A closed-loop human simulator for investigating the role of feedback control in brain-machine interfaces

John P. Cunningham, Paul Nuyujukian, Vikash Gilja, Cindy A. Chestek, Stephen I. Ryu and Krishna V. Shenoy

J Neurophysiol 105:1932-1949, 2011. First published 13 October 2010; doi:10.1152/jn.00503.2010

J Neurophysiol 105: 1932–1949, 2011.
First published October 13, 2010; doi:10.1152/jn.00503.2010.

A closed-loop human simulator for investigating the role of feedback control in brain-machine interfaces

John P. Cunningham,[1,2] Paul Nuyujukian,[3,4] Vikash Gilja,[5] Cindy A. Chestek,[1] Stephen I. Ryu,[1,7] and Krishna V. Shenoy[1,3,6]

[1]*Department of Electrical Engineering, Stanford University, Stanford, California;* [2]*Department of Engineering, University of Cambridge, Cambridge, United Kingdom;* [3]*Department of Bioengineering,* [4]*School of Medicine,* [5]*Department of Computer Science,* [6]*Neurosciences Program, Stanford University, Stanford, California;* [7]*Palo Alto Medical Foundation, Palo Alto, California*

Submitted 4 June 2010; accepted in final form 7 October 2010

Cunningham JP, Nuyujukian P, Gilja V, Chestek CA, Ryu SI, Shenoy KV. A closed-loop human simulator for investigating the role of feedback control in brain-machine interfaces. *J Neurophysiol* 105: 1932–1949, 2011. First published October 13, 2010; doi:10.1152/jn.00503.2010.—Neural prosthetic systems seek to improve the lives of severely disabled people by decoding neural activity into useful behavioral commands. These systems and their decoding algorithms are typically developed "offline," using neural activity previously gathered from a healthy animal, and the decoded movement is then compared with the true movement that accompanied the recorded neural activity. However, this offline design and testing may neglect important features of a real prosthesis, most notably the critical role of feedback control, which enables the user to adjust neural activity while using the prosthesis. We hypothesize that understanding and optimally designing high-performance decoders require an experimental platform where humans are in closed-loop with the various candidate decode systems and algorithms. It remains unexplored the extent to which the subject can, for a particular decode system, algorithm, or parameter, engage feedback and other strategies to improve decode performance. Closed-loop testing may suggest different choices than offline analyses. Here we ask if a healthy human subject, using a closed-loop neural prosthesis driven by synthetic neural activity, can inform system design. We use this online prosthesis simulator (OPS) to optimize "online" decode performance based on a key parameter of a current state-of-the-art decode algorithm, the bin width of a Kalman filter. First, we show that offline and online analyses indeed suggest different parameter choices. Previous literature and our offline analyses agree that neural activity should be analyzed in bins of 100- to 300-ms width. OPS analysis, which incorporates feedback control, suggests that much shorter bin widths (25–50 ms) yield higher decode performance. Second, we confirm this surprising finding using a closed-loop rhesus monkey prosthetic system. These findings illustrate the type of discovery made possible by the OPS, and so we hypothesize that this novel testing approach will help in the design of prosthetic systems that will translate well to human patients.

neural prostheses; brain-computer interfaces

DEBILITATING CONDITIONS LIKE spinal cord injuries can leave a human without voluntary motor control. However, in many cases, the brain itself maintains normal function. Millions of people worldwide suffer motor deficits due to these diseases and injuries that result in a significantly diminished ability to interact with the physical world. Indeed, tetrapalegic humans list regaining "arm/hand function" as the top priority for improving their quality of life, as restoring this function would allow significant independence (Anderson 2004). To address this huge medical need, brain-machine interfaces (BMI, also called neural prosthetic systems or brain-computer interfaces) seek to access the information in the brain and use that information to control a prosthetic device such as a robotic arm or a computer cursor. Such systems, if successful, would have a large quality of life impact for many people living with these debilitating medical conditions.

In the last decade, advances in neural recording technologies have accelerated research in neural prostheses. Technologies for neural recording include electroencephalography (EEG), electrocorticography (ECoG), and penetrating electrode or microwire arrays [see Lebedev and Nicolelis (2006) for a review]. To design a prosthetic arm that can be controlled continuously with high precision, most work has focused on penetrating electrodes implanted directly into motor cortical areas (Schwartz 2004; Velliste et al. 2008). Researchers use nonhuman primates (e.g., rhesus monkeys) or, increasingly, human participants (Hochberg et al. 2006; Kim et al. 2008). There are many medical, scientific, and engineering challenges in developing such a system (Lebedev and Nicolelis 2006; Ryu and Shenoy 2009; Schwartz 2004; Schwartz et al. 2006), but all neural prostheses share in common the need for a decode algorithm. Decode algorithms map neural activity into physical commands such as kinematic parameters to control a robotic arm.

Much work has gone into this domain, and many experimental paradigms and decoding approaches have been developed and used (Artemiadis et al. 2007; Brockwell et al. 2004; Brown et al. 1998; Carmena et al. 2003, 2005; Chase et al. 2009; Chestek et al. 2007; Eden et al. 2004; Ganguly and Carmena 2009; Gao et al. 2002; Georgopoulos et al. 1986; Hatsopoulos et al. 2004; Hochberg et al. 2006; Kemere et al. 2004; Kim et al. 2006, 2008; Koyama et al. 2010; Kulkarni and Paninski 2008; Lebedev et al. 2005; Li et al. 2009; Moritz et al. 2008; Mulliken et al. 2008; Musallam et al. 2004; Paninski et al. 2004; Sanchez et al. 2008; Santhanam et al. 2006; Serruya et al. 2002; Shakhnarovich et al. 2006; Shenoy et al. 2003; Shoham et al. 2005; Srinivasan and Brown 2007; Srinivasan et al. 2007, 2006; Taylor et al. 2002; Velliste et al. 2008; Ventura 2008; Wessberg et al. 2000; Wu et al. 2004, 2006; Wu and Hatsopoulos 2008; Yu et al. 2007). Most research in decode algorithms has been done with offline data analysis using

Address for reprint requests and other correspondence: K. V. Shenoy, 319 Allen Building, 330 Serra Mall, Stanford University, Stanford, CA 94305-4075 (e-mail: shenoy@stanford.edu).

