accuracy, some of the BMI errors were due to behavioral errors (i.e., the monkey not remembering the correct sequence) as opposed to decoder errors. Hence a more relevant accuracy number for the performance of the BMI could be the sequence accuracy obtained during the training session using leave-one-out cross-validation. This is because in the cross-validation analysis we calculate the accuracy by comparing the decoded sequence with the sequence the monkeys actually select after the "go" cue. For the BMI sessions, however, we compare the decoded sequence to the instructed sequence to find the accuracy. We hence tested whether after taking into account the primates' natural error rates, the accuracy during training sessions would be close to the BMI accuracy. Denoting the behavioral accuracy of the monkeys by P_b and the decoder accuracy found from the training session by P_t we can calculate what the accuracy of selecting the instructed sequence would be after taking into account the behavioral errors. Denoting the resulting accuracy by P_f we have that

$$P_f = P_b P_t + (1-P_b)(1-P_t)\times\frac{1}{S-1}.$$

In other words when the monkey and the decoder are both correct, the instructed sequence is selected. However, if the monkey is incorrect and the decoder is also incorrect in decoding the monkey's intended sequence, the probability of the decoder selecting the correct instructed sequence by random chance is $1/(S-1)$. We can find the mean and s.e.m. of P_f from those of P_b and P_t assuming P_b and P_t are independent[50] and then compare it with the BMI accuracy.

Supplementary Results

Comparison of homogeneous and inhomogeneous Poisson models

In the analysis of standard sessions, we used the EM procedure to estimate a smooth continuous firing rate function over the length of the trial and hence modeled the spiking activity as an inhomogeneous Poisson process for each cell and under each sequence. An alternative approach is to model the spiking activity during a given window of interest (e.g., the working memory

period) as a homogeneous Poisson process by assuming that the firing rate is constant in the given window and estimating it using maximum likelihood techniques. However, fitting the models as an inhomogeneous Poisson process has the advantage that a continuous rate function is estimated that is independent of the window size chosen for decoding. In other words, we can perform the decoding in any window length of interest without the need to refit a new model. Additionally, since the EM procedure uses the spiking activity across the whole length of the trial for the rate estimate of each single time point, it is potentially more accurate and moreover it does not mask the fine-scaled evolution of the firing rate. To examine this, we repeated the decoding analysis of all 11 standard sessions during the 500 ms working memory period using leave-one-out cross-validation, but this time fitting a homogenous Poisson process for the working memory period for each cell and under each sequence. Doing so, across sessions we had an average sequence decoding accuracy of 42% compared to 45% using the EM procedure. Hence the EM procedure produced more accurate decoding accuracies across sessions (paired *t*-test, $P < 0.005$).

Predictive relation between neural activity and upcoming behavior

As described in the main text, most neurons within the population encoded significant information about only one of the two targets within a sequence during the working memory period (i.e. either the first or the second). This suggested that the population was partitioned into two largely disjoint groups of neurons based on the type of information they encoded. It was not clear, however, what relation the activity in each group of cells had to upcoming behavior.

We find that the activity of each of the partitioned subpopulations of neurons was highly predictive of whether the primates were going to move to the correct target or not for only one of the two distinct targets. Cells that selectively encoded the first target location during the working memory period correctly predicted whether the primates were going to perform the correct first movement 73% of the time (one-sided *Z*-test, $P < 10^{-15}$). These neurons, however, were poor in predicting whether the primates would perform the second movement correctly (prediction accuracy of 39% compared with 99% upper confidence bound for chance of 38%). This was similarly true for neurons that selectively encoded the second target; they correctly predicted

whether the primates would perform the correct second movement 60% of the time (one-sided *Z*-test, $P < 10^{-15}$), but were poorly predictive of the primates' performance on the first movement (prediction accuracy of 42% compared with the 99% upper confidence bound for chance of 40%). This suggested that information encoded by each partitioned group of neurons was only predictive of the correct or incorrect execution of their corresponding (first or second) upcoming movement.

Controls for visual related responses

All of the main analyses focused on the working memory period during which no visual cues were presented. Nonetheless, we were also interested in examining whether some of the activity recorded during the first and second target presentation periods themselves could be attributable to a simple visual response to the target images. To investigate this question directly, for each cell, we compared the first and second target accuracies during their corresponding presentation periods. Since all four spatial locations (U,D,L,R) were used for both the first and second targets (across the session), the visual images shown during the two presentation periods were identical. Therefore, if neuronal responses were simply due to a visual response to the images, then there would be no significant difference between the two target accuracies in their presentation periods based on their order in the sequence. We found that 112 cells had significant target accuracies for at least one of the two visual targets during their presentation ($P < 0.01$). Of these cells, however, the vast majority (79%) had a significant target accuracy during presentation of only one of the two visual targets but not the other ($P < 0.01$), indicating that the activity during target presentation for most cells could not be attributed to a simple visual response.

Controls for potential movement related responses during working memory

Differences in the neuronal response across sequences during the working memory period were not associated with movement related activity. Monkeys were trained to move the joystick only after presentation of the "go" cue and any premature movement terminated the trial. Nonetheless, it was still possible that small unconstrained movements made by the animals

during the working memory period may have contributed to differences in neuronal activity observed. To examine this possibility directly we performed two control analyses.

First, we examined electromyography (EMG) activity recorded during the working memory period itself. EMG recordings were obtained from the contralateral arm (covering the deltoid, triceps and biceps muscles for one monkey) using tin surface electrodes. EMG signals were amplified and recorded on the Plexon work-station (0.7Hz – 300 Hz; 1 pole low-cut and 1 pole high-cut with 50x gain), and sampled at 1 kHz. EMG activity was set above a threshold 3 standard deviations above baseline, template-matched and clustered using principal component analysis. EMG activity was then used to decode the targets using the same maximum-likelihood approach as used for decoding the spiking activity. As expected, we found that EMG activity during the first movement period, which we defined as the 500 ms interval after the "go" cue, predicted the direction of limb movement for the first target with an accuracy significantly higher than chance (one-sided *Z*-test, $P = 0.01$). In contrast, EMG activity recorded during the working memory period carried no significant information about the first target (one-sided *Z*-test, $P = 0.14$). These findings suggest that any unconstrained movements potentially performed during working memory did not account for the selectivity observed by neuronal recordings.

Second, if neuronal activity during the working memory period was purely related to muscle movement, then cells that had a first target selective activity during the first reaching movement would also demonstrate such activity during the preceding working memory period. We hence examined what proportion of cells encoding the first target during the first movement period itself also encoded it during the working memory period. We found that 127 cells had a significant first target accuracy during the first movement period; of these, however, a considerable proportion (37%, 47 cells) did not have a significant first target accuracy during the working memory period ($P < 0.01$).

Even though the above controls strongly argue against the potential contribution of movement related activity during the working memory period, perhaps the most compelling argument comes from the main findings themselves. As described in more detail in the main text, cells across the population encoded significant information about both targets during the working

memory period by partitioning into two largely disjoint subpopulations, each encoding information about only one of the two targets. Therefore, for example, if cells responded purely to the first movement during the working memory period, then no cell would encode the second target during this period and *vice versa*. In other words, if neuronal responses during the working memory period were a direct reflection of the muscle movements made by the animals (whether first or second), then the population as a whole would only reflect that single movement. We find, however, that the population encoded significant information about both targets in the single window in time during the working memory period.

Effect of training set size and number of targets on the neuronal selectivity

To explore whether the number of sequences in the training set altered the neuronal responses, we conducted an experiment where one monkey performed a task in which only four sequences were presented (R-U, R-D, L-U, L-D). We then compared the sequence decoding accuracy in this task with that calculated over a subset of trials in the 12 sequence dual-target task (on the same day) in which the same 4 sequences were presented. This means that over this subset of trials, we were decoding one out of four possible sequences. We used leave-one-out cross-validation for all these analyses. We found that the population sequence decoding accuracy for the overlapping sequences did not significantly change even when the monkey was behaviorally performing a few select sequences vs. the full set (χ^2 test, $P = 0.18$). When similarly considering the sequence decoding accuracy of the individual cells, the vast majority (85%) demonstrated no significant change when performing 4 vs. 12 sequence sets (χ^2 test, $P > 0.05$).

To further examine whether neuronal encoding of the first target was affected by the number of targets (i.e., one target versus two sequentially presented targets) presented in a single session, we had one monkey perform only single-target trials. We then compared these single-target only sessions with sessions in which single-target trials were interleaved with dual-target trials on the same day. In other words, it was possible that expecting an additional sequential target to appear in some trials in the interleaved sessions influenced the encoding of the first target. We found, however, that there was no significant difference between the population decoding accuracies of

the first target on single-target trials between the two session types (χ^2 test; $P > 0.15$; **Supplementary Fig. 5**).

Speed-accuracy tradeoff in the BMI

We find that, using the maximum-likelihood decoder, sequence decoding accuracy for the set of four, eight, and twelve sequences reached 90% of the maximum asymptotic accuracy possible (i.e., accuracy achieved using the entire activity from the initial presentation of the second target until the "go" cue) on average after 488 ms, 561 ms, and 641 ms from the initial presentation of the second target, respectively (**Supplementary Fig. 9**). Once this accurate decoding is achieved, the BMI could, in principle, execute both targeted movements essentially instantaneously (1 ms apart). In comparison, even when excluding any time delays included in the task, the mean times it took for the monkeys to react to the "go" cue and reach the two targets was 791 ms. This is the case even though the monkeys were already aware of the target location by the time the "go" cue appeared and all delays including the time required to move between the targets were excluded. It is important to note that this comparison is relative to the time required to perform a task by natural sequential behavior as opposed to the time it takes the brain to encode a sequential motor plan. Obviously any decoder will be limited in speed by the encoding speed of the brain.

There are two important points to keep in mind when considering the potential of such a concurrent BMI to increase the speed at which sequential movements can be performed. Similar to natural behavior[51], there is a clear tradeoff in decoding between accuracy and speed. Even though significant information about both targets was available very shortly after the second target presentation, decoding accuracy continued to rapidly improve over time. Therefore, the speed at which such a BMI can perform sequential movements depends on the relative importance of accuracy vs. speed in a given task. Second, some tasks involve sequential movements that may be intrinsically performed faster than those tested here and, therefore, the benefit of using a concurrent BMI in such settings for speed enhancement depends on the natural reaction and movement times in the task.

Comparison across primates in the standard and BMI sessions

In the standard sessions, the first target, second target, and sequence accuracies were 76 ± 12%, 60 ± 19%, and 48 ± 13% for the first monkey and 74 ± 11%, 43 ± 3%, and 36 ± 3% for the second monkey, respectively. All these accuracies were significantly above chance for both monkeys (one-sided *Z*-test, $P < 10^{-15}$). The first monkey performed these standard sessions for 8 days and the second monkey performed them for 3 days. In the BMI sessions, the first monkey had an accuracy of 69 ± 3% and the second monkey had an accuracy of 75 ± 3% for the 4 sequence sets (only the second monkey performed the 8 sequence sets with results described in the main text). For the training sessions in the BMI experiments, the decoding accuracy for the 4 sequence sets for the first monkey was 77 ± 2% and for the second monkey was 82 ± 2%. These findings demonstrate that similar encoding strategies/profiles were present in both monkeys (**Supplementary Fig. 8**).

Supplementary References

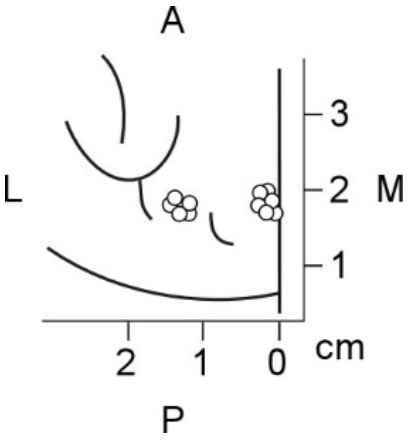
47. Eden, U. T., Frank, L. M., Barbieri, R., Solo, V. & Brown, E. N. Dynamic analysis of neural encoding by point process adaptive filtering. *Neural Comput.* **16**, 971–998 (2004).

48. Brown, E. N., Frank, L. M., Tang, D., Quirk, M. C. & Wilson, M. A. A statistical paradigm for neural spike train decoding applied to position prediction from ensemble firing patterns of rat hippocampal place cells. *J. Neurosci.* **18**, 7411–7425 (1998).

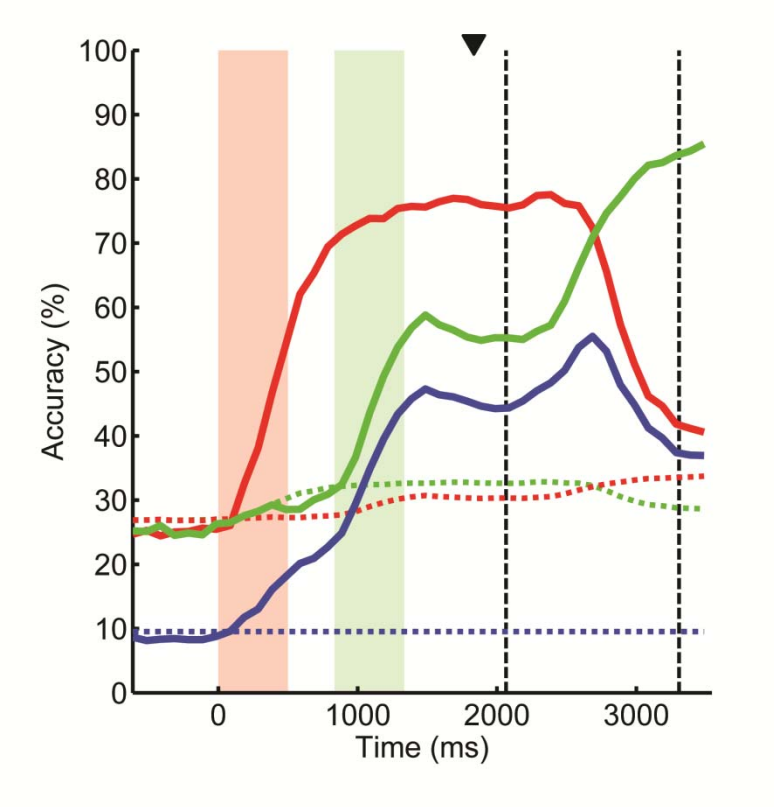
49. Jong, P. D. & Mackinnon, M. J. Covariances for smoothed estimates in state space models. *Biometrika* **75**, 601–602 (1988).

50. Goodman, L. A. On the exact variance of products. *J. Amer. Statistical Assoc.* **55**, 708–713 (1960).

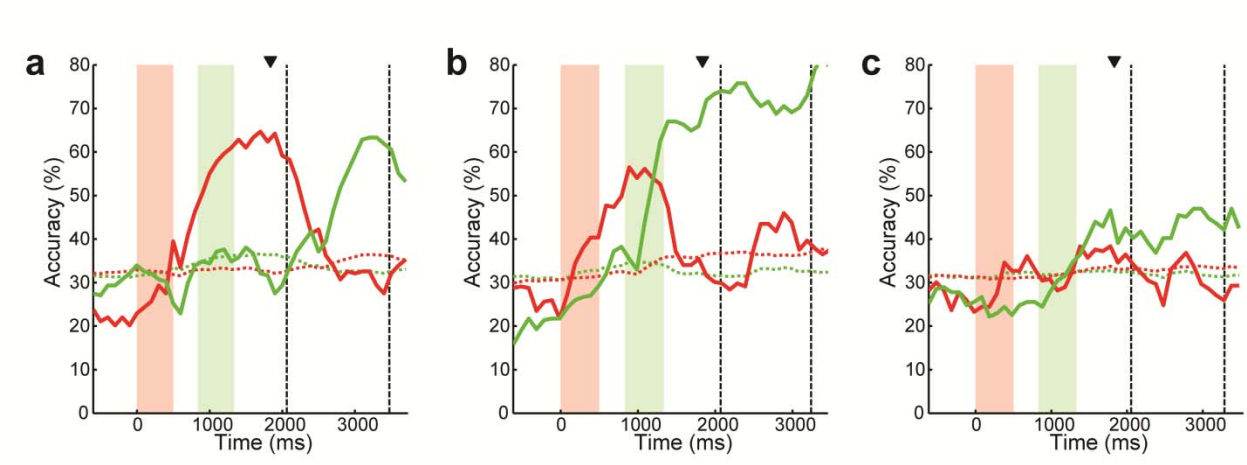
51. Bogacz, R., Wagenmakers, E.-J., Forstmann, B. U. & Nieuwenhuis, S. The neural basis of the speed-accuracy tradeoff. *Trends Neurosci.* **33**, 10–16 (2010).



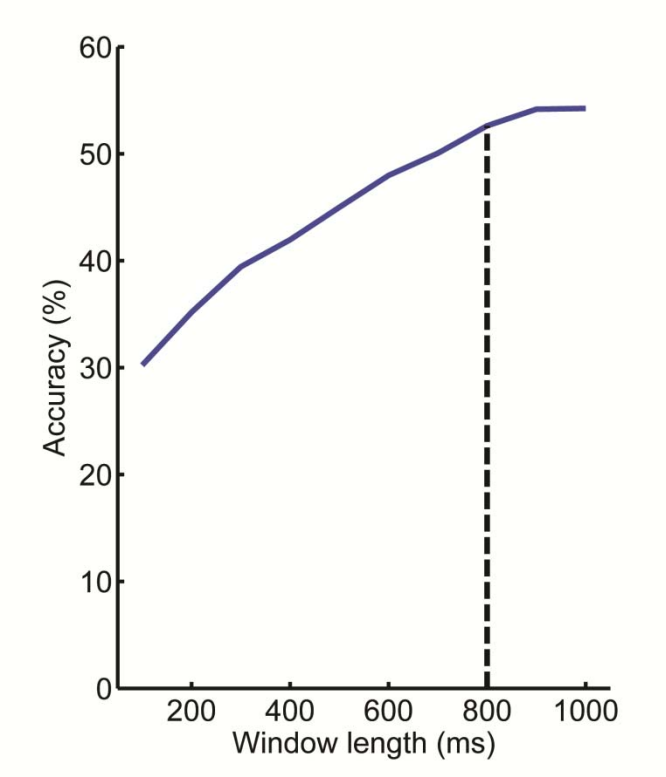
Supplementary Figure 1 Electrode recording sites. Top-view schematic of the electrode array positions. Each array (black circle) contains 36 electrode contracts. The bar in centimeters is referenced in relation to interaural antero-postero coordinates and midline medio-lateral coordinates. Here, A is anterior, P posterior, M medial and L lateral.



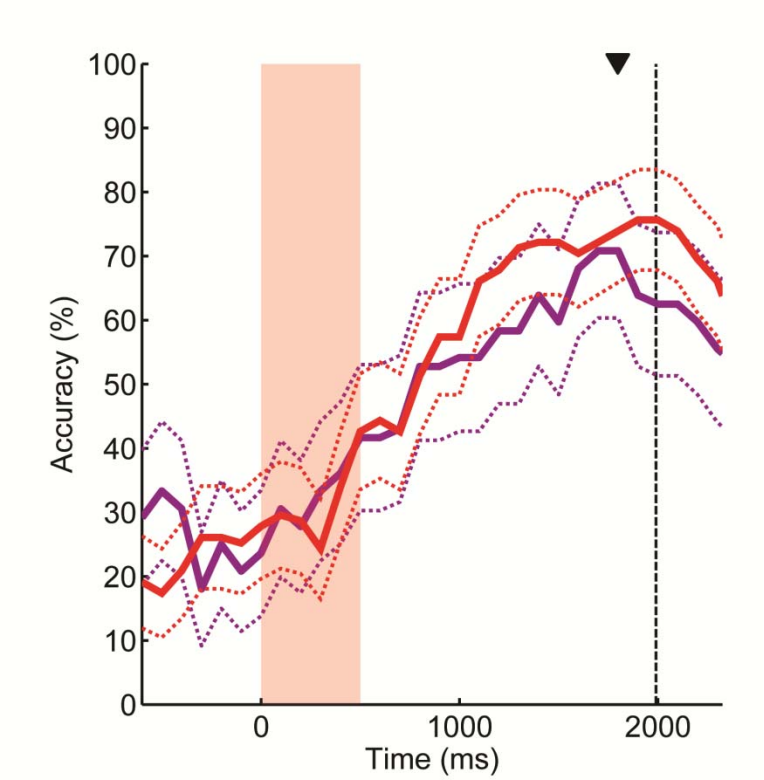
Supplementary Figure 2 Mean population decoding accuracy over time for all recorded sessions. Mean decoding accuracy for the population across all recorded sessions for the first target (red curve), second target (green curve) and the full sequence (blue curve). The figure has the same convention used in **Fig. 2**.



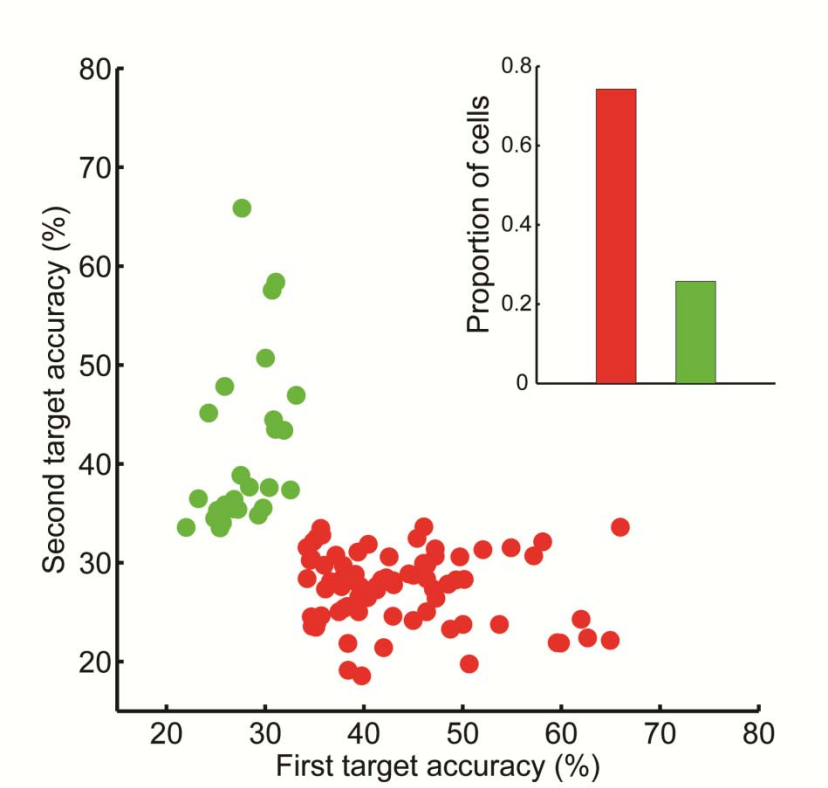
Supplementary Figure 3 Decoding accuracies over time for three sample cells that were selective for the first target only (**a**), second target only (**b**), and both targets (**c**) during the working memory period. Figure conventions are the same as in **Supplementary Fig. 2**.



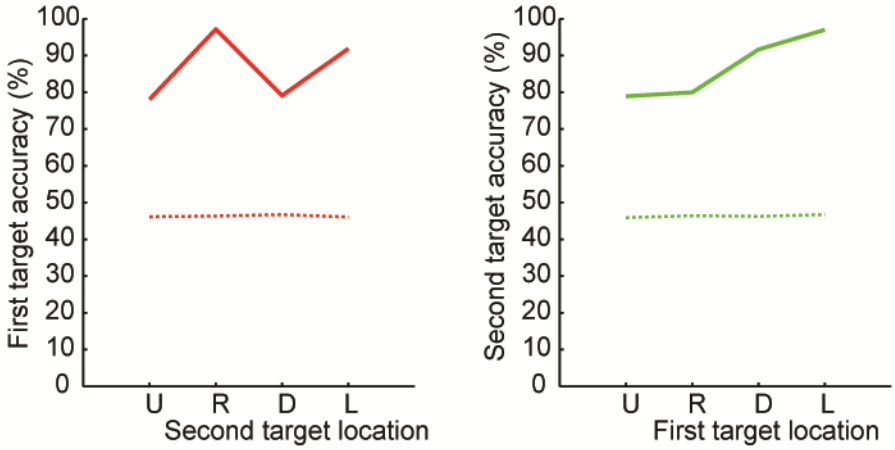
Supplementary Figure 4 Mean sequence decoding accuracy as a function of the time window length preceding the earliest "go" cue used in decoding. The blue curve shows the mean population sequence decoding accuracy (out of 12 possibilities) across all dual-target sessions. Using an 800 ms window, the sequence decoding accuracy exceeds 95% of the maximum possible when using the neuronal activity from the start of second target presentation until the earliest "go" cue (maximum of 1000 ms duration tested).