simulated neural data (Brockwell et al. 2004; Kemere et al. 2004; Srinivasan and Brown 2007; Srinivasan et al. 2006, 2007; Ventura 2008) and/or neural data that were previously recorded from a healthy animal (Artemiadis et al. 2007; Brockwell et al. 2004; Brown et al. 1998; Carmena et al. 2005; Chestek et al. 2007; Eden et al. 2004; Gao et al. 2002; Georgopoulos et al. 1986; Hatsopoulos et al. 2004; Kim et al. 2006; Lebedev et al. 2005; Mulliken et al. 2008; Paninski et al. 2004; Santhanam et al. 2006; Serruya et al. 2002; Shakhnarovich et al. 2006; Shenoy et al. 2003; Shoham et al. 2005; Taylor et al. 2002; Wessberg et al. 2000; Wu et al. 2004, 2006; Wu and Hatsopoulos 2008; Yu et al. 2007). In these studies, the benchmark for success is often how well the decoded arm trajectory matches the true arm movement that was recorded in conjunction with the (possibly simulated) neural activity. A smaller number of studies have used an online, closed-loop paradigm to illustrate that prostheses can be meaningfully controlled by humans or monkeys (Carmena et al. 2003; Chase et al. 2009; Ganguly and Carmena 2009; Hochberg et al. 2006; Kim et al. 2008; Koyama et al. 2010; Li et al. 2009; Moritz et al. 2008; Mulliken et al. 2008; Musallam et al. 2004; Santhanam et al. 2006; Serruya et al. 2002; Taylor et al. 2002; Velliste et al. 2008), but only a few of these studies (Chase et al. 2009; Koyama et al. 2009; Li et al. 2009) compare closed-loop performance of different algorithms in monkeys, and only one of these studies (Kim et al. 2008) compares the closed-loop performance of two different decode algorithms in humans. This reality is at least in part driven by the substantial resources and effort required for online studies in animals and humans, thereby prohibiting extensive online algorithmic comparisons.

Despite this abundance of work, our ability to decode arm movements accurately remains limited. To decode an arbitrary reach, one current state-of-the-art algorithm is perhaps the Kalman filter [introduced nearly 50 yr ago in Kalman (1960), used in this context in Kim et al. (2008); Wu et al. (2006)], which is the only algorithm that has been vetted in online human experiments as having better performance than some competing possibilities such as a linear filter (Kim et al. 2008) [although the closed-loop monkey studies of Chase et al. (2009); Koyama et al. (2010); Li et al. (2009) indicate that other algorithms are also competitive]. Current achievable performance is encouraging and we have exciting proofs of concept (Hochberg et al. 2006; Santhanam et al. 2006; Velliste et al. 2008), but we must advance considerably before these systems are clinically viable and further still before we achieve decoded movements with speed and accuracy comparable to a healthy arm (e.g., near perfect, subsecond accuracy).

In response to this critical need, several groups have proposed more advanced mathematical approaches to neural prosthetic decoding (Artemiadis et al. 2007; Brockwell et al. 2004; Brown et al. 1998; Eden et al. 2004; Gao et al. 2002; Kemere et al. 2004; Kim et al. 2006; Kulkarni and Paninski 2008; Li et al. 2009; Mulliken et al. 2008; Paninski et al. 2004; Sanchez et al. 2008; Shakhnarovich et al. 2006; Shoham et al. 2005; Srinivasan and Brown 2007; Srinivasan et al. 2007, 2006; Ventura 2008; Wu and Hatsopoulos 2008; Yu et al. 2007). However, none of these methods has seen widespread adoption (across research studies or in critical translational work), in part due to the uncertainty of how these methods translate to closed-loop decode performance.

Offline evaluation of algorithms may neglect potentially important features of a real neural prosthesis, including the user's ability to modify control strategies to improve prosthetic performance. Truly understanding decode performance requires the human learning machine (the brain and motor plant) to be in closed-loop with the decode algorithm. In this online, closed-loop setting, as soon as a prosthesis user sees a decoded arm reach (the action of a robotic arm or the path of a cursor on a computer screen), he/she will bring to bear all of his/her modification strategies to drive a desirable reach.

As a specific example of offline vs. online evaluation (in anticipation of the experiments done here), a previous offline study found that prosthetic decode error is minimized when the time bin over which neural activity is integrated (a windowed spike count in the Kalman filter) is 200–300 ms (Wu et al. 2006). This bin width also represents the time step at which the algorithm updates its estimate of the decoded reach. However, it may be that in a closed-loop experiment, when reaches last only roughly 1,000 ms, the intermittent "hopping" behavior of a decoded reach will frustrate the user. Perhaps better control could be gained with a more frequent update, where feedback control would compensate for the increased noise in the decode. Indeed, in Kim et al. (2008), shorter bin widths (50 and 100 ms) were used in online human experiments. It would appear that shorter bin widths were found to be better in initial online testing, although perhaps not optimized online, which may be in part due to the overall difficulty/challenges of testing with disabled human participants. Thus it remains unclear how this and other parameters should be set in future studies. This simple question, motivated in part by the work of Kim et al. (2008) and Wu et al. (2006), can be answered with current algorithmic technologies, but it requires closed-loop validation. The field should investigate the extent to which the subject can, for a given decode algorithm, engage online control strategies to improve decode performance. Closed-loop testing may suggest different priorities for algorithmic development than offline analyses.

Addressing this problem is highly challenging, since fully doing so would imply validating every algorithmic choice, ideally, in a human clinical trial. Algorithmic choices include both the structure of the algorithm itself and the parameter settings that should be optimized, resulting in thousands of decode possibilities. Given the invasiveness and resource requirements of a full neural prosthetic clinical trial, this approach is infeasible. To address this challenge, the field has employed an appropriate animal model such as a rhesus monkey. However, given the large resource and temporal requirements of awake behaving intracranial experiments, such an approach to widespread algorithm design is still impractical. Faced with this reality, most algorithmic work has been done in offline neural data (simulated or real).

We ask here if a healthy human subject, using an entirely noninvasive prosthetic device driven by synthetic neural activity, can meaningfully inform the design of prosthetic decode algorithms. This system, which we call an online prosthesis simulator, represents a middle ground between simple (but perhaps less realistic) offline testing and more realistic (but difficult and resource intensive) animal model and human clinical trials. We detail the concept of this proposed system in Fig. 1. This figure shows in blue the dramatically increasing complexity of testing each algorithmic choice (algorithm or

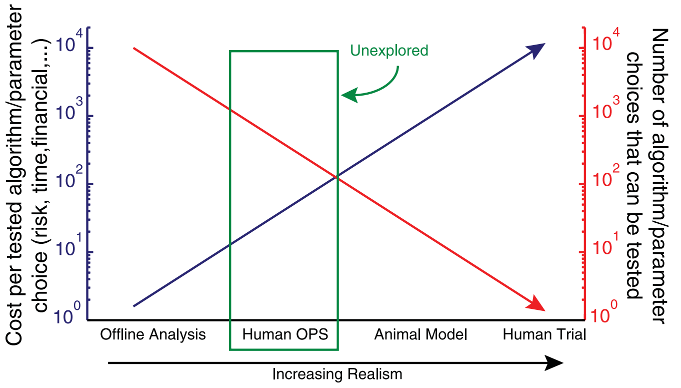

Fig. 1. Concept figure for online prosthetic simulator (OPS) opportunity. The *x*-axis shows 4 testing paradigms in terms of increasing realism. Offline data analysis is perhaps the least reasonable proxy to eventual user mode, as it entirely neglects the closed-loop control. On the other end of the spectrum is the human clinical trial, which is precisely the eventual user mode. Left axis (blue) shows the difficulty associated with testing each algorithm or algorithmic parameter setting. Right axis (red) shows the number of algorithm and parameter choices that are reasonably testable, given costs and other constraints.

parameter setting within a single algorithm) as researchers move towards animal studies or human clinical trials. This figure also shows (in red) the corresponding dramatic decrease in the number of algorithmic choices that can be meaningfully tested. Here we ask two questions: first, how different are offline and online analyses; and second, how similar are the OPS and closed-loop animal model BMI? The OPS, by analogy to flight simulators or silicon integrated circuit simulation software like "SPICE," may allow more realistic evaluation of current and future prosthetic decode approaches.

The creation of the OPS and the messages of this study have important connections to previous work. First, we note that previous literature has used signal sources other than motor cortex to control an external BMI-like interface. In Radhakrishnan et al. (2008), the authors noted that electromyography (EMG) may serve as a useful proxy to motor cortical signal for a BMI (a possible extension to the OPS that we discuss at length in the DISCUSSION). While connections to BMI design were discussed, this study primarily investigated the ability of the motor system to learn a nonintuitive control mapping (connecting to an important "fundamental neuroscience" aspect of the OPS that we describe in the DISCUSSION). From that study, we draw the key message that aspects of BMI design can be studied without the need for an invasive BMI. More closely, Danziger et al. (2009) used a sensored glove to map hand movements to cursor control and to study how subjects learned in the presence of an adaptive control mapping. They draw connections to "co-adaptation" for BMI, which has been a question of interest since Taylor et al. (2002). These two studies support the notion that a simulation system like the OPS can be used to study nontrivial interface controllers. While these studies focus primarily on neuroscientific aspects of human motor learning, the OPS is distinct by being designed specifically to investigate neural prosthetic algorithm and system design choices.

Second, two recent related works have investigated online vs. offline analysis in a specific BMI algorithm setting. First, Chase et al. (2009) compared two BMI algorithms in standard offline analysis and showed stark performance differences. However, when the same two algorithms were then analyzed under closed-loop control in a monkey experiment, the performance differences between these two algorithms became considerably less. One significant message of that study is that certain types of error (directional biases) are readily learnable in an online context, and so algorithms should not necessarily be discarded because of biases that impact offline performance negatively. Their second work (Koyama et al. 2010) then compared more algorithmic features, such as directional biases (again finding a discrepancy between offline and online performance implications) and trajectory smoothness (which seemed to matter in both online and offline contexts). This work is important in supporting the distinction between offline and online, but the OPS is distinct in that it offers a noninvasive simulation environment for rapidly testing such algorithmic features in true closed-loop. One abstract (Marathe et al. 2009) did use humans noninvasively in this offline vs. online context, with a similar spirit to the OPS. Here subjects used a joystick or EMG to control a virtual arm, and the authors revealed the negative impact of systems that accentuate low-frequency control signals, insomuch as such signals slow online feedback control.

The studies (Chase et al. 2009; Koyama et al. 2010) also included simulation of one difference between online and offline control and supported the value of further simulations. However, as the authors noted, this simulation model did not include online correction and the use of feedback control. They rightly noted that such a computer simulation would be considerably more involved and heavy with assumptions. Here, we introduce the OPS as a system to accurately model the human feedback-control system, using an actual human in closed-loop with the decoding algorithm. Furthermore, the OPS system allows verification of these findings and extensions to new algorithmic questions without full animal model experiments.

The remainder of this study is as follows: in METHODS, we describe the experimental hardware and software platform that allows us to test the OPS in humans, the OPS in monkeys, and a real neural prosthetic BMI in monkeys (we term these testing scenarios "human OPS mode," "monkey OPS mode," and "monkey BMI mode," respectively). We describe two variants to a simple center out reaching task that we had both humans and monkey perform, and we detail relevant data analysis methods. In RESULTS, data from humans and monkey demonstrate that the subjects using the OPS paradigm show significant performance differences when using different bin widths of a Kalman filter decode algorithm. We then compare these online results to offline analysis to make the first major point of this study: offline analysis does not provide an accurate picture of online performance. As a second major point, we then show using the monkey data that the OPS paradigm accurately reflects these trends, indicating similar algorithmic choices as does real BMI mode. Finally, in the DISCUSSION, we discuss the implications of these results.