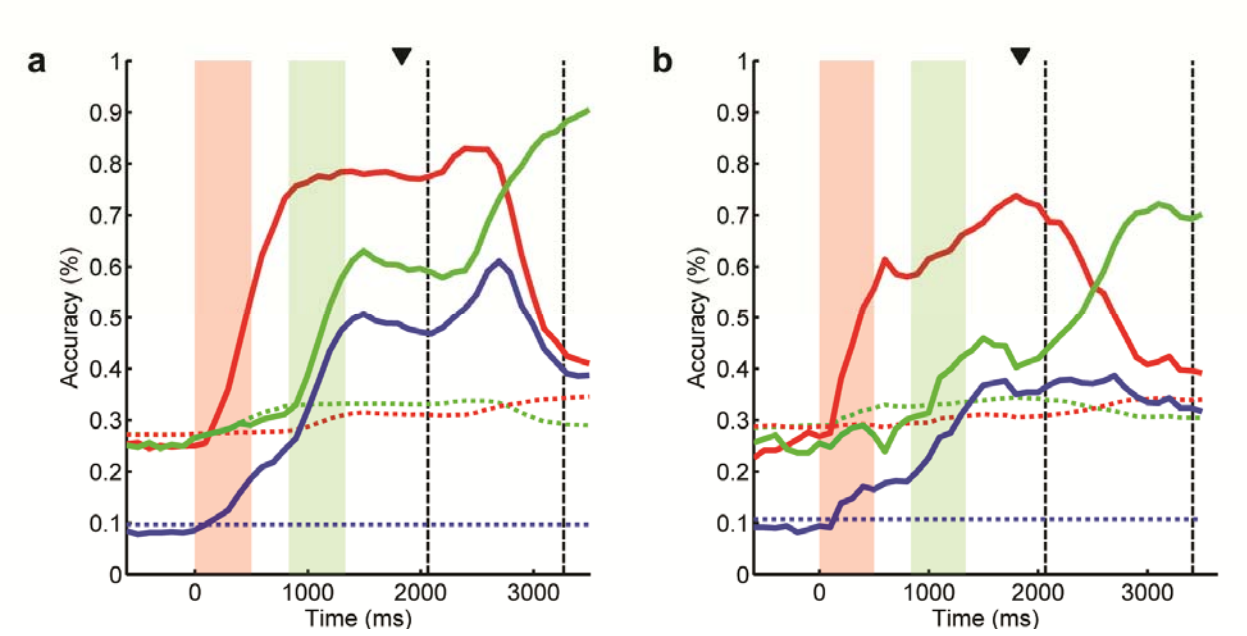
Supplementary Figure 5 Comparison of first target decoding accuracy for the population in interleaved and non-interleaved sessions. In the interleaved dual-target/single-target session, target decoding accuracy on single-target trials is shown in red. In the single-target only session, target decoding accuracy is shown in magenta. Each point on the curves indicates the decoding accuracy for the population over the *preceding* 500 ms window. Dotted lines indicate the 95% confidence bounds for each accuracy curve (rather than chance level). The red vertical bar indicates the time during which the (first) target was presented, and the vertical dotted line indicates the average time of the first "go" cue presentation onset. The arrow indicates the time point corresponding to the decoding accuracy of the preceding working memory period.



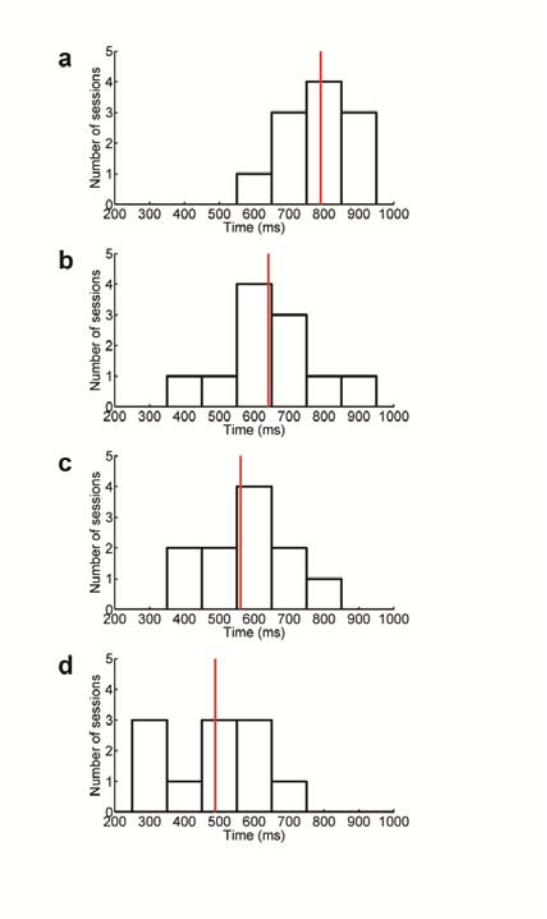
Supplementary Figure 6 Partitioning of the population during working memory. Scatter plot of the first and second target accuracies of the cells that significantly encoded at least one target during the working memory period. Statistical significance of the target accuracies was tested here at a stricter level ($P < 0.001$) than in **Fig. 5**. Red points indicate cells that significantly encoded only the first target and green points indicate those that significantly encoded only the second target. At this statistical level, no cell had a significant accuracy for both targets. The *inset* indicates the proportion of cells that significantly encoded only the first or only the second target during the working memory period with the same coloring schemes from left to right.



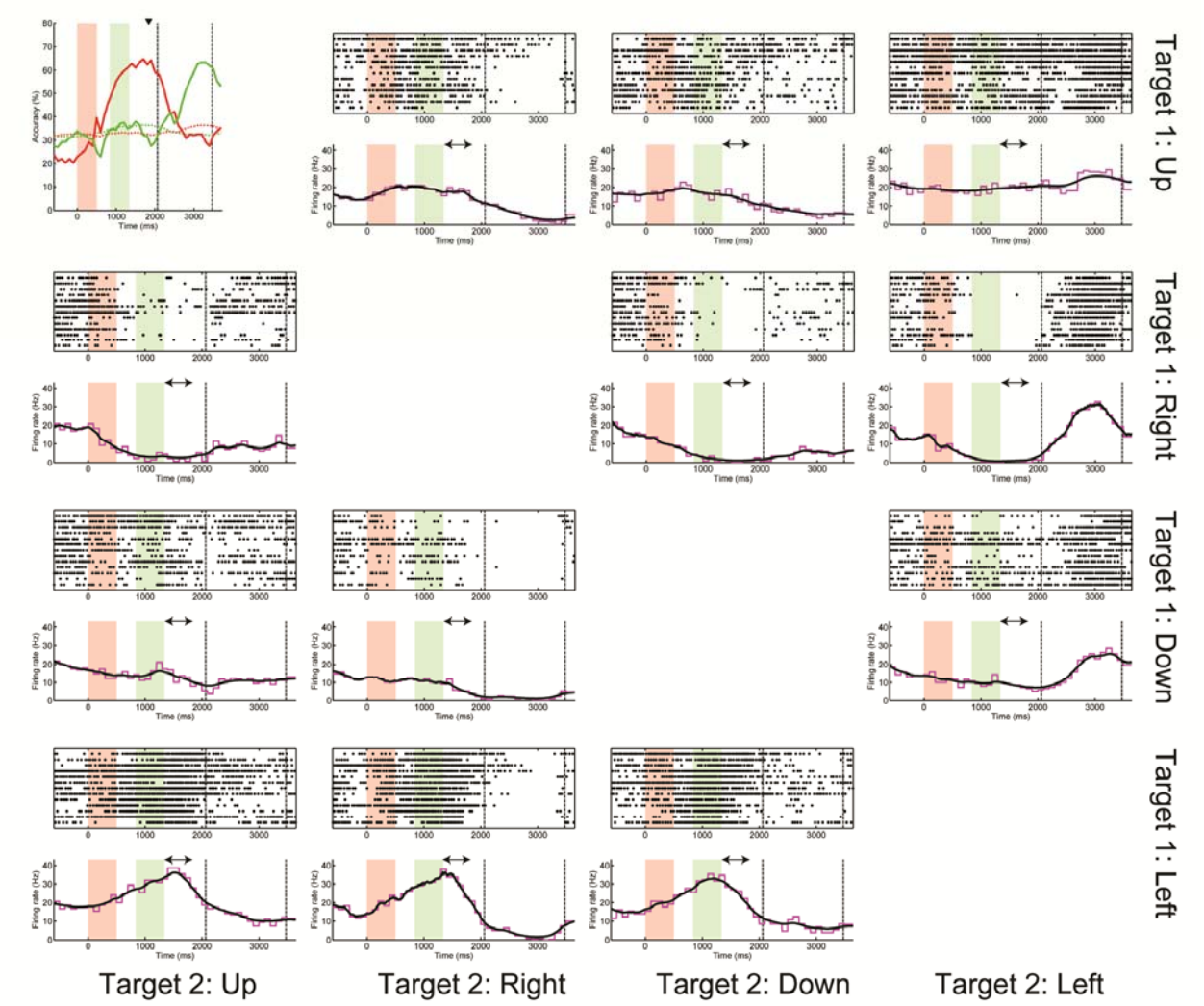
Supplementary Figure 7 Conditional decoding accuracies. The conditional first target accuracy given the possible second target locations (i.e., U, R, D, L) and the conditional second target accuracy given the possible first target locations are shown for the population in a sample session (same session as in **Fig. 2**). The dotted lines indicate the 99% chance upper confidence bounds.



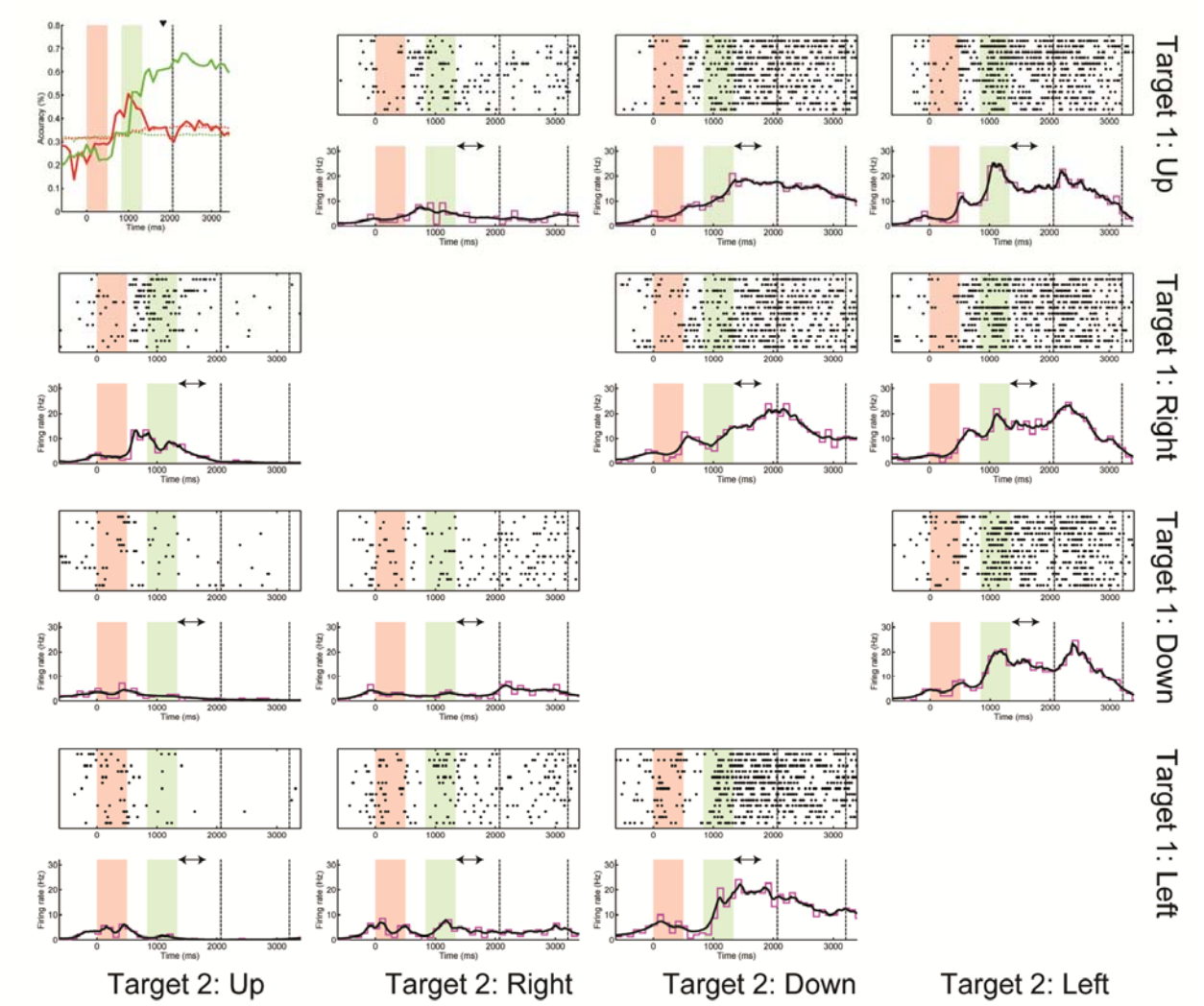
Supplementary Figure 8 Mean population decoding accuracy over time for all recorded sessions in monkey 1 (a) and monkey 2 (b). The figure has the same convention used in **Fig. 2**.



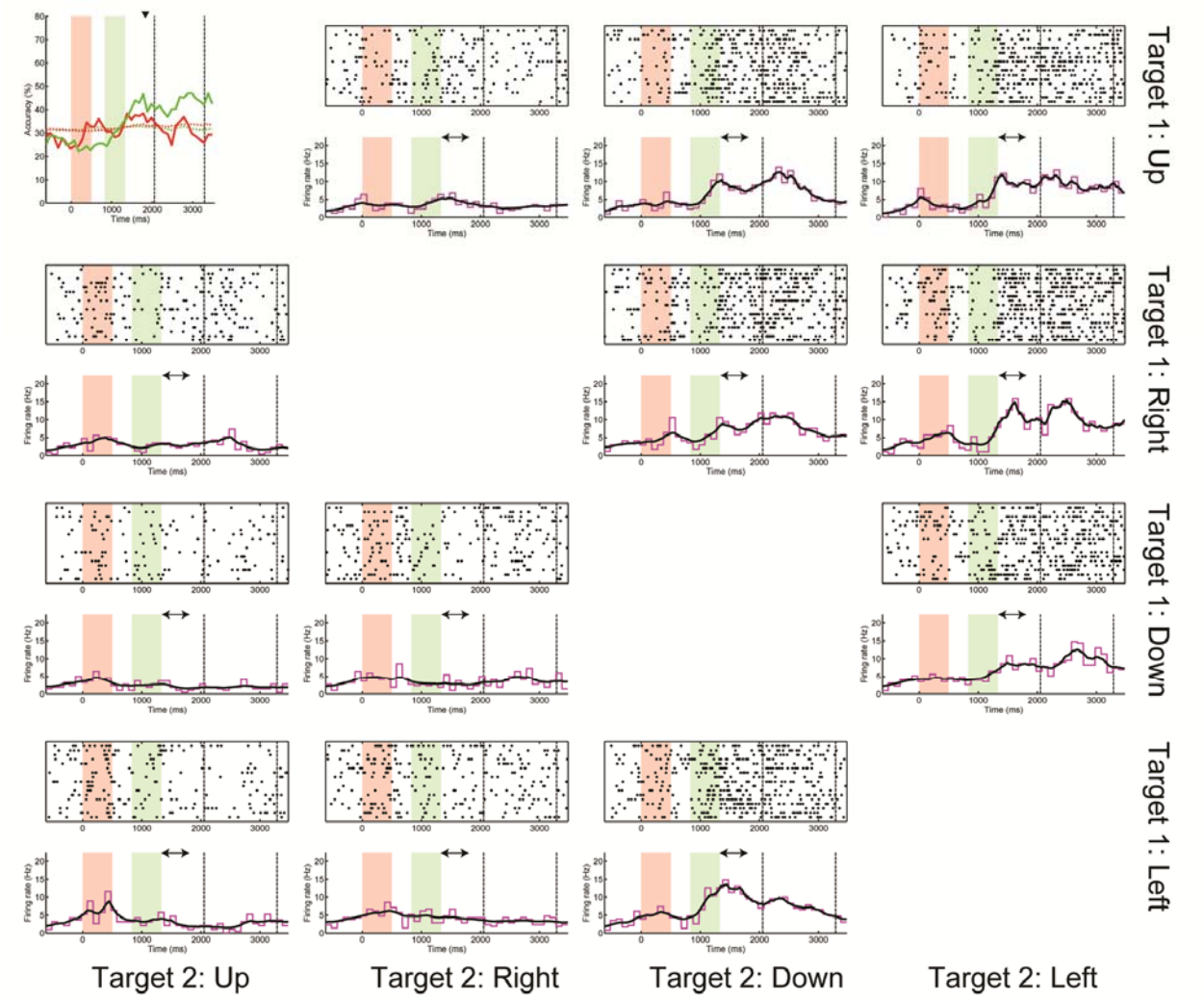
Supplementary Figure 9 Time histograms across the 11 sessions. The histogram of the time it took the monkeys to react to the "go" cues and reach both targets (i.e., excluding any task delays and the time required to move between targets) is shown in (a). The histogram of the times required after the start of second target presentation for the decoding accuracy to reach 90% of the asymptotic accuracy for 12, 8, and 4 sequences are presented in (b), (c), and (d), respectively. The red line shows the average time.



Supplementary Figure 10 Encoding of a first target selective neuron. The subfigure at the upper left corner shows the first and second target accuracies of the cell as a function of time into the trial. The vertical bars/lines and their timings follow the same convention as **Fig. 2**. In all other subfigures, each *top* panel corresponds to a different sequence of movements with each row illustrating the spiking activity during a single trial and the black dots indicating the spike times. Each *bottom* panel indicates the corresponding mean firing rate estimates using the expectation-maximization procedure (black curve) and the corresponding peristimulus time histogram (PSTH) (magenta curve). The arrow indicates the working memory period. The subfigures in the same row correspond to sequences with the same first target location. The subfigures in the same column correspond to sequences with the same second target location. Note that repeated targets locations were not used in the sequences and hence there are 3 subfigures per row/column.



Supplementary Figure 11 Encoding of a second target selective neuron. Figure has the same convention used in **Supplementary Fig. 10**.



Supplementary Figure 12 Encoding of a neuron selective for both targets. Figure has the same convention used in **Supplementary Fig. 10**.

Feedback-Controlled Parallel Point Process Filter for Estimation of Goal-Directed Movements from Neural Signals

Maryam M. Shanechi, *Member, IEEE,* Gregory W. Wornell, *Fellow, IEEE,* Ziv M. Williams, and Emery N. Brown, *Fellow, IEEE*

Abstract

Real-time brain-machine interfaces have estimated either the target of a movement, or its kinematics. However, both are encoded in the brain. Moreover, movements are often made to reach a target. Hence incorporating the target information in the kinematic decoder can increase its accuracy. Using an optimal feedback control design, we develop a recursive Bayesian kinematic decoder that combines the target information with the ensemble spiking activity during movement. To do so, we build a prior goal-directed state-space model for the movement using an optimal feedback control model of the sensorimotor system that aims to emulate the processes underlying actual motor control and takes into account the sensory feedback. Most goal-directed models, however, depend on the movement duration, not known a priori to the decoder. This has prevented their real-time implementation. To resolve this duration uncertainty, the decoder discretizes the duration and consists of a bank of parallel point process filters, each combining the prior model of a discretized duration with the neural activity. The kinematics are computed by optimally combining these filter estimates. Using the feedback-controlled model and even a coarse discretization, the decoder significantly reduces the root-mean-square (RMS) error in estimation of reaching movements performed by a monkey.

This work was presented in part at the IEEE Conference on Acoustics, Speech, and Signal Processing (ICASSP), Dallas, Texas, March, 2010.

M. Shanechi and G. Wornell are with the Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, 02139 USA e-mail: (shanechi@mit.edu,gww@mit.edu).

Z. Williams is with the Neurosurgery Department, Massachusetts General Hospital, Boston, MA, 02114 USA e-mail: (ZWILLIAMS1@partners.org)

E. Brown is with the Brain and Cognitive Sciences Department, Massachusetts Institute of Technology, Cambridge, MA, 02139 USA e-mail: (enb@neurostat.mit.edu).

Index Terms

Brain-machine interfaces, neural decoding, point process modeling, optimal feedback control.

I. Introduction

Information about various states of an intended movement is encoded in the neural activity from the motor cortical areas. Peri-movement activity, i.e., the activity around the time of movement, in the primary motor cortex, posterior parietal cortex (PPC), and dorsal premotor cortex (PMd) is related to the movement kinematics such as direction, velocity, position, and acceleration (see e.g., [1]–[5]). Also neural activity from the PPC and the premotor cortex is related to the intended target of the movement prior to movement initiation (see e.g., [6]–[8]). Brain-machine interfaces (BMIs) work by recording these neural activities such as the ensemble spiking activity, mapping these activities to the intended behavior also known as decoding, and using the decoded signal to control a device such as a robotic arm or computer cursor [5], [9]–[24].

Many typical tasks—such as reaching for a book—involve performing goal-directed movements. Most work on real-time BMIs have focused on individually decoding either the movement's kinematics to reconstruct its trajectory [5], [9]–[21], or the movement's intended reach goal or target [22], [23]. However, designing the kinematic decoder to take into account the goal-directed nature of such movements could improve their estimation accuracy. Additionally, when both target and kinematic related neural activities are recorded, combining them in the decoding algorithm could result in more accurate estimation of goal-directed movements. In particular, it has been shown that information about the intended target location can be decoded reliably from neural activity prior to movement [22], [23]. Hence the main question is how to combine the target information with the peri-movement activity to improve the accuracy of trajectory estimation. One way to do so is to incorporate the target information into the construction of a prior goal-directed state-space model for the kinematics. This prior model can in turn be used in a recursive Bayesian decoder to estimate the kinematics [17], [25]–[27].

Prior work have built this goal-directed model for reaching movements by conditioning a linear Gaussian state-space model, also known as a random-walk model, on being at the target at a *known* arrival time [26], [27] or using a linear feedforward controlled (i.e., not taking into account the sensory feedback) model again assuming a known arrival time [28]. Alternatively, goal-directed prior models have been built by using a training data set, for example fitting a linear Gaussian state-space model for a given target to empirical reaches to its location [25] or fitting a single model for arbitrary targets

based on a data set of reaches to their locations [17]. Since the goal of the decoder is to estimate natural movement, a prior model that aims to closely emulate the sensorimotor processing underlying actual motor control could result in more accurate estimation of intended movement. Hence we propose a prior model for goal-directed movements based on the optimal feedback control theory of the sensorimotor system [29], [30], used to interpret its function. This results in a feedback-controlled prior state-space model for the kinematics that exploits the information about the target location and models the sensory feedback. Also, unlike the models in [17], [25], this prior model does not rely on a training data set and can therefore easily extend to different target locations without requiring a set of empirical reaches to these locations. Finally, by using the optimal feedback control formulation, it can generalize to tasks other than reaching movements, if desired, by simply quantifying the goals of such tasks as the cost function in this formulation. We derive this optimal feedback-controlled prior model as the first component to develop a recursive Bayesian decoder for goal-directed movements.

Regardless of the approach taken to incorporate the target information into the prior state-space model, most goal-directed state-space models are inherently dependent on the arrival time at the target or equivalently on the movement duration. For example there is much more constraint on the movement kinematics close to the arrival time compared to far from it, as in the former case the trajectory soon needs to converge to the intended target. Also the assumption of known movement duration used in [26] and [28] prevents their implementation in real time since the duration is not known a priori to a real-time decoder observing only the neural signal. Hence to be implementable in a BMI, in addition to using a goal-directed state-space model, a practical decoder needs to address this uncertainty in movement duration. We introduced a framework to resolve this duration uncertainty in [31] by discretizing the movement duration and then optimally combining the kinematic estimates corresponding to the discretized durations based on the neural data. A similar approach using a discretized set of durations was subsequently used in [32] to resolve the duration uncertainty of the goal-directed prior model developed in [26] for estimation of simulated reaching trajectories. In this paper, as the second component of our decoder, we develop our framework to resolve the duration uncertainty for any goal-directed state-space model (including our feedback-controlled one) in more detail. Additionally, we examine the loss in estimation accuracy due to the unknown duration and the effect of the number of discretization points on this loss. Moreover, we discuss two possible approaches to treating a state-space model after its arrival time is reached and their effect on decoder accuracy.