METHODS

In this work, we performed human and animal experiments, offline and online analyses, in OPS and BMI modes, and with two different task variants. To manage the description, we break up the methods below as follows. First, we describe the relevant experimental hardware that was used for all experiments. Second, we describe the reaching task performed by both the animals and humans. Third, we describe human-specific protocols. Fourth, we describe animal-specific protocols (including surgery and neural data acquisition). Fifth, as the ability to generate sensible synthetic neural data is a key aspect of the OPS, we describe in detail our methods and choices for neural spiking models. Sixth, we describe the decode algorithm that was used to generate prosthetic cursor control in both the online and offline cases. Seventh and finally, we describe the methods used to analyze the performance of these varied experimental conditions.

Neural prosthetic experimental hardware. We first describe the relevant experimental hardware and system. This experimental rig is diagrammed schematically in Fig. 2. Importantly, this hardware was used in all experiments, both by the human and animal subjects, in both OPS and BMI modes, so we describe it here without distinction to the user. This choice is intentional to further emphasize our effort to make the OPS a close proxy to real BMI mode. In the human experiments, the subject sat in a chair and placed his/her head comfortably on a chin rest. In the animal experiments, the subject sat with a fixed head position in a custom chair. In both cases, the subject's nose was positioned directly in front of a pair of mirrors (at 45-degree angles from the eye). Each mirror reflected an image from a pair of LCD monitors on either side of the subject's head. These monitors displayed identical images but with a slight (multipixel) offset to create a disparity cue leading to a stereoscopic depth percept, creating the illusion of a 3-D environment [a typical Wheatstone stereo 3-D display (Wheatstone 1838)].

The subject made arm reaches in the large 3-D volume behind the mirrors. We recorded kinematic parameters of the reach endpoint using a small reflective bead on the subject's finger, which allowed overhead optical position tracking (Polaris, Northern Digital, Waterloo, Ontario). Hand position was measured at 60 Hz to a resolution of 0.35 mm root mean square. These data were recorded at 1,000 Hz by a generic x86 "behavior" computer running an embedded operating system (xPC; MathWorks, Natick, MA) executing custom software. This computer processed the kinematic data (for example, in OPS mode, to generate synthetic neural data and decode that data into a decoded reach, specifics described below) or neural data (in BMI mode, specifics described below) and transmitted a cursor position via ethernet to a "visualization" x86 computer running a stripped-down Linux operating system. This machine executed visualization software (MSMS; USC MDDF, Los Angeles, CA) that rendered images to the Wheatstone display in near real time (measured latency and jitter in our system: 7 ± 4 ms). This low-latency system critically allowed us to investigate short timescale prosthetic design questions. For example, in this work we were interested in optimizing the integration bin width of the Kalman filter. However, if our system (which includes the end-to-end integration of behavioral control, neural data streaming and recording, and decoding) had latency and jitter in excess of 25 ms, 50 ms, or more, we could not have investigated any bin widths lower than that threshold. This carefully engineered system was thus an important enabler of this study.

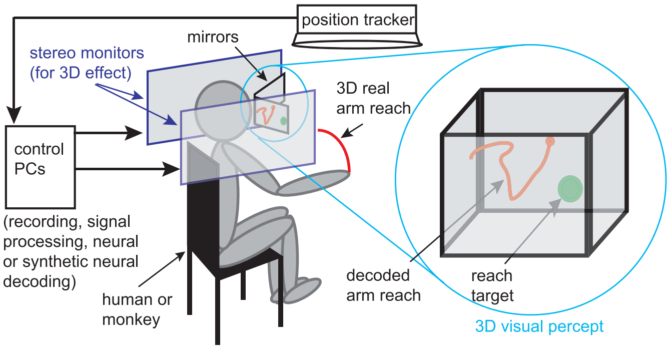

Fig. 2. Experimental rig that can be used both for OPS and real neural. Brain machine interface (BMI) experiments. A human (shown in gray) or monkey subject reaches (red trace) in a 3-D volume obscured from view. An overhead position tracker tracks endpoint kinematics. Control PCs process those data and render the subject's real reach (in control trials) or a prosthetic decoded reach (in OPS or BMI trials, red trace). Two monitors (blue) project a stereo 3-D image onto mirrors (virtual 3-D environment shown at *right*). Also displayed is a the subject's reach target (green sphere).

This virtual environment rendered, against a black background, a reach target (4 cm diameter green sphere) and the subject's decoded hand/cursor position (as a 4-cm diameter gray sphere, although it is shown in Fig. 2 as a red sphere for clarity of illustration). In control reaching trials, we rendered the subject's true hand position back to the display as the gray cursor. In closed-loop prosthetic trials (either OPS or BMI mode), we rendered a decoded prosthetic reach as the same gray cursor. This system, similar in spirit to the virtual environment in Taylor et al. (2002), allowed online, closed-loop neural prosthesis trials appropriate for the work proposed here.

Neural prosthetic experimental task. Having described the hardware in which the subjects made reaches, we here describe the specific structure of the experimental task. To make close comparisons between the offline and online analyses, and between the OPS and BMI modes, we held constant the task design across all human and monkey subjects.

All subjects made center-out reaches in the virtual environment described above. Each reach consisted of the subject moving the gray cursor (under subject control) to the green reach target. The green reach target alternated between a center point and a pseudo-randomly chosen target at one of eight target locations evenly spaced on a ring of radius 8 cm. Only one target appeared per trial. In all data, only the center-out reaches (those from the center point to the periphery) were analyzed.

The specific trial timeline proceeded as follows: the trial began when the green reach target (either center point or one of the peripheral targets) appeared on the screen. As this was a direct-reach paradigm, the target appearance was the subject's "go" cue, and after a short reaction time, the subject moved the gray cursor to the green target. The reach was successful if the gray cursor was held within a demand box 4-cm wide around the reach target. The cursor had to remain within the acceptance window for a hold period of 500 ms, after which a reward was given (a tone for human subjects, and a tone and drop of juice for the monkey). If that success criterion was not satisfied within 3,000 ms, the trial was considered a failure and timed out. After either a trial success or failure, an inter-trial interval of 40 ms was imposed before the beginning of a new trial. The cursor was controlled by the subject using one of three modes: real reaching, OPS prosthetic reaching, and BMI prosthetic reaching, as described in the next paragraph. During all of these control modes, the subject (monkey and human) was allowed free movement of his/her limb [as has been done previously in BMI literature, for example Carmena et al. (2003); Serruya et al. (2002); Taylor et al. (2002)]. Because the subject only saw the cursor (the real arm was hidden from view), the subject remained motivated to complete the task by the reward structure, which rewarded successful control of the cursor. Although there was a sensory discrepancy between the visual feedback and the real arm's proprioception (in prosthetic trials), previous motor neuroscience work (Radhakrishnan et al. 2008) and previous BMI literature (Carmena et al. 2003; Serruya et al. 2002; Taylor et al. 2002) suggest that this confound is minor and does not seriously affect task learning. Whether the true arm is restrained or allowed to reach, there is still a proprioceptive confound, so this potential limitation is not specifically of the OPS or of this study but rather of all able-bodied animal or human BMI studies. We discuss this aspect of the experimental design in more depth in the DISCUSSION.

There are three modes by which the subject controlled the gray cursor: real reaching, OPS prosthetic reaching, and BMI prosthetic reaching. In real reaching, the gray cursor reflected the kinematics of the true underlying arm reach. These data are that are typically collected for offline prosthetic decode analysis, and they were used for training the parameters of the prosthetic decode algorithms. Alhough other training paradigms have been used (Hochberg et al. 2006), we chose this simple training method to parallel existing offline and online literature. In BMI prosthetic reaching (monkey only), neural activity was recorded from the motor cortex, and those data were decoded into kinematics which controlled the gray cursor. In OPS prosthetic reaching mode (monkey and human subjects), synthetic neural activity was generated from the kinematics of the subject's arm, and that neural activity was then decoded with a decoding algorithm into control commands for the gray cursor. The specifics of recording, generating synthetic activity, and decoding are detailed below.

The parallel between OPS and BMI control modes is detailed in Fig. 3. In Fig. 3, *left* and *right*, the subjects make a real reach (*a*; red trace) and then corresponding neural activity (*b*) is recorded [synthetic neural activity generated from the real reach in OPS mode (*left*) and real neural activity corresponding to the real reach in BMI mode (*right*)]. Those data are then input to the decode algorithm (*c*), which decodes the (possibly synthetic) neural activity into kinematics to control the gray cursor, which is rendered back to the subject (*d*). The decoded reach trajectory will differ from the subject's intended arm reach, due to noise (from neural spiking) and algorithmic model mismatch (e.g., the Kalman filter only approximately models the dependency of the neural activity on kinematics). The subject modifies his/her behavioral strategy so that the prosthetic movement will mimic, as closely as possible, the desired reach trajectory. Thus both OPS and BMI modes offer a means to study the relevance of feedback control in neural prosthetic system use.

We also tested two variants of the center-out prosthetic reaching task. In the first, the "continuous" task variant, prosthetic reaches were made both for center-out reaches to the peripheral targets and for the reaches returning to the center target. We collected full data sets from this task in the human OPS, and there was a mildly frustrating "drift" effect that could occur, where, over many reach trials, the cursor would become increasingly offset from the true arm reach. Although this never created a problem for the data (performance was robust to this effect), it was reported by some subjects to be frustrating. As such, we developed a second task variant. In the "interleaved" task variant, the reaches returning to the center target were controlled by the real arm kinematics. This interleaving removed the potentially frustrating drift effect. The center-out reaches (which are the only reaches analyzed) remained entirely under prosthetic control. Although in eventual neural prosthesis system use the "return to center" may be controlled by the system itself, this interleaved variant was a simple way to keep the subject engaged in the task and to avoid confusion. We ran human subjects in both OPS continuous and OPS interleaved. We ran the monkey on BMI continuous and BMI interleaved, which presented no difficulty, since the monkey had been highly trained using real neural activity in exactly this continuous paradigm with this BMI controller. For the monkey in OPS mode, we only ran OPS interleaved, as the moderately frustrating effects reported by human users (with OPS continuous) would require more extensive training with the monkey, and that variant is of unclear additional merit on top of the interleaved task.

For each of these variants, we ran five full experiments. Here we first describe the full experimental structure and how it allowed fitting and testing of numerous decode models; the particulars of task training are described in *Human experiments* and *Animal experiments*. A full experiment is a block structure designed as follows. First, the subject (human or monkey) performed 1 block of 200 trials in real reach mode. These 200 real reaches and the corresponding neural recordings are used to train the decoding models (Kalman filters with different temporal bin widths). The subject was given a roughly 2-min break, during which online prosthetic control mode (either OPS or BMI) was switched on. The subject then performed 7 blocks of 100 prosthetic reaches each, where there was a short 15-s break between blocks. These 7 blocks of 100 reaches were then repeated again in a different order. Thus a "full experiment" included roughly 1,600 reaches, 200 real reaches followed by 2 runs of 7 ×100 reaches. Typically this took subjects about an hour. This experimental structure allowed an algorithm choice or be changed. As discussed, here we are interested in optimizing performance based on the temporal bin width of the Kalman filter decode algorithm. Accordingly, we chose seven bin widths, 25, 50, 100, 150, 200, 250, and 300 ms, and we had the subjects do two blocks with each bin width (1 block of 100 reaches per 7-block run). We randomized the order of the blocks (within the 7-block run) to control for any learning effects. Each of the nine human subjects ran one full experiment (either OPS continuous or OPS interleaved, save for *subject PN*, who ran each on different days), and thus we have five full experiments each of human OPS continuous and human OPS interleaved. The monkey ran all 15 full experiments (5 each of monkey BMI continuous, monkey BMI interleaved, and monkey OPS interleaved). From the real reaches that comprise the first block of the experiment, we are able to run all offline decoders, so the offline analyses were built from each subject's real reaching blocks.