Combining these two components, our decoder can be used in a real-time BMI to improve the estimation accuracy of goal-directed movements. Our focus in this paper is on the detailed derivation of the decoder

and the characterization of its performance for estimation of goal-directed movements performed by a rhesus monkey from simulated neural spiking activity. The simulated neural data is obtained based on the monkey's trajectories using a rigorously validated model of neural activity in the primary motor cortex [2], [33] that relates it to the intended movement parameters. Using thousands of simulated trials, this analysis allows us to characterize the decoder performance under various conditions and give guidelines for its implementation and parameter selection. An application of this decoder to combining target and kinematic related activities in a real-time BMI in an example experimental setup is presented in our work in [34].

II. METHODS

We denote the sequence of kinematic states by $\mathbf{x}_0, \cdots, \mathbf{x}_t$ and the spiking activity of the ensemble of C neurons by $\mathbf{N}_1, \cdots, \mathbf{N}_t$ where $\mathbf{N}_t = (N_t^1, \cdots, N_t^C)$ is the binary spike events of the C neurons at time t. We model the spiking activity of each neuron as a point process. Assuming that the neural spiking observations from the C neurons are conditionally independent given the state, the point process observation model is given by [33], [35]

$$p(\mathbf{N}_t|\mathbf{x}_t, \mathbf{H}_t) = \prod_c \left(\lambda_c(t|\mathbf{x}_t, \mathbf{H}_t^c)\Delta\right)^{N_t^c} e^{-\lambda_c(t|\mathbf{x}_t, \mathbf{H}_t^c)\Delta} \tag{1}$$

where $\mathbf{H}_t^c = N_{1:t-1}^c$, $\mathbf{H}_t = \mathbf{N}_{1:t-1} = \mathbf{H}_t^{1:C}$ model the history dependence of the spiking activity, Δ is the time increment, and $\lambda_c(t|\mathbf{x}_t, \mathbf{H}_t^c)$ is the modeled instantaneous firing rate or equivalently the conditional intensity function of the cth neuron at time t. Note that we have explicitly indicated the encoding of the kinematic states, $\mathbf{x}_t$, in the conditional intensity function (the form of which depends on the kinematics tuning model and will be discussed further in the Results section). If the observation model further assumes that the spiking activity is not history dependent, i.e., that the observation at each time step is conditionally independent of the observations in the previous time steps given the kinematic state, then the observation model simplifies to

$$p(\mathbf{N}_t|\mathbf{x}_t, \mathbf{H}_t) = p(\mathbf{N}_t|\mathbf{x}_t) = \prod_c \left(\lambda_c(t|\mathbf{x}_t)\Delta\right)^{N_t^c} e^{-\lambda_c(t|\mathbf{x}_t)\Delta}$$

The goal of the decoder is to causally calculate the state posterior density, i.e., $p(\mathbf{x}_t|\mathbf{N}_{1:t})$, based on the neural observations. We first develop the feedback-controlled prior state-space model. We then derive the decoder that combines this prior with the point process observation model and also resolves the duration uncertainty inherent to goal-directed state-space models.

A. Optimal Feedback-Controlled Prior State-Space Model

To build a prior goal-directed model that incorporates the target information, we develop an optimal feedback-controlled state-space model for the kinematics. This model is inspired by the optimal feedback control theory of the sensorimotor system that is used to interpret its function [29], [30]. In this framework, each task is performed to accomplish a goal during which there is sensory feedback about the state of the musculo-skeletal system. Based on the desired goal, the sensory feedback (for example vision and proprioception) about the current state of the system, and the intended time to accomplish the goal, the subject (controller) decides on the next control signal (for example muscle activation) to generate the movement. As a result, the subject can make real-time adjustments in the control signal based on the feedback to improve behavior. This framework is illustrated in Fig. 1. Once we specify an approximate kinematic model, quantify the task goals as cost functions, and model the sensory feedback, this framework can be used to predict the subject's next control signal. Therefore, we use this framework to develop the optimal feedback-controlled prior model to predict the next control signal in the decoder and consequently the next kinematic state.

We assume the sequence of kinematic states, $\mathbf{x}_0, \cdots, \mathbf{x}_t$, are generated according to the linear dynamical system

$$\mathbf{x}_{t+1} = \mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{u}_t + \mathbf{w}_t. \tag{2}$$

Here $\mathbf{u}_t$ is the control signal at time t, which is decided by the controller (the subject), $\mathbf{w}_t$ is the zero-mean white Gaussian state noise with covariance matrix $\mathbf{W}$, and $\mathbf{A}$ and $\mathbf{B}$ are parameters of the kinematic model. We assume that the subject has perfect sensory feedback of the kinematic states (for example proprioceptive or visual) and therefore the sensory feedback, $\mathbf{y}_t$, is noiseless, i.e., $\mathbf{y}_t = \mathbf{x}_t$. To find $\mathbf{u}_t$ in the control framework, we need to specify a cost function whose expected value will then be minimized by selecting the optimal $\mathbf{u}_t$. The cost function in a given task should quantify its goal. For the above linear Gaussian state-space model, if we pick the cost function as a quadratic function of the state and control signals given by

$$J = \sum_{t=1}^{T-1} \left(\mathbf{x}_t'\mathbf{Q}_t\mathbf{x}_t + \mathbf{u}_t'\mathbf{R}\mathbf{u}_t\right) + \mathbf{x}_T'\mathbf{Q}_T\mathbf{x}_T, \tag{3}$$

where T is the movement duration, $\mathbf{Q}_t$ is positive semidefinite and $\mathbf{R}$ is positive definite, then the optimal control signal at any time, $\mathbf{u}_t$, is simply a linear feedback of the state at that time given by [29], [36]

$$\mathbf{u}_t = -\mathbf{L}_t(T)\mathbf{x}_t, \tag{4}$$

where $\mathbf{L}_t$ can be found recursively and offline using [36]

$$\mathbf{L}_t = (\mathbf{R} + \mathbf{B}'\mathbf{P}_{t+1}\mathbf{B})^{-1}\mathbf{B}'\mathbf{P}_{t+1}\mathbf{A}. \tag{5}$$

Here $\mathbf{P}_t$ is in turn found recursively and backwards in time using

$$\mathbf{P}_t = \mathbf{Q}_t + \mathbf{A}'\left(\mathbf{P}_{t+1} - \mathbf{P}_{t+1}\mathbf{B}\left(\mathbf{R} + \mathbf{B}'\mathbf{P}_{t+1}\mathbf{B}\right)^{-1}\mathbf{B}'\mathbf{P}_{t+1}\right)\mathbf{A} \tag{6}$$

with the initial condition

$$\mathbf{P}_T = \mathbf{Q}_T. \tag{7}$$

This is the linear quadratic Gaussian (LQG) solution. Substituting (4) in (2) reduces this state-space model to the optimal feedback-controlled state-space model

$$\mathbf{x}_{t+1} \;=\; (\mathbf{A} - \mathbf{B}\mathbf{L}_t(T))\mathbf{x}_t + \mathbf{w}_t, \tag{8}$$

which can now be used to predict the next kinematic state in the decoder. Note that $\mathbf{Q}_t$ and $\mathbf{R}$ should be appropriately designed for an application of interest (we will show examples of these for reaching movements). Also $\mathbf{L}_t(T)$ is time-varying and a function of the duration, T, and hence the state-space model is dependent on this duration and time-varying.

1) Optimal Feedback-Controlled Prior Model for a Reaching Movement: One of the most common goal-directed movements is a reaching movement. We can now specialize the feedback-controlled prior model to the case of reaching movements, which are used in most real-time BMI experiments. For a reaching movement, the goal is to reach a target position at a desired arrival time and stop there, while being energy efficient. Hence the cost function in this case should enforce positional accuracy at the arrival time, stopping condition, and energetic efficiency [37], [38]. Therefore, denoting the desired final target position in the two dimensions by $\mathbf{d}^* = [d_1^*, d_2^*]'$, and the position, velocity, and force in the two dimensions by $\mathbf{d}_t = [d_1(t), d_2(t)]$, $\mathbf{v}_t = [v_1(t), v_2(t)]$, and $\mathbf{a}_t = [a_1(t), a_2(t)]$, respectively, similar to previous studies [37], [38] we take this cost function to be the weighted sum

$$J = \|\mathbf{d}_T - \mathbf{d}^*\|^2 + w_v\,\|\mathbf{v}_T\|^2 + w_a\,\|\mathbf{a}_T\|^2 + w_r\sum_{t=1}^{T-1}\|\mathbf{u}_t\|^2 \tag{9}$$

where the weights are chosen to penalize the terms in the cost function approximately equally on average [37], [38]. Taking the state to be $\mathbf{x}_t = [d_1(t), v_1(t), a_1(t), d_2(t), v_2(t), a_2(t)]'$, we adapt a first-order lowpass muscle-like system [38] for the dynamical system in (2), which in each dimension i is given by

$$\begin{bmatrix} d_i(t+1) \\ v_i(t+1) \\ a_i(t+1) \end{bmatrix} = \begin{bmatrix} 1 & \Delta & 0 \\ 0 & 1-\frac{b\Delta}{m} & \frac{\Delta}{m} \\ 0 & 0 & 1-\frac{\Delta}{\tau} \end{bmatrix} \begin{bmatrix} d_i(t) \\ v_i(t) \\ a_i(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{\Delta}{\tau} \end{bmatrix} u_i(t) + \begin{bmatrix} 0 \\ 0 \\ w_i(t) \end{bmatrix} \tag{10}$$

where Δ is the time increment, and the parameters $b = 10$ Ns/m, $\tau = 0.05$ s, and $m = 1$ kg come from biomechanics [38].

Having specified the kinematic model in (10) and the cost function in (9) for the reaching movements, the feedback matrices $\mathbf{L}_t(T)$ can be computed from the recursive solution of LQG in (5). To do so, we augment the state to include the target position in the two dimensions [37], [38], $\mathbf{d}^* = [d_1^*, d_2^*]'$, i.e., $\mathbf{x}_{\text{aug}}(t) = [d_1(t), v_1(t), a_1(t), d_1^*, d_2(t), v_2(t), a_2(t), d_2^*]'$. Hence the augmented state-space model in each dimension takes the form

$$\begin{bmatrix} d_i(t+1) \\ v_i(t+1) \\ a_i(t+1) \\ d_i^* \end{bmatrix} = \begin{bmatrix} 1 & \Delta & 0 & 0 \\ 0 & 1-\frac{b\Delta}{m} & \frac{\Delta}{m} & 0 \\ 0 & 0 & 1-\frac{\Delta}{\tau} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} d_i(t) \\ v_i(t) \\ a_i(t) \\ d_i^* \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{\Delta}{\tau} \\ 0 \end{bmatrix} u_i(t) + \begin{bmatrix} 0 \\ 0 \\ w_i(t) \\ 0 \end{bmatrix} \tag{11}$$

for $i = 1, 2$. Now comparing (9) with (3) and denoting the two dimensional identity matrix with $\mathbf{I}$, we find $\mathbf{R} = w_r\mathbf{I}$, and $\mathbf{Q}_t = 0$ for $t < T$. To find $\mathbf{Q}_T$ note that $d_1(T) - d_1^* = p_1'\mathbf{x}_{\text{aug}}(T)$ with $p_1 = [1, 0, 0, -1, 0, 0, 0, 0]'$ and similarly for p_2, and that $v_1(T) = p_{v_1}'\mathbf{x}_{\text{aug}}(T)$ with $p_{v_1} = [0, 1, 0, 0, \cdots, 0]'$ and similarly for p_{v_2}, p_{a_1}, and p_{a_2}. Therefore we find $\mathbf{Q}_T = p_1p_1' + p_2p_2' + w_v(p_{v_1}p_{v_1}' + p_{v_2}p_{v_2}') + w_a(p_{a_1}p_{a_1}' + p_{a_2}p_{a_2}')$. Now using (5)–(7) with these choices we obtain the feedback matrices, $\mathbf{L}_t(T)$, and consequently the optimal feedback-controlled state-space model for the reaching movement from (8). These feedback matrices can be precomputed offline and then stored for later real-time use in a BMI. Also, the prior model can be generalized to any target location by just changing $\mathbf{d}^*$ accordingly in (9).

B. Estimation of Goal-Directed Kinematic States from Neural Observations

Having developed the prior model, we now derive the decoder. To do so, we combine the prior model with the ensemble spiking activity during movement and resolve the duration uncertainty inherent to the prior model using mixture modeling. Mixture modeling is a common framework in statistical inference that is used to estimate a desired density in different applications. For example, mixture modeling combined with sequential state estimation in dynamical systems, when the system could be operating in changing environments (leading to different regimes of operation), has been used in [39]. See also the mixture Kalman filtering work in [40] and references therein. For decoding the kinematics from neural activity, mixture modeling was used in [25] and successfully applied to combine empirically fitted and time-invariant state models for reaching movements to different targets. Here we use mixture modeling to combine optimal feedback-controlled prior models of different durations and hence resolve the duration uncertainty inherent to this prior model (and other goal-directed models in general).

To decode the kinematics, we need to find the posterior density $p(\mathbf{x}_t|\mathbf{N}_{1:t})$. Conditioning on the arrival time, we can expand this density as,

$$p(\mathbf{x}_t|\mathbf{N}_{1:t}) = \int p(\mathbf{x}_t|\mathbf{N}_{1:t},\, T)\, p(\,T|\mathbf{N}_{1:t})\, dT \tag{12}$$

where $p(\,T|\mathbf{N}_{1:t})$ is the causal likelihood function for the arrival time based on the neural observations. However, in its continuous form, this likelihood is difficult to find and in general is not Gaussian. Hence to make the computations tractable, we discretize the arrival time and instead, using the law of total probability, expand the posterior density as

$$p(\mathbf{x}_t|\mathbf{N}_{1:t}) = \sum_{j=1}^{J} p(\mathbf{x}_t|\mathbf{N}_{1:t},\, T_j)\, p(\,T_j|\mathbf{N}_{1:t}) \tag{13}$$

where T is discretized to J possibilities and consequently a prior model is placed on it given by $p_T(T_j), j = 1, \cdots, J$. This prior model (including its support) can be selected based on empirical durations observed in a given task. We will later examine how the discretization refinement, or equivalently the number of discretization points used for a given support, affects the performance of the decoder. We show in the Results section that even a coarse discretization is sufficient to achieve an asymptote performance.

We now need to calculate two quantities: the posterior density $p(\mathbf{x}_t|\mathbf{N}_{1:t},\, T)$ for a given *known* arrival time, T, and the corresponding weights $p(\,T|\mathbf{N}_{1:t})$.

To find the posterior for a given arrival time we write it as,

$$p(\mathbf{x}_t|\mathbf{N}_{1:t},\, T) = \frac{p(\mathbf{N}_t|\mathbf{x}_t, \mathbf{N}_{1:t-1})\, p(\mathbf{x}_t|\mathbf{N}_{1:t-1},\, T)}{p(\mathbf{N}_t|\mathbf{N}_{1:t-1},\, T)} \tag{14}$$

The first term in the numerator comes from the observation model in (1) and the second term is the one-step prediction density. Note that we used the fact $p(\mathbf{N}_t|\mathbf{x}_t, \mathbf{N}_{1:t-1},\, T) = p(\mathbf{N}_t|\mathbf{x}_t, \mathbf{N}_{1:t-1})$ based on the observation model in (1). We denote the normalization constant by

$$g(\mathbf{N}_t|\,T) = p(\mathbf{N}_t|\mathbf{N}_{1:t-1},\, T) \tag{15}$$

since we exploit it later to find the corresponding weights. To get the recursion for the posterior density, we use the Chapman-Kolmogorov equation to write the prediction density as

$$p(\mathbf{x}_t|\mathbf{N}_{1:t-1},\, T) = \int p(\mathbf{x}_t|\mathbf{x}_{t-1},\, T) p(\mathbf{x}_{t-1}|\mathbf{N}_{1:t-1},\, T) d\mathbf{x}_{t-1}. \tag{16}$$

Here we have used the conditional independence, $p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{N}_{1:t-1},\, T) = p(\mathbf{x}_t|\mathbf{x}_{t-1},\, T)$, which comes from the state-space model in (8). Now the second term inside the integral is just the posterior density from the previous time step. Hence substituting (16) into (14) generates the recursion. The exact expression in

(14) is in general complicated. Hence we make a Gaussian approximation to the posterior density similar to [35]. Making this approximation and since the state-space model in (8) is also Gaussian, the prediction density in (16) will be Gaussian. Let's denote the minimum MSE (MMSE) estimator, i.e., $E(\mathbf{x}_t|\mathbf{N}_{1:t}, \mathcal{T})$, by $\mathbf{x}_{t|t,\mathcal{T}}$ and its covariance matrix by $\mathbf{W}_{t|t,\mathcal{T}}$. Similarly, we denote the one step prediction mean by $\mathbf{x}_{t|t-1,\mathcal{T}} = E(\mathbf{x}_t|\mathbf{N}_{1:t-1}, \mathcal{T})$ and its covariance matrix by $\mathbf{W}_{t|t-1,\mathcal{T}}$. The Gaussian approximation to the posterior in (14) is made by expanding its logarithm in a Taylor series expansion around the prediction mean $\mathbf{x}_{t|t-1,\mathcal{T}}$ and keeping terms of second order or smaller. The recursions for the MMSE estimator with this Gaussian approximation have been derived in [35]. The difference here is in the prediction step as it uses the controlled state-space model in (8). The recursions of this point process filter in our case become

$$\mathbf{x}_{t|t-1,\mathcal{T}} = (\mathbf{A} - \mathbf{B}\mathbf{L}_t(\mathcal{T}))\mathbf{x}_{t-1|t-1,\mathcal{T}} \tag{17}$$

$$\mathbf{W}_{t|t-1,\mathcal{T}} = (\mathbf{A} - \mathbf{B}\mathbf{L}_t(\mathcal{T}))\mathbf{W}_{t-1|t-1,\mathcal{T}}(\mathbf{A} - \mathbf{B}\mathbf{L}_t(\mathcal{T}))' + \mathbf{W} \tag{18}$$

$$\mathbf{W}^{-1}_{t|t,\mathcal{T}} = \mathbf{W}^{-1}_{t|t-1,\mathcal{T}} + \sum_{c=1}^{C}\left[\left(\frac{\partial \log \lambda_c}{\partial \mathbf{x}_t}\right)'\left(\frac{\partial \log \lambda_c}{\partial \mathbf{x}_t}\right)\lambda_c\Delta - (N^c_t - \lambda_c\Delta)\frac{\partial^2 \log \lambda_c}{\partial \mathbf{x}_t \partial \mathbf{x}_t'}\right]_{\mathbf{x}_{t|t-1,\mathcal{T}}} \tag{19}$$

$$\mathbf{x}_{t|t,\mathcal{T}} = \mathbf{x}_{t|t-1,\mathcal{T}} + \mathbf{W}_{t|t,\mathcal{T}}\sum_{c=1}^{C}\left[\left(\frac{\partial \log \lambda_c}{\partial \mathbf{x}_t}\right)'(N^c_t - \lambda_c\Delta)\right]_{\mathbf{x}_{t|t-1,\mathcal{T}}} \tag{20}$$

where $[\cdot]_{\mathbf{x}_{t|t-1,\mathcal{T}}}$ denotes the evaluation of the expression at $\mathbf{x}_{t|t-1,\mathcal{T}}$ and λ_c is used instead of $\lambda_c(t|\mathbf{x}_t, \mathbf{H}^c_t)$ for notational convenience. When the point process model for the spiking activity is assumed to have no history dependence, the recursions are obtained by simply replacing $\lambda_c(t|\mathbf{x}_t, \mathbf{H}^c_t)$ with $\lambda_c(t|\mathbf{x}_t)$. These recursions give the feedback-controlled point process filter (FC-PPF).

Note that a random-walk state model is also in the form of (8) but with $\mathbf{B} = \mathbf{0}$ and equivalently the dynamics matrix given by $\mathbf{A}$. Hence the recursions in (17)–(20) with this choice recover the random-walk point process filter (RW-PPF).

So far we have found an approximation to the posterior assuming a *known* arrival time. To relax this assumption, we should find the corresponding weights for each arrival time, $p(\mathcal{T}_j|\mathbf{N}_{1:t})$, and use (13) to find the posterior density. We have

$$p(\mathcal{T}_j|\mathbf{N}_{1:t}) = \frac{p(\mathbf{N}_{1:t}|\mathcal{T}_j)p_{\mathcal{T}}(\mathcal{T}_j)}{p(\mathbf{N}_{1:t})} \tag{21}$$

where $p(\mathbf{N}_{1:t})$ is independent of $\mathcal{T}_j$ and treated as a constant and $p(\mathbf{N}_{1:t}|\mathcal{T}_j)$ represents the likelihood of the observed neural data under a state-space model with the arrival time of $\mathcal{T}_j$. Hence the latter is the normalization constant for the posterior $p(\mathbf{x}_{1:t}|\mathbf{N}_{1:t}, \mathcal{T}_j)$ and its exact computation requires an integration,

which is computationally prohibitive. However, using the Gaussian approximation to the posterior, we can find this without integration as follows. Using the chain rule we find

$$p(\mathbf{N}_{1:t}|\,\mathcal{T}_j) = \prod_{i=1}^{t} p(\mathbf{N}_i|\mathbf{N}_{1:i-1},\,\mathcal{T}_j) = \prod_{i=1}^{t} g(\mathbf{N}_i|\,\mathcal{T}_j) \tag{22}$$

where $g(\mathbf{N}_i|\,\mathcal{T}_j)$ is defined in (15) and is the ith step normalization constant in the recursive filter. Now exploiting the Gaussian approximation of the posterior and hence the prediction densities in (14), they are completely characterized by their means and covariances given in (17)–(20) for any $\mathcal{T}_j$. We can hence explicitly evaluate (14) at $\mathbf{x}_{i|i,\mathcal{T}_j}$ to get

$$\begin{aligned} g(\mathbf{N}_i|\,\mathcal{T}_j) &= \sqrt{\frac{|\mathbf{W}_{i|i,\mathcal{T}_j}|}{|\mathbf{W}_{i|i-1,\mathcal{T}_j}|}}\, p(\mathbf{N}_i|\mathbf{x}_{i|i,\mathcal{T}_j}, \mathbf{N}_{1:i-1})\times \\ &\exp\left[-\frac{1}{2}(\mathbf{x}_{i|i,\mathcal{T}_j} - \mathbf{x}_{i|i-1,\mathcal{T}_j})'\mathbf{W}^{-1}_{i|i-1,\mathcal{T}_j}(\mathbf{x}_{i|i,\mathcal{T}_j} - \mathbf{x}_{i|i-1,\mathcal{T}_j})\right] \\ &= \sqrt{\frac{|\mathbf{W}_{i|i,\mathcal{T}_j}|}{|\mathbf{W}_{i|i-1,\mathcal{T}_j}|}} \prod_c \left(\lambda_c(i|\mathbf{x}_{i|i,\mathcal{T}_j}, \mathbf{H}_i^c)\Delta\right)^{N_i^c} e^{-\lambda_c(i|\mathbf{x}_{i|i,\mathcal{T}_j},\mathbf{H}_i^c)\Delta} \\ &\times \exp\left[-\frac{1}{2}(\mathbf{x}_{i|i,\mathcal{T}_j} - \mathbf{x}_{i|i-1,\mathcal{T}_j})'\mathbf{W}^{-1}_{i|i-1,\mathcal{T}_j}(\mathbf{x}_{i|i,\mathcal{T}_j} - \mathbf{x}_{i|i-1,\mathcal{T}_j})\right] \end{aligned} \tag{23}$$

for $j = 1, \cdots, J$ where all the quantities are known. Combining (13)–(23) gives the posterior. The MMSE estimate in the case of unknown arrival time is then given by

$$\mathbf{x}_{t|t} = E(\mathbf{x}_t|\mathbf{N}_{1:t}) = \sum_j p(\,\mathcal{T}_j|\mathbf{N}_{1:t})\mathbf{x}_{t|t,\mathcal{T}_j}.$$

We call the resulting filter the feedback-controlled parallel point process filter (FC-P-PPF) shown in Fig. 2; it consists of J parallel branches of feedback-controlled point process filters, each calculating not only the MMSE estimate of $\mathbf{x}_t$ assuming a duration of $\mathcal{T}_j$, but also the corresponding likelihood $p(\mathbf{N}_{1:t}|\,\mathcal{T}_j)$.