This general task structure allows us to study online vs. offline analyses, BMI vs. OPS mode, continuous vs. interleaved, in humans vs. monkey. To clarify, in RESULTS we show 10 different types of prosthetic decode analysis: offline human OPS continuous, offline

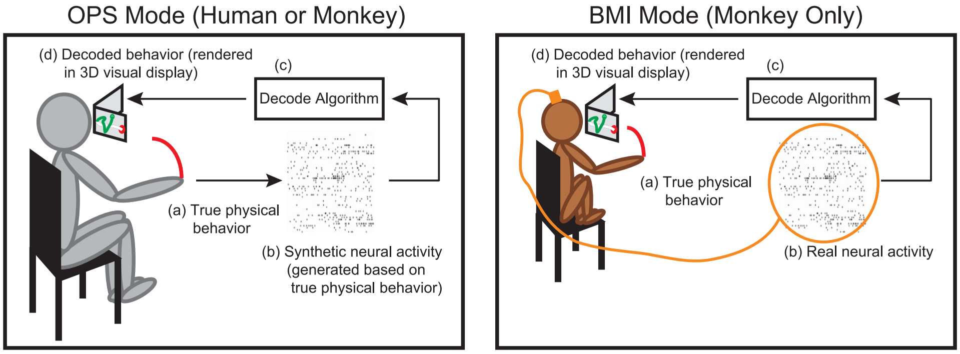
OPS Mode (Human or Monkey)
(d) Decoded behavior (rendered in 3D visual display)
(c)
Decode Algorithm
(a) True physical behavior
(b) Synthetic neural activity (generated based on true physical behavior)
BMI Mode (Monkey Only)
(d) Decoded behavior (rendered in 3D visual display)
(c)
Decode Algorithm
(a) True physical behavior
(b) Real neural activity

Fig. 3. OPS and BMI schematic. *Left*: OPS mode. A healthy human subject or a monkey (human shown) makes real arm reaches in a 3-D reaching environment (*a*) (see Fig. 2). Recorded kinematics of that real reach are used to generate synthetic neural activity (*b*), which is then input to the Kalman filter neural decode algorithm (*c*). Decode algorithm decodes synthetic neural activity into physical reaching behavior, and decoded reach is then rendered back to the subject in the 3-D visual environment (*d*), which allows the subject to bring to bear all of his/her online, closed-loop control strategies to drive a desired reach. *Right*: BMI mode. A monkey makes real arm reaches (*a*). Real neural activity associated with that reach is recorded (*b*), which is then input to the Kalman filter decode algorithm (*c*). Decode algorithm decodes real neural activity into physical reaching behavior, and decoded reach is then rendered back to the monkey in the 3-D visual environment (*d*), which allows similar closed-loop control as in OPS mode. It is important to note that the OPS mode of the *left* can be used both by a human and a monkey (this will provide an important validation of the OPS).

human OPS interleaved, offline monkey OPS interleaved, offline monkey BMI continuous, offline monkey BMI interleaved, online human OPS continuous, online human OPS interleaved, online monkey OPS interleaved, online monkey BMI continuous, and online monkey BMI interleaved. These data sets are categorized explicitly in Table 1. These permutations allow us to perform a breadth of comparisons and controls to answer the key questions of this study. In the following two sections, we describe particular protocols used for the monkey and human subjects only (not shared across monkeys and humans as the above task and hardware details are).

Human experiments. Human protocols were approved by the Stanford University Institutional Review Board. Nine healthy adult human subjects performed the reaching tasks in both real reach and OPS modes as described above. These subjects reached in the experimental apparatus previously described. Five subjects performed a full experiment (200 real reaches plus 2 blocks of 100 reaches at each of 7 bin widths, as described above) of the OPS continuous task, and five subjects (4 new subjects, and 1 repeated from the OPS continuous) performed a full experiment of the OPS interleaved task.

Across all experiments, in OPS and real reach modes, human subjects required very little training (only a few trials, which are not included in the results and analysis) to understand the task and control the cursor well. During real reaching trials (200 reaches), which are used both for training the decode algorithm and for offline analysis, we asked the subjects to reach to the targets at a comfortable, normal speed. Since the virtual path of the cursor matches perfectly the true kinematics of the arm, the subjects had no trouble performing these reaches with 100% accuracy. We then paused the task and informed subjects that they were being switched into prosthesis mode for several blocks of 100 reaches, where the virtual reach would not perfectly match their true underlying reach. We asked the subjects to try to maintain the same reach speed and accuracy as in their previous reaches, and we enforced the time-to-target timeout if subjects took too long to reach (3,000 ms, as previously noted). Depending on the bin width being used, subjects were more or less able to acquire the targets successfully and quickly (these performance differences are the results of the OPS).

These human experiments provide considerable data for testing the difference between offline and online (OPS) analyses in terms of the subject's ability to exploit feedback control to improve prosthetic reaching performance.

Table 1. *Categorization of the data sets*

	Human Subjects	Monkey Subjects
Online analysis with synthetic neural data	Online Human OPS Continuous Online Human OPS Interleaved (Figures 4, 6)	Online Monkey OPS Interleaved (Figures 5, 7)
Offline analysis with synthetic neural data	Offline Human OPS Continuous Offline Human OPS Interleaved (Figure 6)	Offline Monkey OPS Interleaved (Figure 7)
Online analysis with real neural data	No Data (requires human clinical trial)	Online Monkey BMI Continuous Online Monkey BMI Interleaved (Figures 5, 7)
Offline analysis with real neural data	No Data (requires human clinical trial)	Offline Monkey BMI Continuous Offline Monkey BMI Interleaved (Figure 7)

Categorization of the data sets collected and analyzed in the study. Results show online (*first* and *third rows* in table and blue in figures) and offline analysis (*second* and *fourth rows* in table and red in figures), human and monkey subjects (*middle* and *right columns* in table), synthetic and real neural data (identical to OPS and BMI modes, *top two* and *bottom two rows* in table), and finally the continuous and interleaved task variants (grouped within each cell of this table). These permutations allow us to perform a breadth of comparisons and controls to answer the key questions of this study.

Animal experiments. Animal protocols were approved by the Stanford University Institutional Animal Care and Use Committee. We trained one adult male monkey (*Macaca mulatta*, monkey L) to perform center-out reaches as already described. Unlike humans, monkeys were motivated to complete the reaches with a juice reward. During experiments, the monkey sat in a custom chair (Crist Instruments, Hagerstown, MD) with the head braced. Hand position recording and experimental control were as previously described.