One question that arises at this point is how to estimate the state once $t > \mathcal{T}_j$ for some branch $\mathcal{T}_j$. One way to do so is to perform the summation over all j for which $\mathcal{T}_j > t$. This means that after a given arrival time, $\mathcal{T}_j$, the corresponding branch of the decoder exits the computations and is no longer used for state estimation. This approach is justified since the prior for a given branch is only defined up to its corresponding arrival time. Using this implementation and as time passes by, fewer branches are active and hence the computational load is lower. Another way to address this issue is to keep a branch in the decoder even after its arrival time is reached by changing its prior to one that corresponds to a still condition. This still condition will have its position fixed at the final position reached to at the arrival time and hence a zero velocity (and other higher order derivatives of the position) after the arrival time.

This means that all the branches will be active at all time until the decoder is terminated at the largest arrival time of the branches. We will examine the effect of the two alternative implementations in detail in the Results section.

Finally, the parallel point process filter can be used to remove the duration uncertainty from any other goal-directed state-space model and not just the feedback-controlled one we have proposed here. This can be done simply by changing the prediction step in (17) and (18) using the desired state-space model. This is expanded on in the Appendix. Also, our approach to resolving the duration uncertainty can be applied in cases where the observations are taken to be the neural firing rates instead of the spiking activity and a Gaussian likelihood model is assumed on these firing rates as in [24]. In this case the Gaussian approximation to the posterior in our FC-P-PPF becomes exact and the update recursions in the filter become those of the well-known Kalman filter. The calculation of the arrival time likelihoods, however, remain the same.

III. Results

We show the application of FC-P-PPF to decoding of reaching movements performed by a rhesus monkey from ensemble spiking activity simulated based on a rigorously validated model of neural activity in the primary motor cortex [2], [33]. Our data consists of 55 reaching trajectories performed by a rhesus monkey[1], based on which neural spiking activity is simulated. Using a joystick, the monkey moved a cursor from the center to one of four possible radial locations at the top, bottom, left, or right sides of a square computer screen 14 cm long in each dimension. The 55 trajectories had different durations in the range of 140–400 ms. Hence the window of uncertainty for the arrival time is 140–400 ms. The only parameter of the state-space model in (10) fitted to reaching trajectories using maximum-likelihood parameter estimation was the state noise covariance, $\mathbf{W}$.

To dissociate the effect of using the feedback-controlled prior model from that of using the parallel filters to resolve the duration uncertainty, we first consider the scenario where the decoder knows the exact duration for each trajectory and then remove this assumption and consider the general case with unknown duration. Also, in each case, we compare the decoder with RW-PPF in which no target information is incorporated and show that the decoder improves the RW-PPF performance considerably, where performance is measured as the average root-mean-square (RMS) error in the estimated trajectories.

[1]This study was performed in strict accordance with the recommendations in the Guide for the Care and Use of Laboratory Animals of the National Institutes of Health, and under regulation of the Subcommittee on Research Animal Care at Harvard Medical School.

In terms of the decoder implementation, we examine the effect of two factors on its performance. First, we examine the effect of the discretization refinement, i.e., the number of discretization points or parallel branches used in the decoder, and show that a relatively coarse discretization is sufficient for the decoder to achieve an asymptote average RMS error. This guides the selection of the number of parallel branches in the decoder. Second we investigate the effect of the two possible implementations for treating a branch after its arrival time. In the first implementation the branches exit the decoder after their arrival time and in the second implementation they stay in the decoder in their still condition after their arrival time. We show that the average RMS errors using the two implementations are comparable.

A. Estimation of Reaching Movements of Known Duration from Neural Signals

To assess the advantage of using the feedback-controlled prior model, we first implement the FC-P-PPF for estimation of reaching movements that start at a known reference position at rest, acquire a target position at a *known* time, and come to a stop at that time. Note that when duration is known, FC-P-PPF uses a single branch corresponding to the true duration for each trajectory.

To get the corresponding ensemble spiking activity for each real reaching trajectory, we used the model of neural activity in the motor cortex, also known as the cosine tuning model of the conditional intensity function, which has been extensively validated on primate motor cortical data [2], [33]. In this model, the firing rate of each neuron is related to the movement speed and direction. Using this model and for each trajectory, we simulated the spiking activity of an ensemble of $C = 20$ neurons, which is a typical ensemble size in real-time BMIs. The spiking activity for each neuron in the ensemble was independently simulated as a point process whose instantaneous firing rate or conditional intensity function during a two-dimensional movement was given by [33],

$$\begin{aligned} \lambda_c(t|\mathbf{x}_t) &= \exp\left(\alpha_0 + \alpha_1 |\mathbf{v}_t| \cos(\theta_t - \theta_p^c)\right) \\ &= \exp\left(\alpha_0 + \alpha_y^c v_2(t) + \alpha_x^c v_1(t)\right) \end{aligned} \tag{24}$$

where θ_p^c is the preferred angle of the cth neuron, θ_t is the movement angle at t, $v_2(t)$ and $v_1(t)$ are the velocities in the y and x directions, α_0 and α_1 are parameters of the model, and consequently, $\alpha_x^c = \alpha_1 \cos(\theta_p^c)$ and $\alpha_y^c = \alpha_1 \sin(\theta_p^c)$. Here the preferred direction for each neuron is sampled randomly from $[-\pi, \pi]$, $\alpha_0 = 1.6$ is chosen to have a background firing rate of 5 Hz for each neuron and $\alpha_1 = 0.04$ s/cm is chosen to have a maximum possible firing rate of 25 Hz on average (over the trajectories) for each neuron. We simulated 100 realizations (trials) of the point process neural signal for each trajectory using the time-rescaling theorem described in [41].

1) Performance Measure: We define the performance measure of the decoder as its average RMS error over the trajectories. To find this, we first find the RMS error for each time step along a trajectory by averaging the square estimation error at that time step over the 100 simulated neural realizations. To quantify an average RMS error for the entire trajectory, we then average the RMS error over its duration. Finally, we find the mean of the average RMS errors over all 55 trajectories as the measure of performance for a decoder.

2) Comparison to a Random-Walk Model: Assuming a known movement duration, we compare the average RMS error of FC-P-PPF to that of RW-PPF. This comparison illustrates the advantage of using a feedback-controlled prior model that incorporates the target information over a random-walk one. The average RMS errors of RW-PPF and FC-P-PPF, which are calculated for each trajectory until the end of movement, are 1.40 cm and 0.87 cm, respectively. Hence the RW-PPF average RMS is 61% higher than that of FC-P-PPF. This shows the benefit of using a more accurate prior model in the decoder. Fig. 3a and 3b show a sample reaching trajectory performed by the monkey with a duration of $T = 296$ ms and 10 sample decoded trajectories from RW-PPF and FC-P-PPF when T is known exactly to the decoder. Fig. 3c and 3d show the estimated velocity in the x and y direction for a sample neural realization (trial) using RW-PPF and FC-P-PPF. We can also visually observe that using the feedback-controlled prior model improves the estimation accuracy. Note that the average RMS error of FC-P-PPF assuming knowledge of movement duration provides a lower bound on its error in the general realistic case where this duration is not known, as considered in the next section.

B. Estimation of Reaching Movements of Unknown Duration from Neural Signals

We now remove the assumption of known movement duration from the problem and use FC-P-PPF to decode the same reaching movements. We examine the effects of the discretization refinement and two possible implementations on FC-P-PPF performance and also compare it to RW-PPF. Since the movement duration is unknown, we run both decoders until the end of the uncertainty window, i.e., until 400 ms.

1) Performance Loss Due to Unknown Duration: We first examine the loss incurred due to lack of a priori knowledge of the movement duration by FC-P-PPF and the effect of the number of discretization points on this loss. This will guide the selection of the number of branches needed in the decoder. To do so, we compare the average RMS error of FC-P-PPF with various number of discretization points to its error assuming exact a priori knowledge of duration as considered in the previous section. The latter provides a lower bound on the former and is hence used as a baseline.

We discretize the uncertainty interval starting at 150 ms, i.e., the interval $[150, 400]$ ms, in steps of 250

ms, 125 ms, 83 ms, 50 ms, or 25 ms, corresponding to 2, 3, 4, 6, or 11 discretization points, respectively, and find the FC-P-PPF average RMS error in each case. We also find the average RMS error using a single branch at 400 ms, which corresponds to a single discretization point at the largest possible duration. In each case, we assume a uniform prior distribution on the discretization points. To make the comparison to the baseline case with known duration, we compute the average RMS error of FC-P-PPF with different number of discretization points for each trajectory up to the end of movement as opposed to the end of the uncertainty window. In each case, we also examine the two possible implementations for treating a branch after its arrival time.

Fig. 4 shows the FC-P-PPF average RMS error as a function of the number of discretization points for both implementations and also the baseline average RMS error assuming movement duration is known. In the first implementation, the average RMS error has a steep drop to an asymptote value by increasing the number of discretization points. Taking the average RMS error achieved using 11 discretization points as the asymptote value, using only 4 discretization points results in the difference between the average RMS error and the asymptote value to be only 1% of the latter. This shows that a relatively coarse discretization is sufficient to get to this asymptote. Also, increasing the number of discretization points from 1 to 4 reduces the difference between the average RMS error of FC-P-PPF and the baseline (i.e., with a priori knowledge of the duration) by over 48%.

Similar results hold for the second implementation. The two implementations have very close average RMS error during movement. For example using 4 discretization points, the difference between the average RMS error of the two implementations is less than 1% of the smaller value.

2) Comparison to a Random-Walk Point Process Filter: We now compare the performance of FC-P-PPF to RW-PPF that does not exploit the target information. In our comparison, we use 4 branches for FC-P-PPF as this choice is sufficient to get within 1% of the asymptote average RMS error.

Fig. 5a and 5b show a sample trajectory with a duration of $T = 296$ ms (the same as in Fig. 3) and 10 sample decoded trajectories using RW-PPF and FC-P-PPF when T is unknown to the decoders and hence they decode the neural activity until the end of the uncertainty window. We have shown the FC-P-PPF results using the first implementation. As the figure demonstrates, FC-P-PPF generates more accurate trajectory estimates compared to RW-PPF (as will be quantified in Table I). Fig. 5c–5f show the decoded velocities in the x and y directions for a single realization of the neural signal and the time evolution of the optimal combining weights in FC-P-PPF for the given realization and also averaged over all 100 realizations for the given trajectory. Here the true arrival time is at $T = 296$ ms and the closest weight, i.e., the one corresponding to 316 ms, soon dominates up to its arrival time. Also, FC-P-PPF

brings the estimated velocity close to zero at the end of movement as opposed to RW-PPF.

To quantify these observations, Table I shows the average RMS error in the estimate of the 55 trajectories during movement and during the entire decoding period. These are found by averaging the RMS error for each trajectory until the end of its corresponding movement and until the end of the uncertainty window at 400 ms, respectively. During movement, the average RMS error of RW-PPF is 47% larger than that of FC-P-PPF. Moreover, the two implementations of FC-P-PPF have very similar average RMS errors, as is also observed in Fig. 4. When also considering the error after the end of movement, RW-PPF average RMS error is 67% higher than FC-P-PPF. In this case, FC-P-PPF does better than RW-PPF for two reasons: First, it puts a better prior model on the reaching movements. Second, it detects the end of movement and hence brings the velocity close to zero after the movement ends. To do so, FC-P-PPF finds the likelihoods of the discretized arrival times and reflects them into its estimate. This is done by combining the estimates of the parallel filters, each designed for a different arrival time, with time-varying optimal weights that are updated purely based on the neural observation. In contrast, RW-PPF cannot detect the movement termination. Therefore, when considering its error until the end of the uncertainty window, it does worse. Note that the two implementations of FC-P-PPF again have a close average RMS error.

Fig. 6a–6d show sample estimated trajectories, velocities, and average combining weights for the second implementation of FC-P-PPF. Keeping all the branches in the filter even after their arrival time, by changing their prior model to a still model, allows the second implementation to bring the velocity closer to zero after the movement ends and results in a lower average RMS error after the end of movement compared to the first implementation (1.08 cm vs. 1.16 cm, calculated from the end of movements until the end of uncertainty window at 400 ms). Therefore, in applications where the error after the end of movement and hence standing still at a position is important, for example when reaching a target and then holding it, this implementation could result in additional reduction of error. However, the first implementation is computationally more efficient as it does not keep all the branches in its calculations until the end of decoding. Hence the first implementation is more appropriate in real-time applications in which the main purpose is only to reach a target.

DISCUSSION

We have developed a recursive Bayesian decoder for goal-directed movements from ensemble neural spiking activity. Our decoder employs two main components. First, it uses the sensorimotor optimal feedback control principles to build an optimal feedback-controlled prior model for goal-directed movements.

Second, it resolves the duration uncertainty inherent to this model (and other goal-directed prior models) based on the neural activity. The result is the feedback-controlled parallel point process filter (FC-P-PPF) that can be used in a real-time BMI for estimation of goal-directed movements.

We used an optimal feedback control model of the sensorimotor system to build the prior model. Our prior model incorporates the target information and takes into account the sensory feedback. It can also generalize to different target locations without requiring a training data set of reaches to these locations. Even though we have illustrated one example of the optimal feedback-controlled prior model in (8) for the specific task of reaching a target, this model could be specialized to other tasks by quantifying their goals as the cost function in (3), thus providing a possible framework for construction of goal-directed prior models.

Despite the advantage of goal-directed prior models for estimation of movement, one key factor that has prevented their implementation in real-time BMIs is their dependence on the movement duration, which is not known a priori to a real-time decoder. Previous work using duration-dependent goal-directed models have therefore assumed that the decoder has knowledge of this duration [26], [28], which is not the case in a real-time BMI. We thus developed a parallel point process filter framework to resolve this duration uncertainty in any goal-directed prior model. The framework works by discretizing the duration, causally estimating the kinematics for each discretized duration based on the neural activity, and optimally combining these estimates according to the duration likelihoods, which are again calculated from the neural activity. Due to the parallel nature of our decoder, its overall run-time is on the order of the run-time for a single branch, a property important for real-time implementation.

We additionally examined the effect of the discretization refinement of the duration on the average RMS error and showed that using only a coarse discretization is sufficient to achieve an asymptote average RMS error. This guides the selection of the number of branches in the decoder and shows that the added computational complexity involved to address the duration uncertainty is not large.

We further examined the effect of two possible filter implementations for FC-P-PPF, which differed in treating the branches of the filter after their arrival time. In the first implementation, any branch exits the computation after its arrival time. In the second implementation, we keep all the branches in the filter after their arrival time, but change their prior model to a still model. The still model for a branch assumes that the kinematic state has stopped at the location estimated at its arrival time. We showed that the two implementations have similar average RMS errors. However, the second implementation brings the velocity closer to zero after the movement ends (and therefore may have an advantage for applications such as holding a target), whereas the first implementation is computationally more efficient.

We also examined the advantages of using our decoder compared to RW-PPF that uses a random-walk prior model. We showed that using a very coarse discretization (and hence few branches in FC-P-PPF) the average RMS error of our decoder is significantly lower than that of RW-PPF. The reason for this improved performance is the fact that FC-P-PPF not only uses a better prior model on the kinematic states, but also attempts to identify the end of movement and hence bring the velocity close to zero at that point. This is in contrast to RW-PPF, which cannot detect the end of movement.

Motivated by the observation that target location can be accurately decoded from motor cortical areas before movement initiation [22], [23], in this work we assumed that target location is known and developed a decoder that combines it with the peri-movement activity to estimate the intended kinematics. Note, however, that we can easily extend this decoder for cases where target location is not known prior to movement by using an approach similar to the work in [25], which combines time-invariant and empirically fitted trajectory models to different targets. To do so, we can include branches in FC-P-PPF for all possible target locations and their corresponding discretized durations. We can then calculate the overall kinematic estimate as the weighted average of the branch estimates corresponding to all possible target locations, as opposed to just the correct target location. The calculation of the branch likelihoods, that in turn determine their weights in calculating the overall kinematic estimate, remains exactly the same (see (21)–(23)). In this case, if target related activity before movement initiation is recorded, we can incorporate it by initializing the weight of the branches corresponding to each target proportional to its likelihood calculated from this activity [25], as opposed to equiprobably.

Real-time BMIs so far have decoded individually either the target of the movement or its corresponding trajectory. Here we developed a decoder that can enable joint decoding of both target and trajectory in a real-time BMI. This can in turn result in more accurate estimation of intended movement than is possible by decoding only target or kinematic related information alone.

APPENDIX
RESOLVING THE DURATION UNCERTAINTY FROM A GENERAL LINEAR GOAL-DIRECTED STATE-SPACE MODEL

A general linear goal-directed state-space model depends on the target and movement duration and can be written as

$$\mathbf{x}_{t+1} = \mathbf{G}(t, T, \mathbf{x}^*)\mathbf{x}_t + \mathbf{b}(t, T, \mathbf{x}^*) + \mathbf{w}_t, \tag{25}$$

where $\mathbf{x}^*$ is the target state, $\mathbf{G}$ and $\mathbf{b}$ are the dynamics matrix and a constant term, respectively, that in general are time-varying and also dependent on $\mathbf{x}^*$ and T, and $\mathbf{w}_t$ is the zero-mean white Gaussian state

noise with covariance matrix $\mathbf{W}_t$. Note that the state-space models in both [26] and [28] can be written in this form and assume prior knowledge of T. Hence the exact same filter recursions in (17)–(20) can be applied to a general model by just changing the filter prediction steps to

$$\mathbf{x}_{t|t-1,T} = \mathbf{G}(t, T, \mathbf{x}^*)\mathbf{x}_{t-1|t-1,T} + \mathbf{b}(t, T, \mathbf{x}^*)$$

$$\mathbf{W}_{t|t-1,T} = \mathbf{G}(t, T, \mathbf{x}^*)\mathbf{W}_{t-1|t-1,T}\mathbf{G}'(t, T, \mathbf{x}^*) + \mathbf{W}_t$$

All the other expressions including those for calculating the duration likelihoods remain the same.

REFERENCES

[1] A. Georgopoulos, J. F. Kalaska, R. Caminiti, and J. T. Massey, "On the relations between the direction of two-dimensional arm movement and cell discharge in primate motor cortex," *J. Neurosci.*, vol. 2, no. 11, pp. 1527–1537, Nov. 1982.

[2] D. W. Moran and A. B. Schwartz, "Motor cortical representation of speed and direction during reaching," *J. Neurophysiol.*, vol. 82, pp. 2676–2692, 1999.

[3] R. Caminiti, P. B. Johnson, C. Galli, S. Ferraina, and Y. Burnod, "Making arm movements within different parts of space: The premotor and motor cortical representation of a coordinate system for reaching to visual targets," *J. Neurosci.*, vol. 11, no. 5, pp. 1182–1197, May 1991.

[4] G. H. Mulliken, S. Musallam, and R. A. Andersen, "Forward estimation of movement state in posterior parietal cortex," *Proc. Natl. Acad. Sci. USA*, vol. 105, pp. 8170–8177, 2008.

[5] J. Wessberg, C. R. Stambaugh, J. D. Kralik, P. D. Beck, M. Laubach, J. K. Chapin, J. Kim, S. J. Biggs, M. A. Srinivsan, and M. Nicolelis, "Real-time prediction of hand trajectory by ensembles of cortical neurons in primates," *Nature*, vol. 408, pp. 361–365, Nov. 2000.

[6] L. H. Snyder, A. P. Batista, and R. A. Andersen, "Coding of intention in the posterior parietal cortex," *Nature*, vol. 386, pp. 167–170, Mar. 1997.

[7] A. P. Batista, C. A. Buneo, L. H. Snyder, and R. A. Andersen, "Reach plans in eye-centered coordinates," *Science*, vol. 285, pp. 257–260, Jul. 1999.

[8] J. Messier and J. F. Kalaska, "Covariation of primate dorsal premotor cell activity with direction and amplitude during a memorized-delay reaching task," *J. Neurophysiol.*, vol. 84, pp. 152–165, 2000.

[9] J. K. Chapin, K. A. Moxon, R. S. Markowitz, and M. A. L. Nicolelis, "Real-time control of a robot arm using simultaneously recorded neurons in the motor cortex," *Nat. Neurosci.*, vol. 2, no. 7, pp. 664–670, Jul. 1999.

[10] M. D. Serruya, N. G. Hatsopoulos, L. Paninski, M. R. Fellows, and J. P. Donoghue, "Instant neural control of a movement signal," *Nature*, vol. 416, pp. 141–142, Mar. 2002.

[11] D. M. Taylor, S. I. H. Tillery, and A. B. Schwartz, "Direct cortical control of 3D neuroprosthetic devices," *Science*, vol. 296, pp. 1829–1832, Jun. 2002.

[12] 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 Biol.*, vol. 1, no. 2, pp. 193–208, 2003.

[13] J. R. Wolpaw and D. J. McFarland, "Control of a two-dimensional movement signal by a noninvasive brain-computer interface in humans," *Proc. Natl. Acad. Sci. USA*, vol. 101, no. 51, pp. 17 849–17 854, Dec. 2004.

[14] L. R. Hochberg, M. D. Serruya, G. M. Friehs, J. A. Mukand, M. Saleh, A. H. Caplan, A. Branner, D. Chen, R. D. Penn, and J. P. Donoghue, "Neuronal ensemble control of prosthetic devices by a human with tetraplegia," *Nature*, vol. 442, pp. 164–171, Jul. 2006.

[15] 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, pp. 1098–1101, Jun. 2008.

[16] C. T. Moritz, S. I. Perlmutter, and E. E. Fetz, "Direct control of paralysed muscles by cortical neurons," *Nature*, vol. 456, pp. 639–643, Dec. 2008.

[17] G. H. Mulliken, S. Musallam, and R. A. Andersen, "Decoding trajectories from posterior parietal cortex ensembles," *J. Neurosci.*, vol. 28, no. 48, pp. 12 913–12 926, Nov. 2008.

[18] 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," *J. Neural Eng.*, vol. 5, pp. 455–476, 2008.

[19] K. Ganguly and J. M. Carmena, "Emergence of a stable cortical map for neuroprosthetic control," *PLoS Biol.*, vol. 7, no. 7, pp. 1–13, Jul. 2009.

[20] Z. Li, J. E. O'Doherty, T. L. Hanson, M. A. Lebedev, C. S. Henriquez, and M. A. L. Nicolelis, "Unscented Kalman filter for brain-machine interfaces," *PLoS ONE*, vol. 4, no. 7, pp. 1–18, Jul. 2009.

[21] 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, pp. 1203–1213, 2009.

[22] S. Musallam, B. D. Corneil, B. Greger, H. Scherberger, and R. A. Andersen, "Cognitive control signals for neural prosthetics," *Science*, vol. 305, pp. 258–262, Jul. 2004.

[23] 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.

[24] W. Wu, Y. Gao, E. Bienenstock, J. P. Donoghue, and M. J. Black, "Bayesian population decoding of motor cortical activity using a Kalman filter," *Neural Comput.*, vol. 19, pp. 80–118, 2006.

[25] B. M. Yu, C. Kemere, G. Santhanam, A. Afshar, S. I. Ryu, T. H. Meng, N. Sahani, and K. V. Shenoy, "Mixture of trajectory models for neural decoding of goal-directed movements," *J. Neurophysiol.*, vol. 97, pp. 3763–3780, 2007.

[26] L. Srinivasan, U. T. Eden, A. S. Willsky, and E. N. Brown, "A state-space analysis for reconstruction of goal-directed movements using neural signals," *Neural Comput.*, vol. 18, pp. 2465–2494, 2006.

[27] J. E. Kulkarni and L. Paninski, "State-space decoding of goal-directed movements," *IEEE Signal Process. Mag.*, vol. 25, pp. 78–86, Jan. 2008.

[28] C. Kemere and T. Meng, "Optimal estimation of feed-forward-controlled linear systems," in *Proc. ICASSP*, Mar. 2005, pp. 353–356.

[29] E. Todorov and M. I. Jordan, "Optimal feedback control as a theory of motor coordination," *Nat. Neurosci.*, vol. 5, no. 11, pp. 1226–1235, Nov. 2002.

[30] E. Todorov, "Optimality principles in sensorimotor control," *Nat. Neurosci.*, pp. 907–915, Sep. 2004.

[31] M. M. Shanechi, G. W. Wornell, Z. M. Williams, and E. N. Brown, "A parallel point-process filter for estimation of goal-directed movements from neural signals." in *Proc. ICASSP*, Mar. 2010.

[32] L. Srinivasan and M. da Silva, "Breaking the fixed-arrival-time restriction in reaching movements of neural prosthetic devices," *IEEE Trans. Biomed. Eng.*, vol. 58, no. 6, pp. 1555–1564, Jun. 2011.

[33] W. Truccolo, U. T. Eden, M. R. Fellows, J. P. Donoghue, and E. N. Brown, "A point process framework for relating

neural spiking activity to spiking history, neural ensemble, and extrinsic covariate effects," *J. Neurophysiol.*, vol. 93, pp. 1074–1089, 2005.

[34] M. M. Shanechi, Z. M. Williams, G. W. Wornell, R. 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," under review.

[35] U. T. Eden, L. M. Frank, R. Barbieri, V. Solo, and E. N. Brown, "Dynamic analysis of neural encoding by point process adaptive filtering," *Neural Comput.*, vol. 16, pp. 971–998, 2004.

[36] D. Bertsekas, *Dynamic Programming and Optimal Control*. Athena Scientific, 2005.

[37] E. Todorov, "Stochastic optimal control and estimation methods adapted to the noise characteristics of the sensorimotor system," *Neural Comput.*, vol. 17, pp. 1084–1108, 2005.

[38] D. Liu and E. Todorov, "Evidence for the flexible sensorimotor strategies predicted by optimal feedback control," *J. Neurosci.*, pp. 9354–9368, Aug. 2007.

[39] G. A. Ackerson and K. S. Fu, "On state estimation in switching environments," *IEEE Trans. Autom. Control*, vol. AC-15, no. 1, pp. 10–17, Feb. 1970.

[40] R. Chen and J. S. Liu, "Mixture Kalman filters," *J. Roy. Statist. Soc. B*, vol. 62, pp. 493–508, 2000.

[41] E. N. Brown, R. Barbieri, V. Ventura, R. Kass, and L. Frank, "The time-rescaling theorem and its application to neural spike train data analysis," *Neural Comput.*, vol. 14, pp. 325–346, 2001.

TABLES

TABLE I

RMS ERROR (CM) IN DECODED TRAJECTORY OF THE 55 REAL REACHING MOVEMENTS WITH UNKNOWN DURATION CALCULATED UNTIL THE END OF MOVEMENT AND UNTIL THE END OF THE UNCERTAINTY WINDOW.

	RW-PPF	FC-P-PPF (1)	FC-P-PPF (2)
Until the end of movement	1.40	0.95	0.94
Until the end of uncertainty window	1.69	1.01	0.99

FIGURE LEGENDS

Fig. 1. Optimal feedback control framework. An optimal feedback control framework is used to build a prior state-space model for goal-directed movements. In this framework each task is performed to accomplish a goal during which there is real-time sensory feedback, $\mathbf{y}_t$, about the state of the system, $\mathbf{x}_t$. Based on the intended goal and the sensory feedback about the state of the system, the brain (controller) decides on a plan of action or control signal, $\mathbf{u}_t$, and controls the system.

Fig. 2. Feedback-controlled parallel point process filter (FC-P-PPF). FC-P-PPF consists of J parallel branches of point process filters, each calculating not only the MMSE estimate of $\mathbf{x}_t$ assuming a duration of T_j, but also the corresponding likelihood $p(\mathbf{N}_{1:t}|\, T_j)$.

Fig. 3. Comparison of FC-P-PPF and RW-PPF when movement duration is known. (a) and (b) show a sample trajectory during a reaching movement by the monkey (solid black) and 10 sample decoded trajectories using RW-PPF in (a) and FC-P-PPF in (b) (dashed blue). (c) and (d) show the true velocity (black) in the two dimensions and the decoded velocities using RW-PPF (dashed red) and FC-P-PPF (solid blue) for a sample neural realization.

Fig. 4. The effect of the number of duration discretization points on FC-P-PPF average RMS error. The black curve shows the average RMS error of the first implementation of FC-P-PPF during movement, the dashed red curve shows that of the second implementation, and the dotted blue line shows the lower bound on average RMS error of FC-P-PPF, which corresponds to a known movement duration.

Fig. 5. Comparison of FC-P-PPF and RW-PPF when movement duration is unknown. (a) and (b) show a sample trajectory during a reaching movement by the monkey (solid black) and 10 sample decoded trajectories using RW-PPF in (a) and the first implementation of FC-P-PPF in (b) (dashed blue). The decoders are run until the end of the uncertainty interval. (c) and (d) show the true velocity (black) in the two dimensions and the decoded velocities using RW-PPF (dashed red) and FC-P-PPF (solid blue) for a sample realization of the neural signal. (e) and (f) show the evolution of the optimal combining weights for the branches of FC-P-PPF for the sample realization of the neural signal in (e) and averaged over all 100 realizations in (f).

Fig. 6. Comparison of FC-P-PPF implementations. (a) True trajectory (solid black, same as in Fig. 5) and 10 sample decoded ones (dashed) using the second implementation of FC-P-PPF with unknown duration (compare with Fig. 5b) (b) Combining weights vs. time averaged over all 100 neural realizations for the second implementation of FC-P-PPF (compare with Fig. 5f).(c)–(d) True velocities in the two dimensions (solid black) and decoded velocities for a sample neural realization with the first (solid blue) and second (dashed yellow) implementations of FC-P-PPF. True duration is 296 ms.

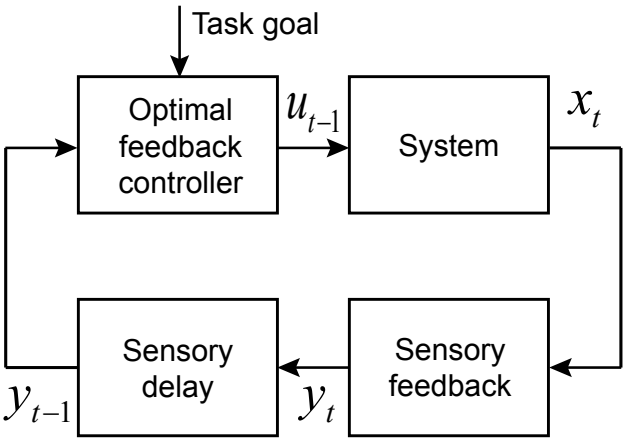

Task goal
Optimal feedback controller
u_{t-1}
System
x_t
Sensory delay
y_{t-1}
y_t
Sensory feedback

Fig. 1.

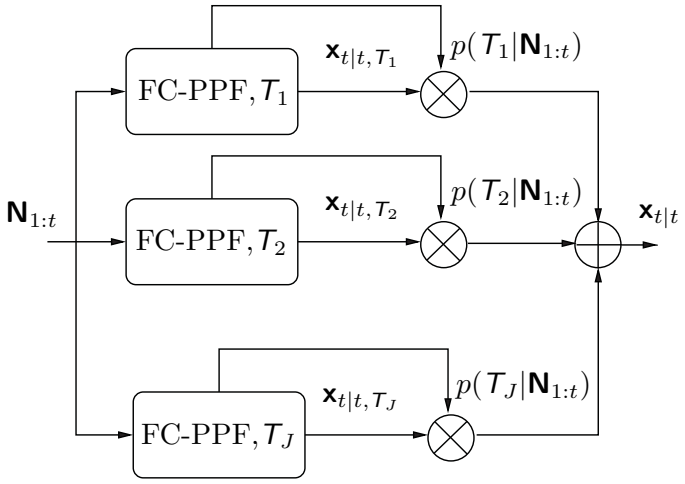

$\mathbf{x}_{t|t,\mathcal{T}_1}$
$p(\mathcal{T}_1|\mathbf{N}_{1:t})$
FC-PPF, $\mathcal{T}_1$
$\mathbf{N}_{1:t}$
$\mathbf{x}_{t|t,\mathcal{T}_2}$
$p(\mathcal{T}_2|\mathbf{N}_{1:t})$
FC-PPF, $\mathcal{T}_2$
$\mathbf{x}_{t|t}$
$\mathbf{x}_{t|t,\mathcal{T}_J}$
$p(\mathcal{T}_J|\mathbf{N}_{1:t})$
FC-PPF, $\mathcal{T}_J$

Fig. 2.

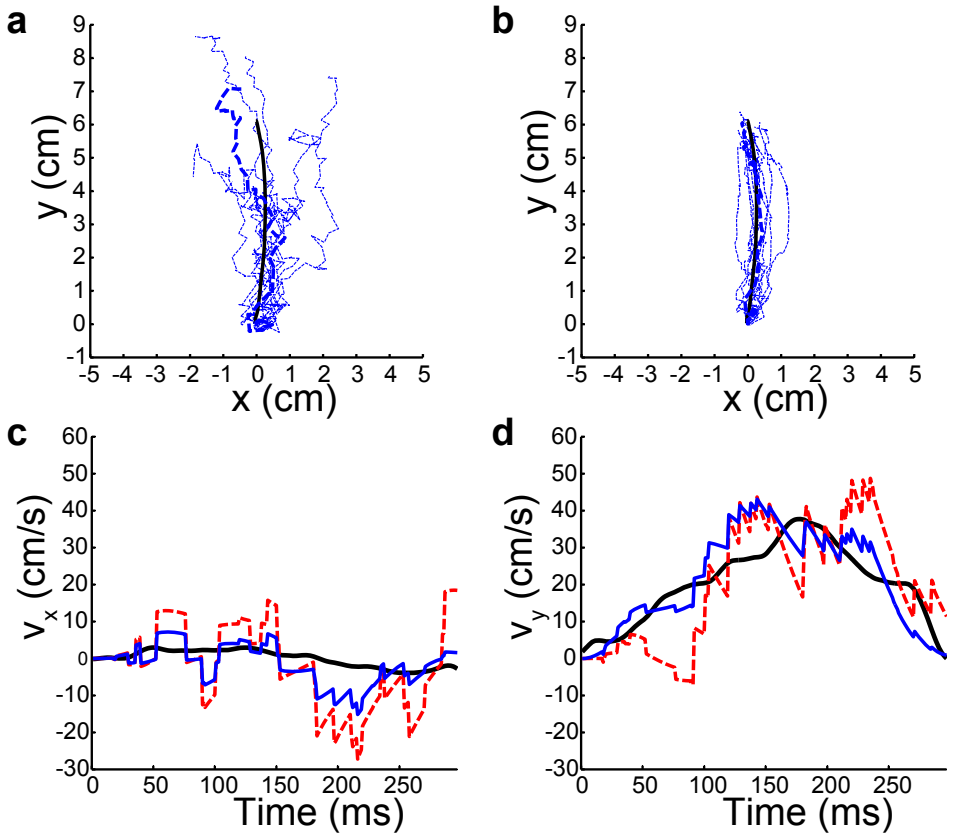


Fig. 3.

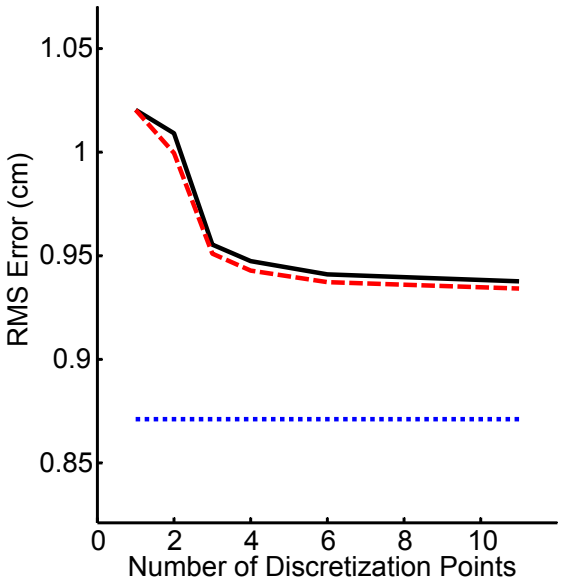

1.05
1
0.95
0.9
0.85
RMS Error (cm)
0
2
4
6
8
10
Number of Discretization Points

Fig. 4.

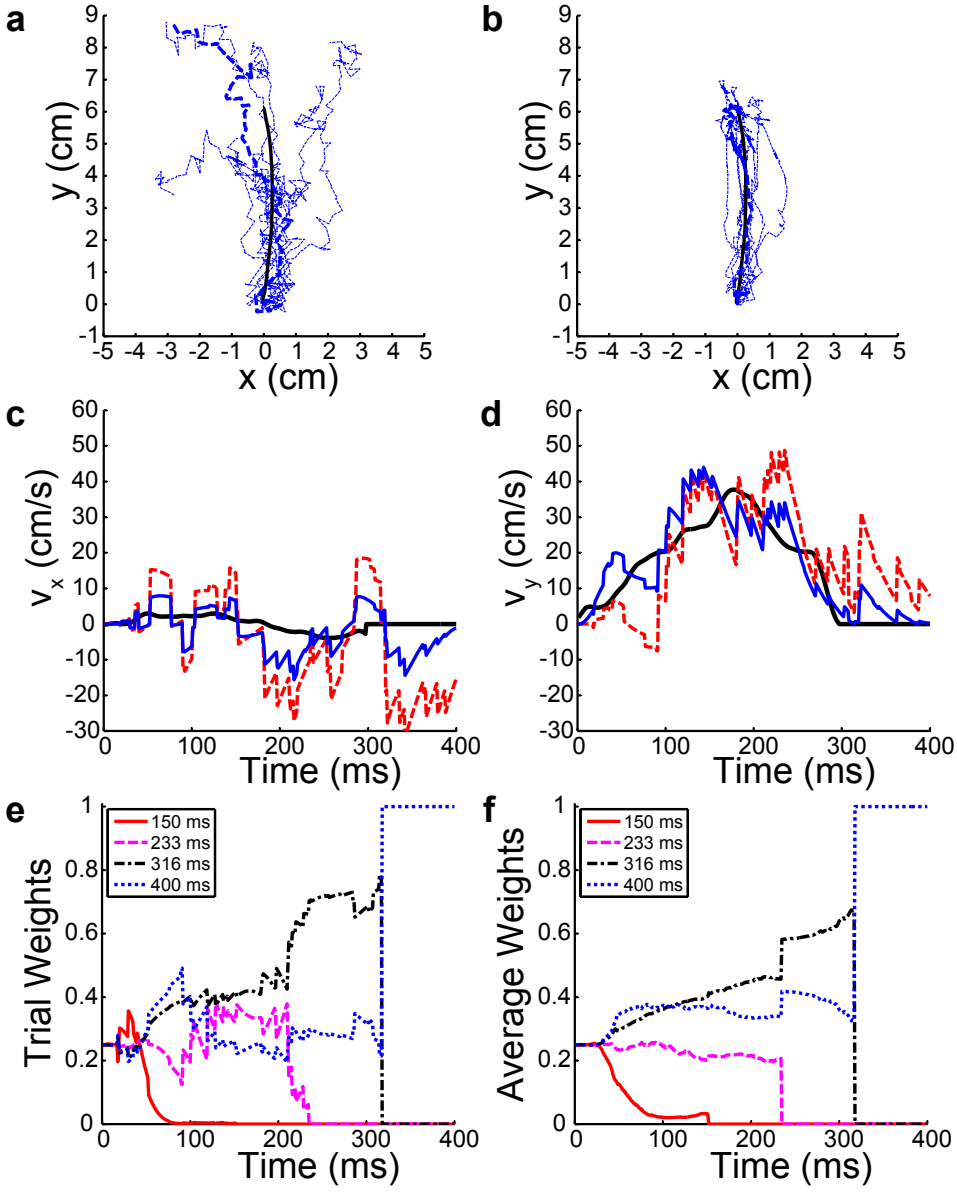

a
b
y (cm)
x (cm)
c
d
v_x (cm/s)
v_y (cm/s)
Time (ms)
e
f
Trial Weights
Average Weights
150 ms
233 ms
316 ms
400 ms

Fig. 5.

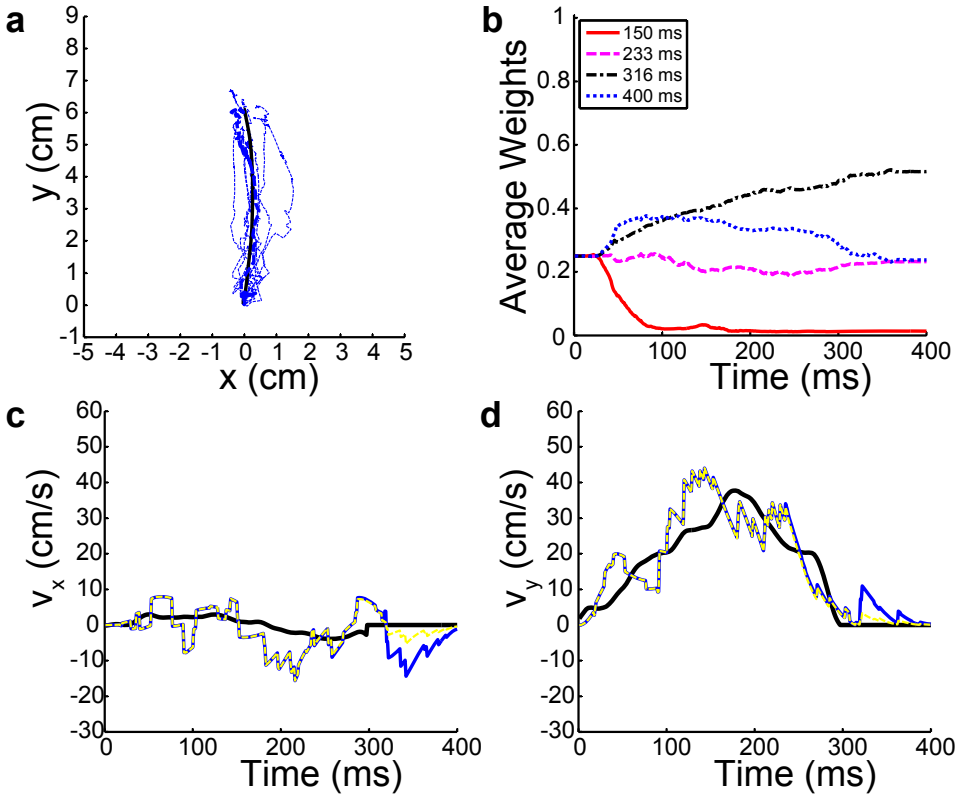

a
y (cm)
x (cm)
b
Average Weights
150 ms
233 ms
316 ms
400 ms
Time (ms)
c
v_x (cm/s)
Time (ms)
d
v_y (cm/s)
Time (ms)

Fig. 6.

A Real-Time Brain-Machine Interface Combining Motor Target and Trajectory Intent Using an Optimal Feedback Control Design

Maryam M. Shanechi[1,2,3*], Ziv M. Williams[2,3], Gregory W. Wornell[1], Rollin C. Hu[2,3], Marissa Powers[2], Emery N. Brown[2,3,4]

1 Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, USA

2 Massachusetts General Hospital, Boston, MA, USA

3 Harvard Medical School, Boston, MA, USA

4 Department of Brain and Cognitive Sciences, Massachusetts Institute of Technology, Cambridge, MA, USA

∗ **E-mail: shanechi@mit.edu**

Abstract

Real-time brain-machine interfaces (BMI) have focused on either estimating the continuous movement trajectory or target intent. However, natural movement often incorporates both. This suggests that BMIs should be designed to combine these principal aspects of movement. We develop a novel real-time BMI that jointly estimates the movement target and trajectory of monkeys in two stages. First, the target is decoded from neural spiking activity before movement initiation. Second, the trajectory is decoded by combining the decoded target with the peri-movement spiking activity using an optimal feedback control design. This design exploits a recursive Bayesian decoder that uses an optimal feedback control model of the sensorimotor system to take into account the intended target location and the sensory feedback in its trajectory estimation from spiking activity. We show that the two-stage BMI performs more accurately than either stage alone. Correct target prediction can compensate for inaccurate trajectory estimation and vice versa. This BMI also performs significantly better than commonly used linear regression approaches. Our results demonstrate the advantage of a BMI design that jointly estimates the target and trajectory of movement and more closely mimics the sensorimotor system.

Introduction

There has been a large body of work in the past decade on real-time brain-machine interfaces (BMI) demonstrating that neural signals from the motor cortical areas can be used to control computer cursors or robotic arms in human and non-human primates [1–16]. One type of such BMIs, which comprises most of this work, aims to estimate a continuous trajectory—for example the position of a computer cursor on the screen moving towards a visual target [1–14]. Recent efforts with this type of BMIs have demonstrated the ability to estimate continuous movement from motor cortical activity. The other type of BMIs aim to predict a desired discrete target without estimating the corresponding desired trajectory towards it [15, 16] and are valuable for purposes such as typing on a keyboard.

These successful real-time attempts at individual decoding of the continuous trajectory or the target of movement motivate the development of a new type of real-time BMIs that aim to estimate jointly both the trajectory of the movement and the intended target. This approach is justified by two main reasons. First, the activity in the motor cortical areas has been shown to be related to both target and kinematics of movement. [2, 15, 17–37]. Peri-movement activity, i.e., the activity around the time of movement, in the primary motor cortex, posterior parietal cortex (PPC), and dorsal premotor cortex (PMd) is related to the movement kinematics such as direction, velocity, position, and acceleration [2, 17–26]. In addition to peri-movement activity, neural activity in the PPC has been shown to encode the intended target [15, 27–30] prior to movement initiation. Similar activity has also been observed in the premotor

cortex including PMd [22, 24, 31–37]. Second, this approach more closely mirrors the natural way in which the sensorimotor system decides on a plan of action and executes its movement. In other words, the several components of the musculo-skeletal system are coordinated in order to reach a target and hence the target of a movement and the desired trajectory to reach it are strongly correlated [38, 39]. Based on these considerations, a more principled BMI design that aims to mirror the sensorimotor system by jointly decoding the movement target and the corresponding trajectory could allow for a more accurate estimation of movement than possible by decoding the target or trajectory alone.

Indeed, there has been a body of offline work demonstrating the advantage of combining both target and trajectory related information in improving the estimation accuracy of movement using simulated [40–43] or previously recorded neural data [11, 44]. Here we develop a real-time BMI that combines information about target and trajectory intent and demonstrate its implementation in sensorimotor tasks performed by two rhesus monkeys. This BMI employs a novel two-stage approach. In the first stage, it uses the neural spiking activity prior to movement initiation to predict the intended target of the movement. In the second stage, it combines this prediction with the peri-movement spiking activity to estimate the movement trajectory. To do so, we take advantage of the optimal feedback control theory of the sensorimotor system [38, 39, 45–47] to build a goal-directed state-space model for the movement which we use in a recursive Bayesian decoder to estimate the trajectory. The two-stage BMI processes the spikes directly in real time, i.e., at the millisecond time-scale on which the neural spiking activity is recorded. Here, we show that the two-stage BMI performs better than either stage alone, demonstrating the advantage of combining both target and trajectory related information in real time. As a baseline, we also make offline comparisons to a commonly used linear regression decoder [2–5, 7] and show that the two-stage decoder outperforms the regression decoder.

Results

We measured the performance of our BMI in standard instructed-delay center-out directional tasks in which two monkeys used a joystick to move a cursor from the center of the screen to one of four targets displayed at its periphery (see Materials and Methods). Unlike 'free-roaming' motor tasks in which the subject can freely move until reaching a target, this task required the monkey to reach the correct target *without* touching any of the incorrect targets under a limited time constraint. Hence only trajectories that reached the correct target and at no point touched an incorrect target placed at the other three sides of the screen were rewarded (Figure 1A). The performance measure used in these experiments was the acquisition accuracy, which is the percentage of trials on which the task is successfully completed. Multi-electrode spiking activity was recorded from PMd and the supplementary motor area (SMA) from which 20 ± 2 neurons (mean $\pm$ s.d.) were isolated and used. At the beginning of each day, the monkey first performed the standard task using a joystick (training session) during which target and kinematic models were constructed. The monkey then performed the same task as before but this time cursor position was controlled by the neural activity recorded from the monkey (BMI sessions; Figure 1A) .