To enable BMI trials, a 96-channel silicon electrode array (Blackrock Microsystems, Salt Lake City, UT) was implanted straddling dorsal premotor (PMd) and motor (M1) cortex (right hemisphere), as estimated visually from local landmarks, contralateral to the reaching arm. Surgical procedures have been described previously (Churchland et al. 2006; Hatsopoulos et al. 2004; Santhanam et al. 2006). Neural signals were monitored on each channel and high-pass filtered. A threshold level of −4.5 times root mean square voltage was established for each channel, and all threshold crossings were recorded as "spike times." Spike sorting was not performed, as our previous experiments indicated that doing so did not significantly alter performance to justify the additional computational load [Chestek et al. (2009); see also, Fraser et al. (2009); Herzfeld and Beardsley (2010); Santhanam et al. (2004)].

The monkey was trained over several months on the reaching task and BMI continuous task variant (Gilja et al. 2010). For the interleaved task variant, in both the OPS and BMI modes, the monkey quickly learned this task in a few days of training, as it was only a minor extension of his well-trained continuous BMI behavior. Multiple data sets of each task were collected. For the continuous BMI task, the monkey completed five full experimental sessions (wherein each 200-trial block with an individual bin width was presented twice in pseudo-random order, as described above) across 3 days, and those sessions are denoted as L20091215.C3.1, L20091217.C3.1, L20091217.C3.2, L20091217.C3.3, and L2009 1218.C3.1. For the interleaved BMI task, the monkey completed all five experimental sessions across 2 days, and those sessions are denoted as L20091218.IRR3.1, L20091220.IRR3.1, L20091220.IRR3.2, L20091220.IRR3.3, and L20091220.IRR3.4. For the interleaved OPS task variant, the monkey completed all five full experimental sessions in a single day, and those sessions are denoted as L20091221.IRR2.1, L20091221.IRR2.2, L20091221.IRR2.3, L20091221.IRR2.4, and L20091221.IRR2.5. To preserve trials, real reach trials were only run once per day for the monkey, not in each experiment.

These animal experiments provide considerable data for again testing the difference between offline and online analyses (whether comparing to BMI online or OPS online). Further, this data allows us to test the extent to which the OPS paradigm accurately reflects the trends of real BMI mode.

Generating synthetic neural activity. There are many behavioral features correlated with motor cortical spiking activity, including features of the arm (hand position and velocity, muscle activity, forces, joint angles, etc.), features of the reach (smoothness, etc.), and features of the eye (visual reference frames, etc.) [see Todorov (2000) for extensive references and discussion]. In this work, we recorded endpoint kinematics of the subject's arm, and we used these data to produce reasonable simulated neural spike trains that were then decoded into kinematics for controlling the prosthetic cursor.

Certainly generating synthetic neural data requires some uncomfortable assumptions about neural tuning and spiking. While synthetic neural activity has been used in the past in prosthesis studies (Brockwell et al. 2004; Kemere et al. 2004; Srinivasan and Brown 2007; Srinivasan et al. 2007, 2006; Ventura 2008), we are particularly interested here in creating a simulation system that translates well into a real neural system. Much care is needed both in construction and in interpretation (discussed more fully in the DISCUSSION) to ensure the legitimacy of the results. Accordingly, our approach was to pick as simple a construction as possible that resulted in decoded reaches (both online and offline) that were qualitatively similar to our core experience in real neural prosthetic experiments (Cunningham et al. 2009, 2008; Kemere et al. 2004; Santhanam et al. 2006; Yu et al. 2007).

To begin, we used cosine-tuning models (Georgopoulos et al. 1986; Moran and Schwartz 1999) to map kinematics to neural firing rates. Under this tuning model, each neuron was defined by a tuning vector, a minimum firing rate, and a maximum firing rate (these last 2 are equivalent to setting a mean rate and a depth of modulation). First, we chose a population size of 96 neurons (equivalent to the 96 electrode channels of real neural data that we recorded with the array). For each of these synthetic neurons, we sampled tuning vectors uniformly from the unit 3-D sphere, for both position and velocity (so these tuning vectors corresponded to "preferred position" and "preferred velocity" vectors). We sampled minimum firing rates uniformly between 0 and 20 spikes per second, and we sampled maximum firing rates uniformly between that neuron's minimum rate and 100 spikes per second, in line with motor cortical neuron behavior we have previously observed (Churchland et al. 2010; Churchland and Shenoy 2007; Cunningham et al. 2009). These firing rates were then put through a spiking process, which we chose to be a simple Poisson process with rate equal to the firing rate in each time bin t. This is also a standard choice in literature (Dayan and Abbott 2001; Schwartz 2004).

Mathematically, we say that neuron k had tuning vector $c^{(k)}$, maximum firing rate $\lambda_{max}^{(k)}$, and minimum firing rate $\lambda_{min}^{(k)}$. Then, we defined x_t to be the kinematic parameters of the arm in time bin t, where $x_t \in \mathbb{R}^6$ a vector of 3-D position and velocity of the hand. The x_t was suitably scaled such that the range of kinematics exhibited by the subject produced firing rates within the range of minimum and maximum firing rates. We defined $\lambda_t^{(k)} \in \mathbb{R}^K$ and $y_t \in \mathbb{N}_+^K$ to be the neural firing rates and spiking activities at time bin t [each element $\lambda_t^{(k)}$of λ_t and $y_t^{(k)}$ of y_t corresponding to the firing rate and number of spikes of each of the $K = 96$ synthetic neurons being generated]. With these definitions, we write:

$$\lambda_t^{(k)} = (\lambda_{max}^{(k)} - \lambda_{min}^{(k)})c^{(k)} \cdot x_t + \lambda_{min}^{(k)} \tag{1}$$