Jointly Decoding the Target and Trajectory Using an Optimal Feedback Control Design

The BMI processing consisted of two stages and decoded two aspects of movement. During the first stage, it used a maximum-likelihood (ML) decoder based on a point process model of the neural spiking activity to predict the monkey's intended target of movement during the delay period after target presentation but before movement initiation (see Materials and Methods). In the second stage the BMI combined this decoded target with the peri-movement activity to estimate the trajectory. In this stage the spiking activity of each neuron was modeled as a point process fitted to position and velocity. The BMI estimated the trajectory using an optimal feedback control design that combined the decoded target with the peri-

movement activity. This design is inspired by the optimal feedback control theory of the sensorimotor system used to explain its function [38,39,45–47], but is cognizant of the fact that in the BMI context the system to be controlled is the BMI as opposed to the musculo-skeletal system (Figure 1B; see Materials and Methods for details). We term the resulting recursive Bayesian decoder, the feedback-controlled parallel point process filter (FC-P-PPF). In the BMI experiments, the decoder updated the estimated position of the cursor in fine-scaled steps of 5 ms in real time, which was also used to bin the spikes.

Model Training and Validation

Models for the BMI were trained on the neural spiking activity during the training session at the beginning of each day. Neural recordings were made during target presentation prior to the presentation of the "go" cue, which signaled that the monkey could move the joystick, as well as during movement itself after the "go" cue (Figure 1A).

Each training session consisted of an average of 89 ± 2 trials. Point process models relating the spiking activity of each recorded neuron to target location and movement kinematics were constructed based on the known target location and cursor position for each trial and the recorded multiple-neuronal activity using the generalized linear models (GLM) framework [48]. Models were then cross-validated (leave-one-out) on the same data by finding the corresponding target predictions and kinematic estimates. Target location was predicted using the ML decoder from the neural spiking activity in the 800 ms delay period prior to the "go" cue. Kinematics were estimated using either the two-stage decoder that combined the target predictions from the first stage with the peri-movement activity or its second stage, but not taking into account the target predicted from the first stage.

During the delay period, the ensemble spiking activity (20 ± 2 neurons) predicted the correct target with high accuracy in the training sessions (leave-one-out cross-validation). The prediction accuracy of the trained point process target models across sessions (see Materials and Methods), measured as the percentage of trials on which the models correctly predicted the target using the delay neural activity, was $81 \pm 3\%$. To examine further the contribution of the individual neurons to the target prediction accuracy, we performed a neuron dropping analysis in which the spiking activity of a single neuron during the delay period was used to decode the target (Figure S1A–D). We found that across sessions, $48 \pm 12\%$ of the neurons had a target prediction accuracy significantly greater than chance ($P < 0.05$). We further found that relatively few neurons (on average 3.3 ± 1.0 across sessions) were sufficient to obtain a target prediction accuracy that was higher than 90% that of the ensemble (Figure S1E).

During movement, the premotor neurons were tuned to position and velocity. Fitting the point process models for the kinematics using the GLM framework [48] (see Materials and Methods), we found that across sessions $47 \pm 16\%$ of the premotor neurons were significantly tuned to either position or velocity at least in one dimension ($P < 0.05$; Bonferroni correction for multiple comparisons). Of these neurons, 57% were tuned to position only, 15% were tuned to velocity only, and 28% were tuned to both position and velocity. In agreement with previous studies, these findings suggested that the recorded premotor neurons held significant information about both the target and kinematics of the movement.

Offline Model Comparisons

We tested the performance of the two-stage decoder in an offline analysis of the training sessions data using leave-one-out cross-validation. We also compared to the performance of the second stage of the decoder alone by replacing the feedback-controlled state-space model in the FC-P-PPF (see Materials and Methods) with a random-walk (RW) model, which uses no prior target information and only enforces smoothness in the trajectory. The resulting filter is the RW-PPF (see Materials and Methods). We also compared the performance of the two-stage decoder to that of the commonly used least-squares linear regression decoder [2–5, 7]. For each decoder we updated the position estimate every 5 ms, which was also the bin width for the spiking activity. Note that the chance level acquisition accuracy in our task

should be approximately 25% since there are four targets on the screen and hitting the wrong one results in an error. We confirmed this by estimating the trajectory from shuffled neural activity (see Materials and Methods).

We found that the acquisition accuracy of the two-stage decoder across sessions was 83 ± 3% compared to 61 ± 7% for the RW-PPF, which used only the peri-movement activity (Figure 2A). Hence target predictions from the first stage of the decoder resulted in a correction rate of $(83-61)/(100-61) \approx 56\%$ for the inaccurate trajectory estimation in the second stage (Figure 3B). Also, the trajectories estimated by the two-stage decoder were closer to the monkey's trajectory (Figure 3A,B). To quantify this, we measured the average root mean-square (RMS) error across all trials (Figure 2B). We found that the RMS error of the RW-PPF on average was 40% higher than the two-stage decoder. Comparing the accuracy of the two-stage decoder to the target prediction accuracy of its first stage, we found that the second stage resulted in a correction rate of $(83-81)/(100-81) \approx 11\%$ for the target prediction (Figure 3C). However, as these were offline estimation of fast joystick movements, the more appropriate test for the correction of the incorrect target predictions by the peri-movement activity in the second stage is in real-time BMI sessions (see below).

We also compared the performance of the two-stage decoder to that of a linear regression decoder (Figure 3). The regression decoder reconstructed the position at each time as a linear function of the history of the ensemble firing rates, which were calculated every 5 ms in sliding bins of 100 ms. We included up to 800 ms—same duration as the delay period used for target prediction—of history coefficients in the regression decoder. Using an exhaustive search we used the number of history coefficients that maximized the accuracy of this decoder. The average accuracy of the regression decoder across sessions was 47 ± 8%, which was 36% lower than the two-stage decoder (Figure 2A). Also the average RMS error of the regression decoder was 65% higher than the two-stage decoder (Figure 2B). Note that we allowed for a fair comparison to the linear regression decoder by adding history coefficients.

Finally the trajectory estimations in the two-stage decoder were smoother than either the RW-PPF or the regression decoder (Figure 3). To quantify this, we calculated the average roughness coefficient [49] (see Materials and Methods) for each of the decoders (Figure 2C). The roughness coefficient measures the degree of smoothness in the estimated trajectory and is smaller for smoother estimates. We found that the average roughness coefficients of the RW-PPF and the linear regression decoder were 4.5 and 9.6 times larger than that of the two-stage decoder, respectively.

Combined Target and Trajectory Decoding in a Real-Time BMI

To investigate whether kinematic and target related activity can be jointly used to obtain accurate motor performance in real time, the monkeys performed the same task as before but using a BMI. The real-time BMI used the two-stage decoder. During the 800 ms delay period prior to the "go" cue, the BMI predicted the target and after the "go" cue it combined this target information with the peri-movement activity using the FC-P-PPF to decode the trajectory. We found that using the two-stage BMI, the monkeys could perform the task with an average accuracy of 77 ± 9% (Figure 4A). To assess the stability of the performance throughout the recordings per day, we compared it in the first and second half of sessions. We found that accuracy did not change significantly (Wilcoxon rank-sum test, $P > 0.5$) and remained stable.

We next compared the accuracy of the two-stage decoder to that for each of its stages alone and found that the two stages of the BMI performed in a complementary manner; the correct target predictions could compensate for the inaccurate performance of the kinematics decoder and the ongoing trajectory estimation could correct the incorrect target predictions. To compare the accuracy of the BMI with that of using only the peri-movement activity, i.e., using the second stage of the BMI alone without target information, we decoded the trajectory on the same real-time data set using the RW-PPF (Figure 4B–D). The average acquisition accuracy of the RW-PPF was 61 ± 31% (Figure 4A). This suggests that the first stage of the BMI resulted in a correction rate of $(77-61)/(100-61) \approx 41\%$ for the second

stage (Figure 4A,C). The average real-time target prediction accuracy of the first stage of the BMI was 72 ± 3%. This indicates that the second stage of the BMI (FC-P-PPF) resulted in a correction rate of $(77 - 72)/(100 - 72) \approx 18\%$ for the target predictions of the first stage (Figure 4A,D). Hence the joint performance of the BMI was higher than would have been possible using either kinematic or target related activity alone.

The acquisition time in the BMI sessions, i.e., time until the trial ended by rewarding the monkey, was close to the natural acquisition time in the training sessions. In our experiments we used a short 3 s time-out condition to make the task more challenging and the required acquisition time closer to that of monkey's own movement. The median acquisition time for the natural movement was 0.6 ± 0.3 s and for the two-stage BMI was 0.9 ± 0.5 s.

Control Comparisons

First, to relate the performance of the BMI to the linear regression decoder that is frequently used in real time, we compared the accuracy of our two-stage BMI with that of the regression decoder on the same real-time data set (Figure 5). The linear regression decoder used up to 800 ms of history coefficients (same duration as the delay period used for target prediction) and the number of history coefficients in it was chosen by an exhaustive search and to maximize its accuracy. The average accuracy of the linear regression decoder was 44 ± 4%, which is significantly lower than the two-stage BMI or either of its stages alone (Figure 4A). Comparing the trajectories produced by the BMI to those of the linear regression decoder we observed that they were smoother. This was the case even in trials in which both decoders acquired the correct target (Figure 5A) where the average roughness coefficient in the regression decoder was 3.3 times that of the two-stage BMI.

Second, to determine differences in BMI performance across monkeys and therefore the robustness of the two-stage BMI to individual variability, we examined differences in performances between the two monkeys, A and B. For monkey A, the real-time BMI accuracy was 67 ± 4%. Comparing to the second stage alone using only the peri-movement activity, we found that the accuracy of the RW-PPF on the real-time data set for this monkey was 29 ± 1%. Hence, even though the kinematic tuning in this monkey was weak, the real-time BMI had a relative high accuracy. This showed that the first stage of the BMI resulted in a correction rate of $(67 - 29)/(100 - 29) \approx 54\%$ for the inaccurate performance of the trajectory decoder. The real-time target prediction accuracy of the first stage using only the delay activity in this monkey was 71 ± 1%. Note that because of the weak kinematic tuning in the recorded neurons in this monkey, the correction happened only by the first stage. The accuracy of the linear regression decoder on the same real-time data set in this monkey was 45 ± 5%, significantly lower than that of the two-stage BMI using the same neural activity. For monkey B, the real-time BMI accuracy was 82 ± 5%. Comparing to the second stage alone using only the peri-movement activity, we found that the accuracy of the RW-PPF in this monkey was 76 ± 24% and hence the first stage of the BMI resulted in a correction rate of $(82 - 76)/(100 - 76) \approx 25\%$ for the inaccurate trajectory estimation of the second stage. Comparing to the average real-time target prediction accuracy of the first stage, which was 73 ± 4%, suggested that the second stage of the BMI in monkey B resulted in a correction rate of $(82 - 73)/(100 - 73) \approx 33\%$ for the target predictions in the first stage. Therefore, in this monkey both stages exhibited the corrective behavior. The average accuracy of the linear regression decoder on the same real-time data set in this monkey was 43 ± 4%, significantly lower than the two-stage BMI. Together, these findings suggest that although the performance of one stage in the two-stage BMI may not be equal across monkeys or recording areas, by combining both target and trajectory related information the two-stage approach provides a robust computational system that maintains good accuracy under variable experimental conditions.

Discussion

Based on our understanding of the sensorimotor system [38, 39, 45–47], natural movement incorporates information about the intended target as well as the trajectory of the movement. We implemented a novel real-time BMI designed to mimic the sensorimotor system by a two-stage approach: First the activity prior to movement initiation is used to predict the intended target, and second this prediction is combined with the peri-movement spiking activity to estimate the trajectory using an optimal feedback control design. This, to our knowledge, is the first time that combined estimation of target and trajectory, and direct point process modeling of the spiking activity have been done in real time. Also, it is the first time that sensorimotor control principles have been used to decode the intended movement from ensemble neural recordings.

To combine the decoded target and the peri-movement activity, we took advantage of the optimal feedback control theory of the sensorimotor system. In this view, each task is performed to accomplish a goal during which there is sensory feedback about the external state of the system. Specifying an approximate kinematic model, modeling the sensory feedback about the state of the system, and quantifying the task goals as cost functions and the desired time to accomplish them, we can predict the next plan of action or control signal by finding the one that minimizes the cost function. For a BMI, the same framework can be applied to predict the next plan of action. The difference is that the system being controlled is the BMI as opposed to an individual's own musculo-skeletal system. Hence the individual's next plan of action is reflected in the neural activity that, in turn, controls the BMI. Also, the BMI does not have knowledge of the desired time to accomplish the goal, which is decided by the controller (the individual). The present BMI hence resolved this movement duration uncertainty based on the neural spiking activity in real time. The BMI also processed the spiking activity directly and hence operated at the millisecond time-scale of the spikes. In addition to its application to interpreting the sensorimotor function, optimal feedback control has also been deemed valuable for interpreting the neural basis of movement in the motor cortical areas [50]. This further motivates the use of optimal feedback control principles for the design of real-time BMIs.

We demonstrated that the two stages in the BMI functioned in a complementary manner. When the spiking activity for one stage was less informative, the other stage often provided sufficient information for the BMI to reach the correct target. As a result, the two-stage BMI performed better than either stage alone. Overall, the estimated trajectories using the two-stage approach were more accurate, had lower RMS error, and were smoother than the linear regression decoder or a random-walk point-process decoder. Unlike 'free-roaming' motor tasks in which subjects could move freely until reaching a target, the present task was demanding in that at no point the trajectories could touch the incorrect targets and then proceed to the correct target. This was considered an incorrect response. In addition the response time was constrained. Despite this, the monkeys were able to achieve a relatively high accuracy using the two-stage BMI ($77 \pm 9\%$). This accuracy was obtained by using relatively few neurons (11 on average) that were tuned to either target or trajectory.

Our BMI used direct point process modeling of the spiking activity. It hence processed the spikes directly in real time as opposed to a smoothed rate function calculated from these spikes as is done in previous real-time BMI work. Recent work [51] has demonstrated that reducing the bin width used to calculate the firing rates of the spiking activity, which are in turn used as input in a BMI, improves its performance. An interesting question for future investigation in real-time experiments is therefore whether moving to the time-scale on which the spiking activity is recorded, i.e., processing the spikes directly, could improve the performance of real-time BMIs.

In the two-stage BMI, one stage compensated for the inaccuracies of the other and vice versa. Since our model relies on neural activity to estimate the movement, we cannot test directly to what extent the brain uses these two aspects of motor control to execute a movement. It is interesting to speculate, however, that similar to findings made in these experiments, the premotor cortex may use information on intended target location to correct for discrepancies in ongoing movement. Similarly, it may use

information about ongoing movement to fine-tune differences between initially intended target location and target location during movement itself. Hence, in addition to enabling the design of more accurate decoding algorithms, the present feedback-control design may provide insight into the control aspects of the motor function in natural settings. Therefore the two-stage BMI could provide an important and unique new step in developing neuroprosthetics that take advantage of the multiple types of movement information.

Materials and Methods

Ethics Statement

This study was performed in strict accordance with the recommendations in the Guide for the Care and Use of Laboratory Animals of the National Institutes of Health, and under regulation of the Subcommittee on Research Animal Care at Harvard Medical School. The protocol was approved by the Institutional Animal Care and Use Committee for Massachusetts General Hospital (animal welfare assurance number: A3596-01). All surgery was performed under inhalational anesthesia in combination with Ketamine and opiate anaelgesia, and every effort was made to minimize suffering, in accordance with the recommendations of the Weatherall report, "The use of non-human primates in research".

Behavioral Task

We used two adult male rhesus monkeys (*macaca mulatta*) in the study. During the tasks, the animals were seated in a primate chair (Crist Instrument Co Ltd, Damascus, MD). The primates' head were restrained using a head post, and a spout was placed in front of their mouth to deliver juice using an automated solenoid. A spring-loaded, two-degrees of freedom manipulandum was mounted anterior to chair on the side contralateral to recording. A computer monitor was placed in front of the animals at eye level which displayed the task. A NI DAQ card (National Instruments, TX) was used for the I/O behavioral interface, and the behavioral program was run in Matlab (MathWorks, MA) using custom made software (www.monkeylogic.net).

Primates performed a center-out visually-instructed motor directional task. The monkeys held the joystick contralateral to the site of recordings and could move their limb freely in the horizontal and vertical dimensions during the task. A computer monitor displayed the target locations and a cursor was used to represent the position of the joystick handle. Each individual trial began with the presentation of a central fixation point surrounded by four gray circular targets. Once the animals held the cursor within a central radius for a delay of 500 ms, one of the four randomly selected targets would turn green. After another 1000 ms, the fixation point would change color ("go" cue), at which time the monkeys could use the joystick to move the cursor from the center of the screen to the instructed target. Once the cursor reached the target, the animal received a drop of juice following a 320 ms delay if the correct target was selected. If during a trial, the animals moved prematurely, failed to reach any target during the allowed time or touched an incorrect target before reaching the correct target, the trial aborted. Once reward was delivered, another 1000 ms would lapse, the targets would erase, and the sequence would repeat again. The animals were required to return the spring-loaded joystick to the center fixation point before a new trial began.

Neurophysiologic Recordings and BMI Setup

A titanium head post and recording electrodes were surgically implanted in each monkey contralateral to the side of joystick use. All procedures were performed in an IACUC-approved aseptic primate surgical facility. Prior to electrode implantation, craniotomies were performed over the sites of interest using standard stereotactic coordinates. Once the cortex was exposed and the sulcal anatomy identified, several

silicone microelectrode arrays were placed in the cortex (Neuronexus technologies, MI). The electrodes were secured into place using fibrin glue, silicone sealant, and methylmethacrolate. The distal leads were then attached to a female connector and secured to the skull with titanium miniplates and dental acrylic. Anatomic post-mortem confirmation of electrode positioning was performed in one monkey. The second monkey is still performing behavioral tasks.

Recordings began at two weeks following surgical recovery. A Plexon multichannel acquisition processor was used to amplify and band-pass filter the neuronal signals (150 Hz − 8 kHz; Plexon Inc., TX). Shielded cabling carried the signals from the electrode array to a set of six 16-channel amplifiers. Signals were then digitized at 40 kHz and processed to extract action potentials in real time by the Plexon workstation. Classification of the action potential waveforms were accomplished using dual-window discrimination and principle component analysis. Units with stable, identifiable waveform shapes and adequate refractory periods determined by autocorrelation were then used for the real-time experiments. Joystick position was sampled and recorded at 1 kHz. Neuronal data obtained from the Plexon workstation, in the form of action potential time stamps and channel, were then transmitted to a second PC computer running Matlab (Mathworks, MA) in real time. On decoder training sessions, the primates would use the joystick to move a cursor on the screen to one of four randomly selected targets over multiple trials. On decoder BMI sessions, the monkeys would still be allowed to use the joystick but the cursor image displayed on the screen would be supplied by the Matlab real-time decoder (Figure 1A). The cursor was initially placed at the center fixation point at the beginning of each trial. Here, estimated cursor movements would be relayed through a DAQ I/O (National Instruments, TX) to a third PC computer running the behavioral task. The computer would then display the estimated cursor position.

Chance Level Accuracy

To find the chance level accuracy, we used the least-squares linear regression decoder to estimate the trajectory from shuffled neural activity. To do so we shuffled the calculated firing rates for each neuron across time and trials, keeping its average firing rate the same.

Target Decoding

The BMI decodes the monkeys' intended target of movement by recording the ensemble spiking activity during the 800 ms delay interval prior to the "go" cue. Note that the delay between the start of target presentation and "go" cue is 1000 ms. We do not use the activity in the first 200 ms in the BMI in order to allow sufficient time for the visual target information to reach the PMd and SMA [16]. Using offline cross-validation analyses we observed that discarding this activity improves the prediction accuracy. Spiking activity of each neuron during this delay interval is modeled as a homogeneous Poisson process (a point process with constant rate) whose firing rate is a function of the intended target, fitted using the GLM framework. A maximum-likelihood (ML) decoder first calculates the likelihood probability of this ensemble activity for each possible target, G, and then selects the target with the highest likelihood as its prediction. Denoting the neural point process observations of the ensemble of C neurons by $\mathbf{N}_1, \cdots, \mathbf{N}_t$ where $\mathbf{N}_t = (N_t^1, \cdots, N_t^C)$ is the binary spike events of the C neurons at time t, and assuming that the neurons are conditionally independent given the target, the point process likelihood model for the ensemble is given by [48]

$$p(\mathbf{N}_t|G) = \prod_{c=1}^{C} (\lambda_c(G)\Delta)^{N_t^c} \, e^{-\lambda_c(G)\Delta} \tag{1}$$

where $\Delta = 5$ ms is the time increment used for binning the spikes and $\lambda_c(G)$ is the modeled firing rate of the cth neuron during the delay period for target G. The ML decoder then predicts the target as the

one maximizing the ensemble likelihood,

$$\hat{G} = \arg\max_{G=1,\cdots,4} p(\mathbf{N}_1, \cdots, \mathbf{N}_{T_d/\Delta}|G) = \arg\max_{G} \prod_{c=1}^{C} \prod_{t=1}^{T_d/\Delta} (\lambda_c(G)\Delta)^{N_t^c} \, e^{-\lambda_c(G)\Delta} \tag{2}$$

Here $T_d = 800$ ms is the delay period. During the delay period the cursor was held at the center of the screen.

Kinematic Decoding

In the second stage, the BMI combines the decoded target with the peri-movement ensemble spiking activity using a recursive Bayesian decoder for goal-directed movements. A recursive Bayesian decoder in general consists of two probabilistic models: the prior model on the time sequence of kinematic states, and the observation model relating the neural signal to these states. The prior model should in general incorporate any prior information available about the kinematic states, which for a goal-directed movement includes the intended target. We build the prior model of the decoder using an optimal feedback control design, which takes into account the sensory feedback . To develop the observation model, we use a point-process model of the spiking activity whose instantaneous rate is a log-linear function of kinematics. The resulting decoder hence processes the spikes directly in real time and operates at the millisecond time scale of the spiking activity. In the next few sections we present the prior and observation models used in the decoder and its recursions.

Prior Model: an Optimal Feedback Control Design

Previous offline work have built goal-directed prior models for reaching movements by conditioning a linear Gaussian state-space model, also known as a random-walk model, on being at the target at a *known* arrival time [40,52] or using a linear feedforward controlled (i.e., not taking into account the sensory feedback) model again assuming a known arrival time [41]. Alternatively, goal-directed prior models have been built by using a training data set, for example fitting a linear Gaussian state-space model for a given target to empirical reaches to its location [44] or fitting a single model for arbitrary targets based on a data set of reaches to their locations [11]. Since the goal of the decoder is to estimate intended movement, a prior model that aims to emulate the sensorimotor processing underlying actual motor control could result in more accurate estimation of movement. Hence we build the prior goal-directed state-space model for the kinematics based on the optimal feedback control theory of the sensorimotor system [38, 39]. This theory has been successfully used to interpret the sensorimotor function. In this optimal feedback control framework, each task is performed to accomplish a goal during which there is sensory feedback about the state of the musculo-skeletal system. Based on the intended goal, the sensory feedback (for example vision and proprioception) about the current state of the system, and the desired time to accomplish the goal, the subject (controller) decides on the next plan of action or control signal (for example muscle activation) and can hence make real-time adjustments based on the feedback to improve behavior. Specifying an approximate kinematic model and quantifying the task goals as cost functions and also the sensory feedback, this framework can predict the next plan of action in the presence of model uncertainty and sensory noise. The difference in applying this framework to natural movement [38, 39] and movement using a BMI is that in the former case the system to be controlled is the musculo-skeletal system and in the latter, it is the BMI (Figure 1B). While performing the task, the monkey decides on the next plan of action based on the sensory feedback of the cursor position and the intended target. Similar to natural movement, in movement using a BMI, the next plan of action is in turn reflected in the neural activity but this time controls the system or BMI through the decoder (as we will develop) as opposed to directly controlling the musculo-skeletal system (Figure 1B).

Motivated by this view, we develop a prior feedback-controlled state-space model for the kinematics that exploits the information about the target location and models the sensory feedback. Based on this model, the decoder can predict the monkey's next plan of action or control signal and consequently the next kinematic state. Note that our prior model does not rely on a training data set, as is the case in [11,44], and can therefore easily extend to different target locations without requiring a set of empirical reaches to these locations. Finally, by using the optimal feedback control formulation, it could generalize to tasks other than reaching movements, if desired, by simply quantifying the goals of such tasks as the cost function in this formulation. We now present the construction of this model. We have provided more algorithmic details about this model in a simulation study [43] assuming knowledge of the target.

Denoting the sequence of kinematic states by $\mathbf{x}_0, \cdots, \mathbf{x}_t$, we assume they are generated according to the linear dynamical system,

$$\mathbf{x}_{t+1} = \mathbf{A}\mathbf{x}_t + \mathbf{B}\mathbf{u}_t + \mathbf{w}_t. \tag{3}$$

Here, $\mathbf{u}_t$ is the control signal at time t, which is decided by the controller (the primate in the BMI context), $\mathbf{w}_t$ is the zero-mean white Gaussian state noise with covariance matrix $\mathbf{W}$, and $\mathbf{A}$ and $\mathbf{B}$ are parameters of the kinematic model. Here we assume that the sensory feedback, $\mathbf{y}_t$, is noiseless and $\mathbf{y}_t = \mathbf{x}_t$. This means that we assume the monkey has perfect sensory feedback of the cursor position on the screen. To find $\mathbf{u}_t$ in the control framework, we need to specify a cost function that will then be minimized by optimizing over $\mathbf{u}_t$. The cost function in a given task should quantify its goal. For the above linear Gaussian dynamics, if we pick the cost function as a quadratic function of the state and control variables, i.e.,

$$J = \sum_{t=1}^{T-1} (\mathbf{x}_t'\mathbf{Q}_t\mathbf{x}_t + \mathbf{u}_t'\mathbf{R}\mathbf{u}_t) + \mathbf{x}_T'\mathbf{Q}_T\mathbf{x}_T, \tag{4}$$

where T is the movement duration, $\mathbf{Q}_t$ is positive semidefinite and $\mathbf{R}$ is positive definite, then the optimal control signal at any time, $\mathbf{u}_t$, is simply a linear *feedback* of the state at that time [53], i.e.,

$$\mathbf{u}_t = -\mathbf{L}_t(T)\mathbf{x}_t, \tag{5}$$

where $\mathbf{L}_t$ can be found recursively and offline [53]. This is the well-known linear quadratic Gaussian (LQG) solution . Substituting (5) in (3) reduces this state-space model to the optimal feedback-controlled state-space model

$$\mathbf{x}_{t+1} \quad = \quad (\mathbf{A} - \mathbf{B}\mathbf{L}_t(T))\mathbf{x}_t + \mathbf{w}_t, \tag{6}$$

which can now be used as the prior model to make prediction on the kinematic states. Note that $\mathbf{Q}_t$ and $\mathbf{R}$ should be appropriately designed for an application of interest and $\mathbf{L}_t(T)$ is time-varying and a function of the duration T. Note also that the sensory feedback is incorporated in the control term since the control term is simply a linear function of the current kinematic state (see (5)), assumed to be known through the feedback. This is in turn reflected in the prior model in (6).

We can now specialize these to the reaching movements used in our experiments. For a reaching movement the cost function should enforce end-point positional accuracy, stopping condition, and energetic efficiency [46, 47]. Denoting the desired target position by $\mathbf{d}^*$ and taking the state to be $\mathbf{x}_t = [\mathbf{d}_t, \mathbf{v}_t, \mathbf{a}_t]'$ where the components represent position, velocity and force in the two dimensions respectively, similar to previous studies [46, 47] we take this cost function to be the weighted sum

$$J = \| \mathbf{d}_T - \mathbf{d}^* \|^2 + w_v \| \mathbf{v}_T \|^2 + w_a \| \mathbf{a}_T \|^2 + w_r \sum_{t=1}^{T-1} \| \mathbf{u}_t \|^2 \tag{7}$$

where the weights are chosen to penalize the terms in the cost function approximately equally on average [46, 47]. We adapt the following first order lowpass muscle-like system [47] for the dynamical system in

(3) in each dimension,

$$\begin{bmatrix} d_{t+1} \\ v_{t+1} \\ a_{t+1} \end{bmatrix} = \begin{bmatrix} 1 & \Delta & 0 \\ 0 & 1-\frac{b\Delta}{m} & \frac{\Delta}{m} \\ 0 & 0 & 1-\frac{\Delta}{\tau} \end{bmatrix} \begin{bmatrix} d_t \\ v_t \\ a_t \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{\Delta}{\tau} \end{bmatrix} u_t + \begin{bmatrix} 0 \\ 0 \\ w_t \end{bmatrix} \tag{8}$$

where the parameters $b = 10$ Ns/m, $\tau = 0.05$ s, and $m = 1$ kg come from biomechanics [47]. Having specified the kinematic model and the cost function for the reaching movement, the feedback matrices $\mathbf{L}_t(T)$ can now be easily precomputed offline from the recursive solution of LQG [53] and stored for real-time use. Having these matrices, we can predict the monkey's next plan of action reflected in the control signal, $\mathbf{u}_t$, using (5). Note that in our two-stage decoder $\mathbf{d}^*$ is determined from the decoded target location in the first stage.

Observation Model

We build the observation model for each neuron as a point process whose instantaneous firing rate is a function of kinematics [48], i.e.,

$$p(N_t^c|\mathbf{x}_t) = (\lambda_c(t|\mathbf{x}_t)\Delta)^{N_t^c} e^{-\lambda_c(t|\mathbf{x}_t)\Delta}$$

We used a modified version of the cosine tuning model [17,20] for the instantaneous firing rate, modeling it as a log-linear function of position and velocity in the two dimensions [48], i.e.,

$$\lambda_c(t|\mathbf{x}_t) = \exp(\alpha_0^c + \boldsymbol{\alpha}_c'\mathbf{x}_t), \tag{9}$$

where $\mathbf{x}_t$ denotes these kinematic states at time t and α_0^c and $\boldsymbol{\alpha}_c$ are fitted using the GLM framework [48] on the peri-movement spiking activity. More specifically, denoting the model parameters for neuron c by $\boldsymbol{\phi}_c = [\alpha_0^c, \boldsymbol{\alpha}_c]$, the GLM framework finds the maximum likelihood estimate

$$\hat{\boldsymbol{\phi}}_c = \arg\max_{\boldsymbol{\phi}_c} p(N_{1:T_t}^c|\mathbf{x}_{1:T_t}; \boldsymbol{\phi}_c)$$

where $N_{1:T_t}^c$ is the peri-movement spiking activity of the neuron during training and $\mathbf{x}_{1:T_t}$ are the corresponding kinematic states. Using the GLM framework, P-values can also be obtained for all the model parameters [48] (for example using the glmfit function in Matlab) and hence the tuning properties of the neurons can be examined.

We assumed that the spiking activity of the neurons are conditionally independent given the kinematic states and hence the observation model for the ensemble is given by

$$p(\mathbf{N}_t|\mathbf{x}_t) = \prod_{c=1}^{C} (\lambda_c(t|\mathbf{x}_t)\Delta)^{N_t^c} e^{-\lambda_c(t|\mathbf{x}_t)\Delta} \tag{10}$$

Uncertainty in the Movement Duration

Having the prior and the observation models we can now develop the recursions for the Bayesian decoder. However, the prior model built in (6) is dependent on the movement duration, T, which is not known to the decoder. In other words, unlike natural movement in which the monkey (controller) decides on the movement duration, in movement using a BMI the decoder does not have a priori knowledge of this duration. This is typically the case for goal-directed state-space models as there is much more constraint on the movement kinematics close to the arrival time at the target compared to far from it since in the former case the trajectory soon needs to converge to the intended target. Hence we develop the BMI decoder to jointly resolve this duration uncertainty and estimate the trajectory purely based on the neural spiking activity. We first present the recursions of a feedback-controlled point-process filter assuming a known movement duration and then show how we can resolve the duration uncertainty inherent to the prior model.

Feedback-Controlled Point Process Filter (FC-PPF) for a Known Movement Duration

For now we assume that the movement duration is known. The minimum mean-square error (MMSE) estimator is given by the mean of the posterior density that is $p(\mathbf{x}_t|\mathbf{N}_{1:t}, T)$ for a given duration T. Denoting the one step prediction mean by $\mathbf{x}_{t|t-1,T} = E(\mathbf{x}_t|\mathbf{N}_{1:t-1}, T)$, its covariance matrix by $\mathbf{W}_{t|t-1,T}$, the MMSE estimate by $\mathbf{x}_{t|t,T}$, and finally its covariance matrix by $\mathbf{W}_{t|t,T}$, $\mathbf{x}_{t|t,T}$ is found from the following recursions

$$\mathbf{x}_{t|t-1,T} = (\mathbf{A} - \mathbf{B}\mathbf{L}_t(T))\,\mathbf{x}_{t-1|t-1,T} \tag{11}$$

$$\mathbf{W}_{t|t-1,T} = (\mathbf{A} - \mathbf{B}\mathbf{L}_t(T))\,\mathbf{W}_{t-1|t-1,T}\,(\mathbf{A} - \mathbf{B}\mathbf{L}_t(T))' + \mathbf{W}_t \tag{12}$$

$$\begin{aligned}\mathbf{W}^{-1}_{t|t,T} = \mathbf{W}^{-1}_{t|t-1,T} + \sum_{c=1}^{C}\Bigg[&\left(\frac{\partial \log \lambda_c(t|\mathbf{x}_t)}{\partial \mathbf{x}_t}\right)'\left(\frac{\partial \log \lambda_c(t|\mathbf{x}_t)}{\partial \mathbf{x}_t}\right)\lambda_c(t|\mathbf{x}_t)\Delta \\ &-(N^c_t - \lambda_c(t|\mathbf{x}_t)\Delta)\frac{\partial^2 \log \lambda_c(t|\mathbf{x}_t)}{\partial \mathbf{x}_t \partial \mathbf{x}_t'}\Bigg]_{\mathbf{x}_{t|t-1,T}}\end{aligned} \tag{13}$$

$$\mathbf{x}_{t|t,T} = \mathbf{x}_{t|t-1,T} + \mathbf{W}_{t|t,T}\sum_{c=1}^{C}\left[\left(\frac{\partial \log \lambda_c(t|\mathbf{x}_t)}{\partial \mathbf{x}_t}\right)'(N^c_t - \lambda_c(t|\mathbf{x}_t)\Delta)\right]_{\mathbf{x}_{t|t-1,T}} \tag{14}$$

where $[\cdot]_{\mathbf{x}_{t|t-1,T}}$ denotes the evaluation of the expression at $\mathbf{x}_{t|t-1,T}$. These recursions are obtained using (6) for the prediction step and a Gaussian approximation for the update step as previously derived [54]. For the rate model in (9) since $\frac{\partial \log \lambda_c}{\partial \mathbf{x}_t} = \boldsymbol{\alpha}_c'$ and $\frac{\partial^2 \log \lambda_c}{\partial \mathbf{x}_t \partial \mathbf{x}_t'} = \mathbf{0}$ these recursions simplify to

$$\mathbf{x}_{t|t-1,T} = (\mathbf{A} - \mathbf{B}\mathbf{L}_t(T))\mathbf{x}_{t-1|t-1,T} \tag{15}$$

$$\mathbf{W}_{t|t-1,T} = (\mathbf{A} - \mathbf{B}\mathbf{L}_t(T))\mathbf{W}_{t-1|t-1,T}(\mathbf{A} - \mathbf{B}\mathbf{L}_t(T))' + \mathbf{W}_t \tag{16}$$

$$\mathbf{W}^{-1}_{t|t,T} = \mathbf{W}^{-1}_{t|t-1,T} + \sum_{c=1}^{C}\boldsymbol{\alpha}_c\boldsymbol{\alpha}_c'\lambda_c(t|\mathbf{x}_{t|t-1,T})\Delta \tag{17}$$

$$\mathbf{x}_{t|t,T} = \mathbf{x}_{t|t-1,T} + \mathbf{W}_{t|t,T}\sum_{c=1}^{C}\boldsymbol{\alpha}_c(N^c_t - \lambda_c(t|\mathbf{x}_{t|t-1,T})\Delta) \tag{18}$$

To provide some insight into these recursions, note that in the prediction step given in (15) the feedback-controlled prior model is used to move the estimate forward. In the update step given in (18) the estimate is found by making a correction or update to this prediction. Here, $\lambda_c(t|\mathbf{x}_{t|t-1,T})\Delta$ is the predicted probability of having a spike in the time interval Δ and hence the correction is (1- predicted probability of a spike) if a spike occurs and (0- predicted probability of a spike) if no spike occurs. Hence if a spike occurs and the predicted probability of a spike is high this correction is small and vice versa. Therefore the estimate is a combination of the prediction and the correction terms.

Resolving the Duration Uncertainty: Feedback-Controlled Parallel Point Process Filter (FC-P-PPF)

The feedback-controlled state-space model in (6) (and many other goal-directed state-space models) is a function of movement duration, T, not known a priori to the real-time BMI [43, 55]. Hence any goal-directed real-time decoder needs to resolve this duration uncertainty. We introduced a framework to resolve this duration uncertainty in [43] by discretizing the movement duration, finding the kinematic estimate for each discretized duration, and then optimally combining these kinematic estimates based

on the neural data. A similar approach using a discretized set of durations was subsequently used in a simulation study in [56] to resolve the duration uncertainty of the prior model developed in [40] for estimation of simulated trajectories. Our framework for resolving the duration uncertainty is based on mixture modeling, a common approach in statistical inference that is used to estimate a desired density in different applications. For example, mixture modeling combined with sequential state estimation in dynamical systems, when the system is operating under different or changing regimes of operation, has been used as early as in [57]. See also the mixture Kalman filtering work in [58] and references therein. For decoding the kinematics from neural activity, mixture modeling was first used in [44] and successfully applied to combine empirically fitted and time-invariant state models for reaching movements to different targets in an offline study. Here, we use mixture modeling to combine feedback-controlled prior models of different durations and hence resolve the duration uncertainty inherent to this prior model. Our framework works by discretizing the duration and finding the likelihood of each of the discretized duration points jointly with the corresponding trajectory estimate, all purely based on the neural spiking activity in real time. It then optimally combines the trajectory estimates corresponding to the discretized durations to get the overall trajectory estimate. The result is the feedback-controlled parallel point process filter (FC-P-PPF) . Denoting the overall MMSE estimator by $\mathbf{x}_{t|t}$, it is given by the mean of the posterior density, which using the law of total probability is expanded as,

$$\mathbf{x}_{t|t} = E(\mathbf{x}_t|\mathbf{N}_{1:t}) = \sum_{j=1}^{J} p(T_j|\mathbf{N}_{1:t})\mathbf{x}_{t|t,T_j} \tag{19}$$

where $T_1, \cdots, T_J$ are the J discretization points for T, $\mathbf{x}_{t|t,T_j}$ is the estimate given that the duration is T_j and found from the recursions in (15) to (18), $p(T_j|\mathbf{N}_{1:t})$ is the likelihood of the corresponding duration given the peri-movement neural activity, and finally the summation is over all j for which $T_j > t$. The likelihood of the corresponding duration, $p(T_j|\mathbf{N}_{1:t})$, can be computed as derived in our work in [43, 55] and is only a function of the prediction and posterior means and covariances found in (15) to (18) and the parameters of the observation model. Here we provide the final expressions for readers' convenience. These are given by

$$p(T_j|\mathbf{N}_{1:t}) \propto p(\mathbf{N}_{1:t}|T_j) \tag{20}$$

with

$$p(\mathbf{N}_{1:t}|T_j) = \prod_{i=1}^{t} p(\mathbf{N}_i|\mathbf{N}_{1:i-1}, T_j) = \prod_{i=1}^{t} g(\mathbf{N}_i|T_j) \tag{21}$$

Using a Gaussian approximation to the posterior, the term $g(\mathbf{N}_i|T_j)$ as we derive in [43, 55] is given by

$$\begin{aligned}
g(\mathbf{N}_i|T_j) &= \sqrt{\frac{|\mathbf{W}_{i|i,T_j}|}{|\mathbf{W}_{i|i-1,T_j}|}}\, p(\mathbf{N}_i|\mathbf{x}_{i|i,T_j}, \mathbf{N}_{1:i-1})\times \\
&\exp\left[-\frac{1}{2}(\mathbf{x}_{i|i,T_j} - \mathbf{x}_{i|i-1,T_j})'\mathbf{W}^{-1}_{i|i-1,T_j}(\mathbf{x}_{i|i,T_j} - \mathbf{x}_{i|i-1,T_j})\right] \\
&= \sqrt{\frac{|\mathbf{W}_{i|i,T_j}|}{|\mathbf{W}_{i|i-1,T_j}|}} \prod_c \left(\lambda_c(i|\mathbf{x}_{i|i,T_j})\Delta\right)^{N_i^c} e^{-\lambda_c(i|\mathbf{x}_{i|i,T_j})\Delta} \\
&\times \exp\left[-\frac{1}{2}(\mathbf{x}_{i|i,T_j} - \mathbf{x}_{i|i-1,T_j})'\mathbf{W}^{-1}_{i|i-1,T_j}(\mathbf{x}_{i|i,T_j} - \mathbf{x}_{i|i-1,T_j})\right]
\end{aligned} \tag{22}$$

for $j = 1, \cdots, J$ where all the quantities are known. Note that $\sum_{j=1}^{J} p(T_j|\mathbf{N}_{1:t}) = 1$ and hence we can compute all the duration likelihoods. Finally, combining in (19), we get the final estimate. Here, we coarsely discretized the duration of a trial.

Random-Walk Point Process Filter (RW-PPF)

Setting $\mathbf{B} = \mathbf{0}$ reduces the state-space model in (3) to the random-walk model and consequently the recursions of FC-PPF with this choice recover the RW-PPF.

Possible Extensions

In our implementation of the BMI, the first stage of the decoder makes a decision about the intended target. However, instead, the BMI can easily include all targets in the FC-P-PPF but weight them properly by their corresponding likelihood calculated from the first stage similar to a previous offline study [44]. In our case, this means including more parallel filters in the FC-P-PPF for the different targets (and their discretized durations). This will consequently increase the complexity of the decoder. We chose not to implement this extension since, using offline analysis, we observed that it resulted in little improvement at the price of four times the complexity. The absence of a significant improvement in this case was likely a result of the fact that in our experiments the target-related activity during the delay period was strongly tuned to the targets and that overall the peri-movement activity was not as strongly tuned to the task. However, in cases where such target-related activity is not strong, this extension will potentially result in further improvement in the second stage as it allows for its higher weighting compared to the first stage.

Number of Neurons Required for Accurate Target Prediction

To find the number of neurons that were sufficient to obtain an accurate target prediction during the delay period, we performed a single neuron analysis in which the spiking activity of a single neuron was used to decode the target (Figure S1). We then sorted the neurons based on their single-neuron accuracies. From the sorted set, we selected different number of neurons and performed the decoding analysis for them. For example the decoding analysis for two neurons was done for the two neurons with the highest single neuron accuracies. Doing so, we found the target prediction accuracy as a function of the number of top cells included in the decoding. We found that on average across sessions only 17 $\pm$ 6% of the neural ensemble or 3.3 $\pm$ 1.0 neurons were sufficient to obtain a prediction accuracy higher than 90% of the ensemble accuracy.

Roughness Coefficient

The roughness coefficient for a sequence is an indicator of how smooth it is [49]. For a sequence $d_1, \cdots, d_T$, it is defined as

$$r = \frac{\sum_{t=2}^{T}(d_t - d_{t-1})^2}{\sum_{t=2}^{T}(d_t - \bar{d})^2}$$

where $\bar{d}$ denotes the mean of the sequence. It can also be generalized to the vector case by writing

$$r = \frac{\sum_{t=2}^{T} \parallel \mathbf{d}_t - \mathbf{d}_{t-1} \parallel^2}{\sum_{t=2}^{T} \parallel \mathbf{d}_t - \bar{\mathbf{d}} \parallel^2}$$

For a trajectory of duration T this provides a measure of smoothness where smaller coefficients correspond to smoother trajectories. In this case $\mathbf{d}_t$ corresponds to the position vector at time t .

Acknowledgments

References

1. Chapin JK, Moxon KA, Markowitz RS, Nicolelis MAL (1999) Real-time control of a robot arm using simultaneously recorded neurons in the motor cortex. Nat Neurosci 2: 664-670.

2. Wessberg J, Stambaugh CR, Kralik JD, Beck PD, Laubach M, et al. (2000) Real-time prediction of hand trajectory by ensembles of cortical neurons in primates. Nature 408: 361-365.

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

4. Hochberg LR, Serruya MD, Friehs GM, Mukand JA, Saleh M, et al. (2006) Neuronal ensemble control of prosthetic devices by a human with tetraplegia. Nature 442: 164-171.

5. Carmena JM, Lebedev MA, Crist RE, O'Doherty JE, Santucci DM, et al. (2003) Learning to control a brain-machine interface for reaching and grasping by primates. PLoS Biol 1: 193-208.

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

7. Ganguly K, Carmena JM (2009) Emergence of a stable cortical map for neuroprosthetic control. PLoS Biol 7: 1-13.

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

9. Velliste M, Perel S, Spalding MC, Whitford AS, Schwartz AB (2008) Cortical control of a prosthetic arm for self-feeding. Nature 453: 1098-1101.

10. Moritz CT, Perlmutter SI, Fetz EE (2008) Direct control of paralysed muscles by cortical neurons. Nature 456: 639-643.

11. Mulliken GH, Musallam S, Andersen RA (2008) Decoding trajectories from posterior parietal cortex ensembles. J Neurosci 28: 12913-12926.

12. Kim SP, Simeral JD, Hochberg LR, Donoghue JP, Black MJ (2008) Neural control of computer cursor velocity by decoding motor cortical spiking activity in humans with tetraplegia. J Neural Eng 5: 455-476.

13. Li Z, O'Doherty JE, Hanson TL, Lebedev MA, Henriquez CS, et al. (2009) Unscented Kalman filter for brain-machine interfaces. PLoS ONE 4: 1-18.

14. Chase SM, Schwartz AB, Kass RE (2009) Bias, optimal linear estimation, and the differences between open-loop simulation and closed-loop performance of spiking-based brain-computer interface algorithms. Neural Networks 22: 1203-1213.

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

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

17. Georgopoulos A, Kalaska JF, Caminiti R, Massey JT (1982) On the relations between the direction of two-dimensional arm movement and cell discharge in primate motor cortex. J Neurosci 2: 1527-1537.

18. Georgopoulos A, Schwartz AB, Kettner RE (1986) Neural population coding of movement direction. Science 233: 1416-1419.

19. Ashe J, Georgopoulos AP (1994) Movement parameters and neural activity in motor cortex and area 5. Cereb Cortex 6: 590-600.

20. Moran DW, Schwartz AB (1999) Motor cortical representation of speed and direction during reaching. J Neurophysiol 82: 2676-2692.

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

22. Crammond DJ, Kalaska JF (1996) Differential relation of discharge in primary motor cortex and premotor cortex to movements versus actively maintained postures during a reaching task. Exp Brain Res 108: 45-61.

23. Caminiti R, Johnson PB, Galli C, Ferraina S, Burnod Y (1991) Making arm movements within different parts of space: The premotor and motor cortical representation of a coordinate system for reaching to visual targets. J Neurosci 11: 1182-1197.

24. Kurata K (1993) Premotor cortex of monkeys: Set- and movement-related activity reflecting amplitude and direction of wrist movements. J Neurophysiol 69: 187-200.

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

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

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

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

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

30. Andersen RA, Burdick JW, Musallam S, Pesaran B, Cham JG (2004) Cognitive neural prosthetics. Trends in Cognitive Sciences 8: 486-493.

31. Messier J, Kalaska JF (2000) Covariation of primate dorsal premotor cell activity with direction and amplitude during a memorized-delay reaching task. J Neurophysiol 84: 152-165.

32. Crammond DJ, Kalaska JF (1994) Modulation of preparatory neuronal activity in dorsal premotor cortex due to stimulus-response compatibility. J Neurophysiol 71: 1281-1284.

33. Boussaoud D, Bremmer F (1999) Gaze effects in the cerebral cortex: reference frames for space coding and action. Exp Brain Res 128: 170-180.

34. Kakei S, Hoffman DS, Strick PL (2003) Sensorimotor transformations in cortical motor areas. Neurosci Res 46: 1-10.

35. Churchland MM, Santhanam G, Shenoy KV (2006) Preparatory activity in premotor and motor cortex reflects the speed of the upcoming reach. J Neurophysiol 96: 3130-3146.

36. Crammond DJ, Kalaska JF (2000) Prior information in motor and premotor cortex: Activity during the delay period and effect on pre-movement activity. J Neurophysiol 84: 986-1005.

37. Crutcher MD, Russo GS, Ye S, Backus DA (2004) Target-, limb-, and context-dependent neural activity in the cingulate and supplementary motor areas of the monkey. Exp Brain Res 158: 278-288.

38. Todorov E, Jordan MI (2002) Optimal feedback control as a theory of motor coordination. Nat Neurosci 5: 1226-1235.

39. Todorov E (2004) Optimality principles in sensorimotor control. Nat Neurosci : 907-915.

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

41. Kemere C, Meng T (2005) Optimal estimation of feed-forward-controlled linear systems. In: Proc. ICASSP. pp. 353-356.

42. Srinivasan L, Eden UT, Mitter SK, Brown EN (2007) General-purpose filter design for neural prosthetic devices. J Neurophysiol 98: 2456-2475.

43. Shanechi MM, Wornell GW, Williams ZM, Brown EN (2010) A parallel point-process filter for estimation of goal-directed movements from neural signals. In: Proc. ICASSP.

44. Yu BM, Kemere C, Santhanam G, Afshar A, Ryu SI, et al. (2007) Mixture of trajectory models for neural decoding of goal-directed movements. J Neurophysiol 97: 3763-3780.

45. Loeb GE, Levine WS, He J (1990) Understanding sensorimotor feedback through optimal control. Cold Spring Harb Symp Quant Biol 55: 791-803.

46. Todorov E (2005) Stochastic optimal control and estimation methods adapted to the noise characteristics of the sensorimotor system. Neural Comput 17: 1084-1108.

47. Liu D, Todorov E (2007) Evidence for the flexible sensorimotor strategies predicted by optimal feedback control. J Neurosci : 9354-9368.

48. Truccolo W, Eden UT, Fellows MR, Donoghue JP, Brown EN (2005) A point process framework for relating neural spiking activity to spiking history, neural ensemble, and extrinsic covariate effects. J Neurophysiol 93: 1074-1089.

49. Bloomfield P (2000) Fourier analysis of time series. Wiley.

50. Scott SH (2004) Optimal feedback control and the neural basis of volitional motor control. Nat Rev Neurosci 5: 532-546.

51. Cunningham JP, Nuyujukian P, Gilja V, Chestek CA, Ryu SI, et al. (2011) A closed-loop human simulator for investigating the role of feedback control in brain-machine interfaces. J Neurophysiol 105: 1932-1949.

52. Kulkarni JE, Paninski L (2008) State-space decoding of goal-directed movements. IEEE Signal Process Mag 25: 78-86.

53. Bertsekas D (2005) Dynamic Programming and Optimal Control. Athena Scientific.

54. Eden UT, Frank LM, Barbieri R, Solo V, Brown EN (2004) Dynamic analysis of neural encoding by point process adaptive filtering. Neural Comput 16: 971-998.

55. Shanechi MM (2011) Real-Time Brain-Machine Interface Architectures: Neural Decoding from Plan to Movement. Ph.D. thesis, Massachusetts Institute of Technology.

56. Srinivasan L, da Silva M (2011) Breaking the fixed-arrival-time restriction in reaching movements of neural prosthetic devices. IEEE Trans Biomed Eng 58: 1555-1564.

57. Ackerson GA, Fu KS (1970) On state estimation in switching environments. IEEE Trans Autom Control AC-15: 10-17.

58. Chen R, Liu JS (2000) Mixture Kalman filters. J Roy Statist Soc B 62: 493-508.

Figure Legends

Figure 1. Experimental task and the optimal feedback control framework. (A) Experimental task. The experiment consisted of an instructed delay center-out task with four targets. To be rewarded, the monkey had to acquire the correct target without touching any of the incorrect ones first. After target presentation, there was 1 s of delay before the "go" cue, signaling that the monkey could begin moving the joystick. During the training sessions the monkey controlled the position of the cursor using a joystick. During the BMI sessions the joystick was disconnected and the real-time decoder controlled the cursor. (B) Optimal feedback control framework. An optimal feedback control framework is used to model the task. In this framework each task is performed to accomplish a goal during which there is real-time sensory feedback (for example vision and proprioception), $\mathbf{y}_t$, about the state of the system, $\mathbf{x}_t$. Based on the intended goal and the sensory feedback about the state of the system, the brain (controller) decides on a plan of action, which is reflected in its neural activity, $\mathbf{N}_t$, and controls the system. In natural movement, this is the animal's own musculo-skeletal system and in movement using a BMI, this is the BMI.

Figure 2. Offline model comparisons. The bars show mean quantities and the error bars show the standard deviation around the mean across sessions. All quantities are obtained from the training sessions using leave-one-out cross-validation. (A) Accuracy of the different models. (B) RMS error of the different models. (C) Roughness coefficient of the different models. The two-stage decoder (used in the real-time BMI) outperforms all other models in terms of accuracy, RMS error, and smoothness.

Figure 3. Comparison of the offline trajectory estimates. The green circle shows the instructed target and the yellow line shows the monkey's trajectory. The black line shows the trajectory estimate using the two-stage decoder, the red circle shows the predicted target from the first stage, the blue line shows the trajectory estimate of RW-PPF (i.e., the second stage of the decoder without using the target prediction), and the red line shows that of the linear regression decoder. (A) Sample trials in which both the two-stage decoder and RW-PPF acquire the target correctly. (B) Sample trials in which the two-stage decoder acquires the target correctly but RW-PPF does not. (C) Sample trials in which the two-stage decoder acquires the target correctly but the target is predicted incorrectly from the first stage.

Figure 4. Real-time trajectories and the complementary property of the two-stage BMI. (A) Comparison of the real-time BMI accuracy with the real-time target prediction accuracy from the first stage and also with the accuracy of RW-PPF (i.e., the second stage without using the target prediction) and the linear regression decoder obtained using the same real-time data set. The bars show mean quantities and the error bars show the standard deviation around the mean across sessions. The two-stage

BMI outperforms all other decoders. (B–C) Sample decoded trajectories. In all panels, the green circle shows the instructed target, the red circle shows the predicted target from the first stage, the black line shows the trajectory estimate of the two-stage BMI, and the blue line shows that of RW-PPF using only the peri-movement activity. Sample trials where both the two-stage BMI and RW-PPF correctly acquire the target are shown in B. The complementary property of the two-stage BMI is illustrated in C and D. C shows sample trials in which the correct target prediction of the first stage compensates for the inaccurate estimation of the kinematic decoder in the second stage. D shows sample trials in which the kinematic decoder in the second stage, using ongoing peri-movement activity, compensates for the incorrect target prediction of the first stage.

Figure 5. Comparison of the trajectory estimates. The black line shows the trajectory estimate of the real-time BMI. The blue line shows the trajectory estimate of RW-PPF using only the peri-movement activity and the red line shows that of a linear regression decoder, both using the same real-time data set as the BMI. (A) Sample trials in which the linear regression decoder is correct. (B) Sample trials in which the linear regression decoder is incorrect. Even when both the two-stage and the regression decoders acquire the correct target, the trajectory estimates of the two-stage BMI are smoother than the linear regression decoder.

Figure S1. Neuron dropping analysis. (A–D) Activity of a single neuron under the four targets in one training session. In the top figure, each row corresponds to a different trial and the black dots indicate the spike times. The bottom figure shows the corresponding firing rate. Activity is aligned to the target presentation time and vertical dashed lines indicate the target presentation and "go" cue times. The target prediction accuracy (leave-one-out cross-validation) of this neuron is 65%. (E) Target prediction accuracy as a function of the number of top neurons. The solid lines show the target prediction accuracy of six training sessions as a function of the number of neurons included in the prediction. Neurons were sorted based on their single neuron accuracy. Chance level accuracy is 25%.

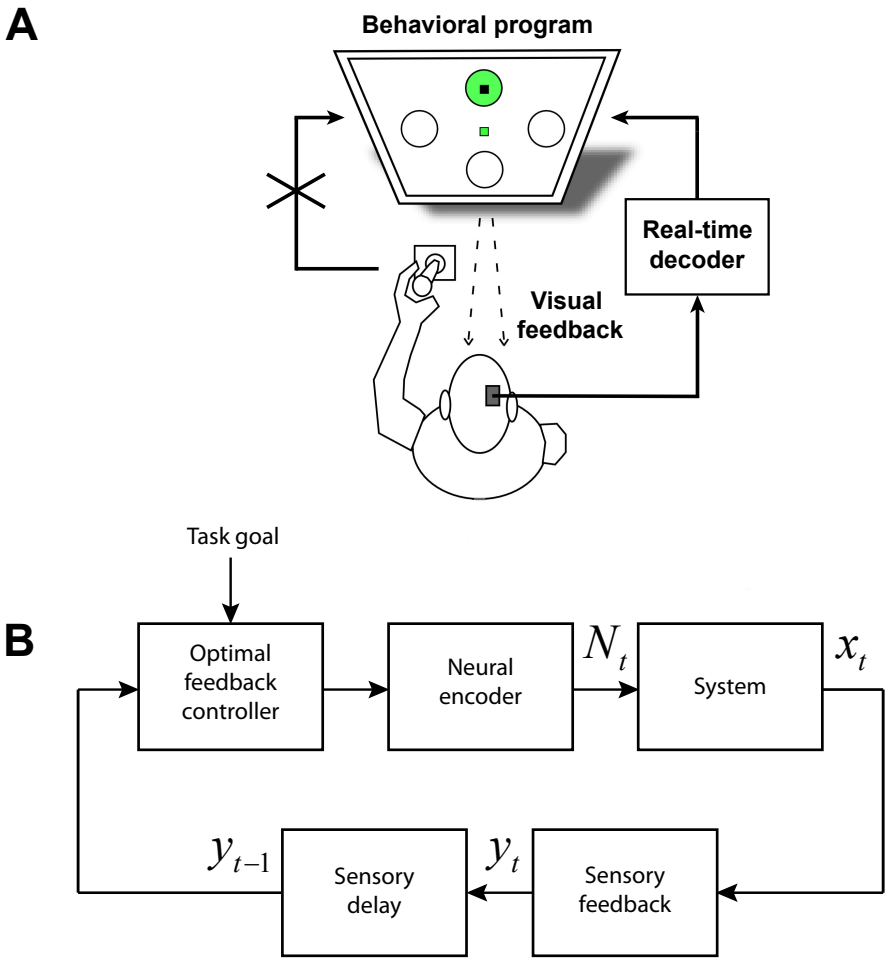

A
Behavioral program
Real-time decoder
Visual feedback
B
Task goal
Optimal feedback controller
Neural encoder
N_t
System
x_t
y_{t-1}
Sensory delay
y_t
Sensory feedback

Figure 1

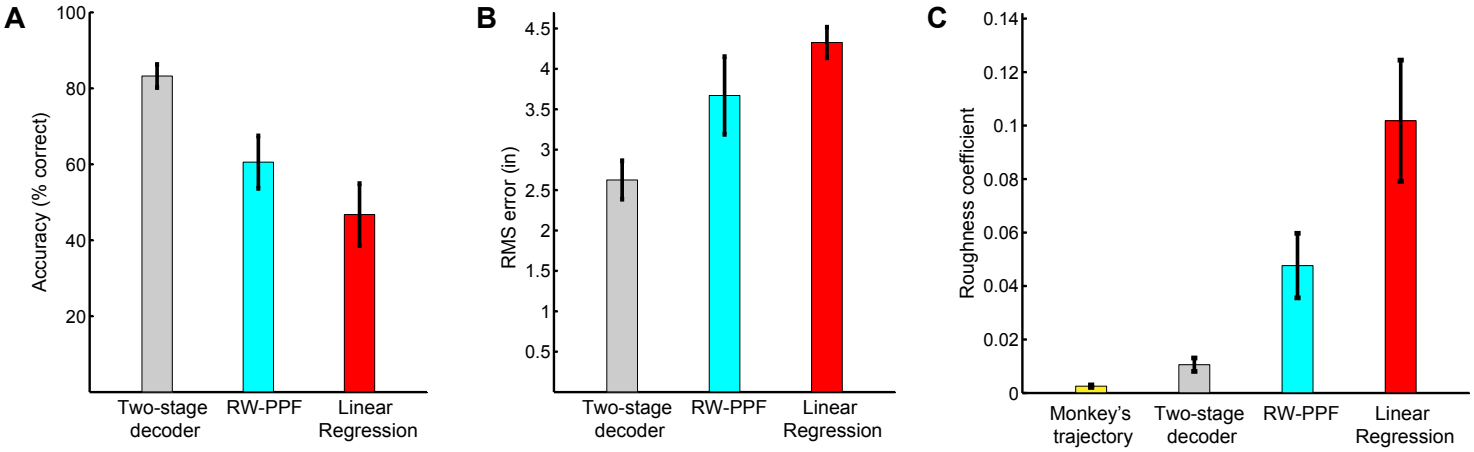

A
Accuracy (% correct)
100
80
60
40
20
Two-stage decoder
RW-PPF
Linear Regression
B
RMS error (in)
4.5
4
3.5
3
2.5
2
1.5
1
0.5
Two-stage decoder
RW-PPF
Linear Regression
C
Roughness coefficient
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
Monkey's trajectory
Two-stage decoder
RW-PPF
Linear Regression

Figure 2

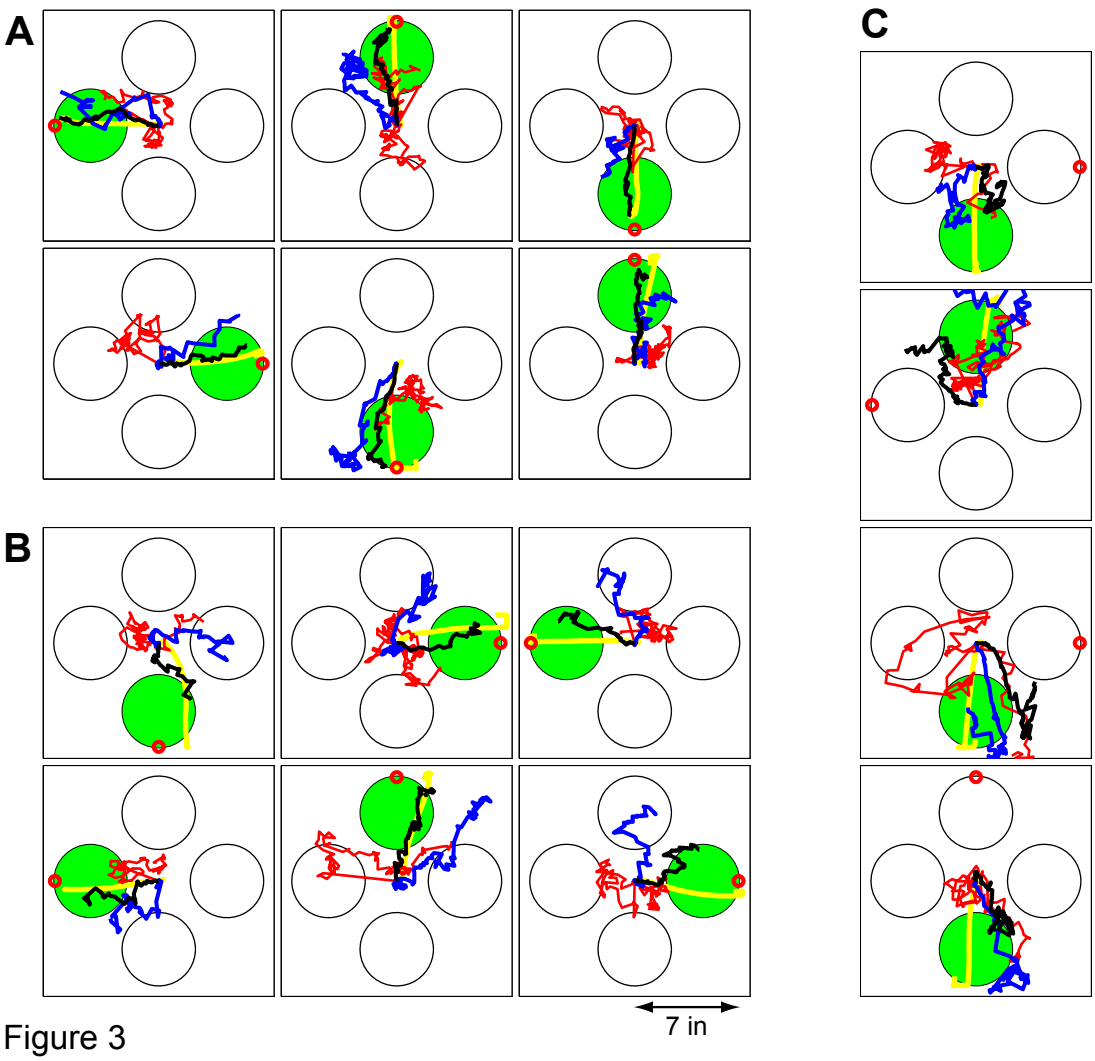


Figure 3

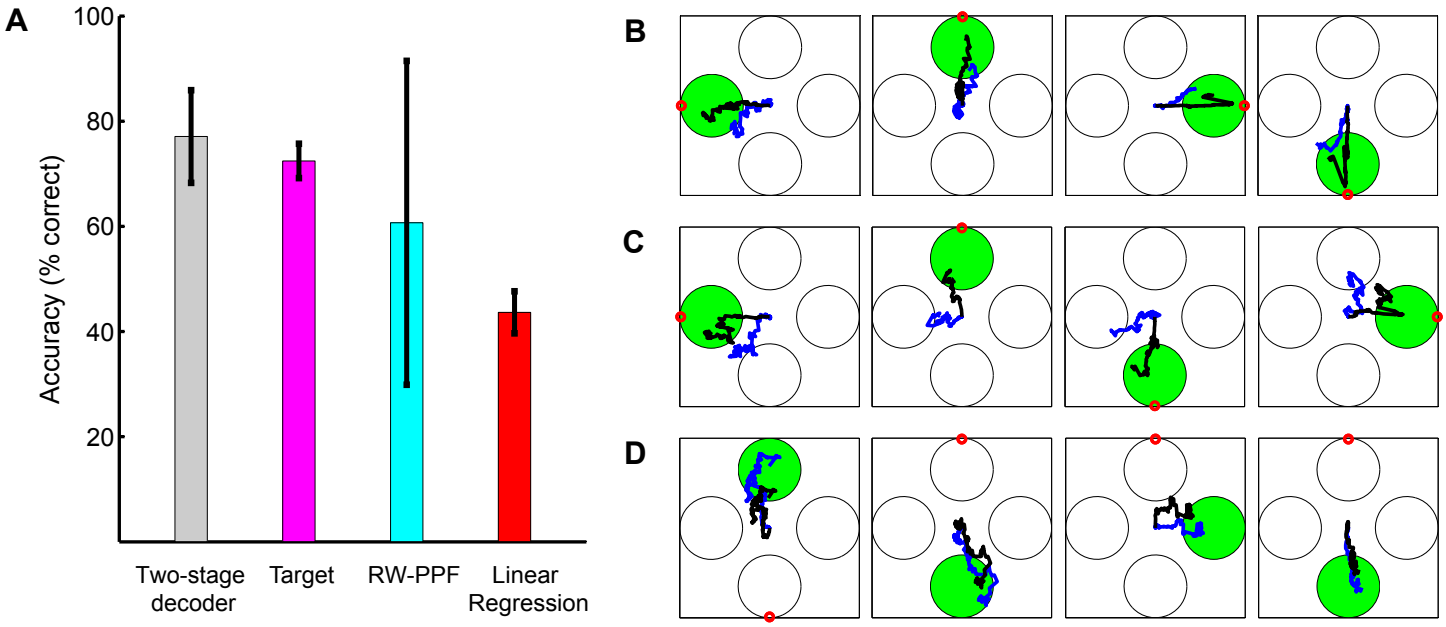

A
100
80
60
40
20
Accuracy (% correct)
Two-stage decoder
Target
RW-PPF
Linear Regression
B
C
D

Figure 4

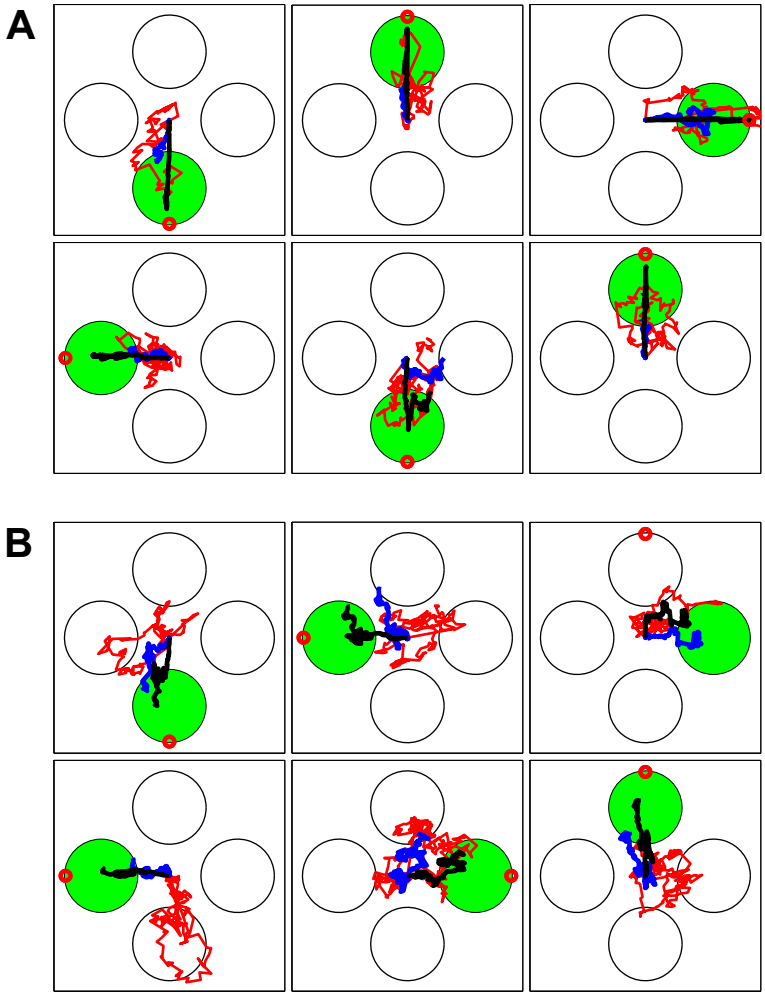


Figure 5

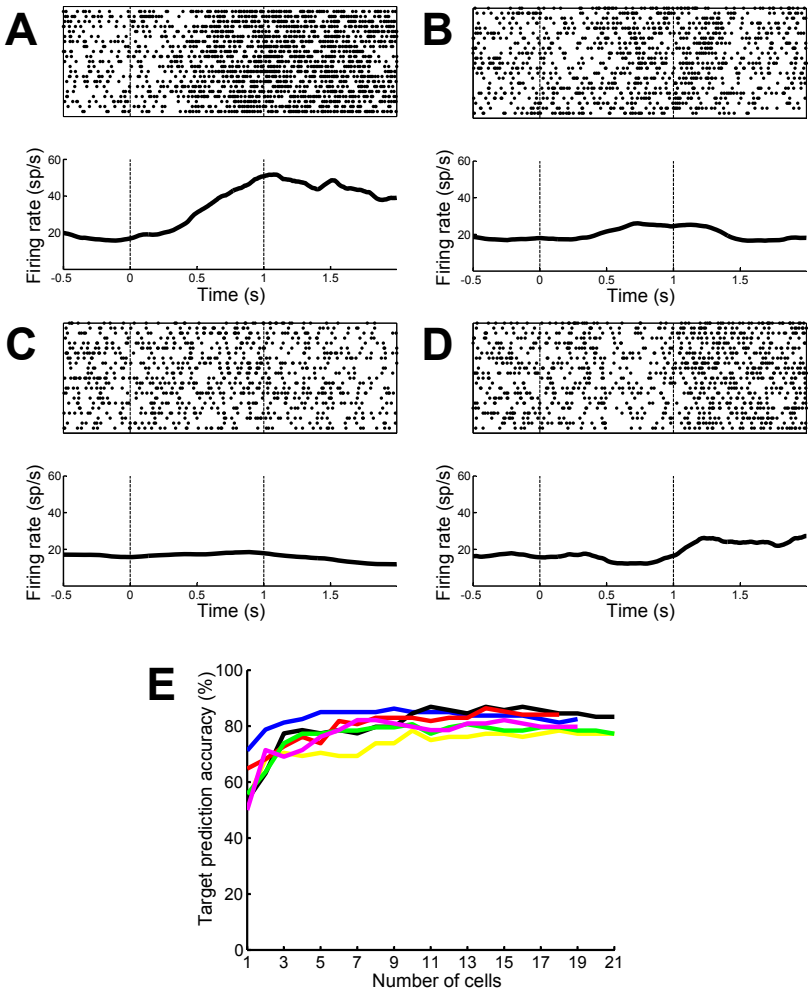

A
B
C
D
E
Firing rate (sp/s)
Time (s)
Target prediction accuracy (%)
Number of cells

Figure S1