$$y_t \Big| \lambda_t \sim \text{Poisson}\Big[h(\lambda_t)\Big] \tag{2}$$

where $c^{(k)} \cdot x_t$ represents the inner product between these two vectors, and the function $h(\lambda_t)$ clips the rate λ_t to 0 if the argument is negative and otherwise equals the argument λ_t [as is commonly done in literature (Chase et al. 2009), although given our parameter choices, this clipping rarely occurred]. In the first generation of the OPS system, we created this tuning to both 3-D position and 3-D velocity kinematics, as has been reported considerably in literature (Georgopoulos et al. 1986; Moran and Schwartz 1999). While this produced reasonable decoded behavior offline, online use was not qualitatively similar to real prosthetic use. With strong position dependence, users immediately developed a strategy of moving the hand to the virtual location of the target and then holding that position. Because there was position tuning information, eventually the cursor would acquire the target for a full hold period. This behavior was qualitatively very different from the observed behavior in monkey BMI experiments, where subjects make several online adjustments to the real reach and are rarely able to successfully acquire a target by holding a hand position for a period of time. To correct this qualitatively inappropriate effect, we removed the position dependence in the simulated neurons. Instead, firing rate was only tuned to velocity [so x and corresponding $c^{(k)}$ are now only 3-D, including 3-D velocity terms]. With this model, the qualitative behavior of OPS users was highly similar to that of a monkey using a real BMI. After substantial testing to ensure that other unrealistic strategies were not developed, we determined that this simple model was adequate for testing the decode implications of the Kalman filter bin width. We note that other possibilities for noise generation could include adding noise sources to either the firing rate or to the spike trains themselves (to model recording noise sources, for example), but we leave those enhancements to future work.

Here we note two important facts about this choice of spiking model. First, this spiking model does not match the decode algorithm, in the sense that the Kalman filter is not an optimal decoder of neural activity generated in this way. Indeed, the Kalman filter is also not an optimal decoder of real neural activity, as the Kalman filter does not exactly model the real neural system. Second, it is important to note that we do not claim that this is what the neural system is actually doing, but instead it is a choice that allows us to study the OPS framework and the relevance of feedback control to the design of neural prosthetic systems.

Finally, this method for generating synthetic neural activity can be extended in a number of ways previously seen in literature [e.g., Srinivasan and Brown (2007); Srinivasan et al. (2007, 2006)], but the simple velocity-tuned model was so qualitatively and quantitatively similar to real BMI mode that we were satisfied with this straightforward choice.

Decoding neural activity. We decoded neural activity using the Kalman filter, as has been used for neural prosthetic algorithms previously (Kim et al. 2008; Wu et al. 2002, 2004, 2006). It is important to note that this algorithm can be used both in online and offline contexts. In the online case, at each time point, the decoded kinematics controlled the cursor, which the subject could see, use as feedback, and react to. In the offline case, the neural (or synthetic neural) activity was collected during real reaches (when the cursor matches the arm kinematics) and later decoded offline. By using identical decode algorithms (and parameter settings, etc.), we can effectively study the difference between online and offline control. The Kalman filter [introduced in Kalman (1960)] stipulates a linear dynamical system for arm movements (often called the prior or the trajectory model), which says that kinematics at time t should look something like kinematics at time $t - 1$, i.e., smoothness. The model also stipulates a linear observation model for neural activity, which says that observed neural spiking is a noisy linear transformation of intended kinematics. As before, we define x_t to be the kinematic parameters of the arm in time bin t, where $x_t \in \mathbb{R}^7$, a vector of 3-D position and velocity of the hand (plus a bias/offset term). We also define $y_t \in \mathbb{R}^K$ to be the neural activity at time bin t (each element of y_t corresponding to the activity of each of the $K = 96$ neurons being recorded). The Kalman filter then assumes the following linear dynamical system:

$$x_t = Ax_{t-1} + w_t \tag{3}$$

$$y_t = Cx_t + q_t \tag{4}$$

where $A \in \mathbb{R}^{p \times p}$ and $C \in \mathbb{R}^{K \times p}$ represent the state and observation matrices, and w_t and q_t are additive, independent Gaussian noise [denoted $w_t \sim N(0, W)$ and $q_t \sim N(0, Q)$]. Such a model is standard for Kalman filtering, and it allows very fast inference (decoding) of kinematics from neural activity. Furthermore, the parameters $\{A,C,W,Q\}$ can also be quickly and exactly learned from training data [the standard details of which are left to the references of Kalman (1960); Kim et al. (2008); Wu et al. (2002, 2004, 2006)].

Intuitively, the Kalman filter starts from kinematics at the beginning of the trial and proceeds iteratively through time, updating its estimates of arm state and error covariance at every time step t. These steps are entirely based on mathematical properties of the Gaussian, and the algorithm is fast and stable. Again, this is precisely the context that has been used for online and offline neural prosthetic algorithms previously (Kim et al. 2008; Wu et al. 2002, 2004, 2006).

One important feature to note within the above description is the time bin t. Since the above dynamical system is discrete time, we must choose a window of time over which to integrate neural activity for our prosthetic decode. This time, which we call the "decode integration bin width" or just "bin width," is a key parameter that can be chosen by the experimenters. As noted in the introduction, previously published offline analysis has suggested that a 200- to 300-ms bin width is optimal (Wu et al. 2006), but this choice has not been validated online. We will see that different settings of this parameter can have a large performance effect both qualitatively and quantitatively. Thus optimizing performance with respect to this parameter is an important research question and a valuable test for the OPS, which we present here.

Performance analysis. Having decoded the neural activity as described in the section above, we must now introduce metrics to quantify the performance of a particular prosthetic mode (task variant, subject, bin width choice, etc.). Error metrics are numerous and a subject of study in their own right (Douglas et al. 1999), but here we chose a few sensible metrics that have been seen previously in literature. For consistency, we present all metrics as error and not performance metrics; i.e., lower is always better throughout the results.

First, for online analysis, failure rate is an obvious and previously used choice [e.g., Santhanam et al. (2006)]. Failure-rate measures the fraction of trials in which the subject was unable to acquire and hold the reach target within the allotted time. Second, time-to-target, as used in Hochberg et al. (2006), measures the amount of time the subject required to reach the target (for the last time, since the subject could pass through the target a number of times before holding the cursor at the target for the required hold period).

While these straightforward performance metrics offer a quantitative view of online performance, they cannot be meaningfully used for offline decode analysis. Offline decoded reaches rarely reached the target and never successfully completed a trial (this makes sense with a noisy decoder in the absence of feedback, which is yet another suggestion that online analysis should perhaps be prioritized), so failure rate is a vacuous metric in comparing algorithms or algorithmic choices offline. Time-to-target is similarly broken and thus cannot be used for offline analyses. Standard metrics for offline analysis include mean-squared error or similar, and correlation coefficient, and these have been used extensively [see description in Cunningham et al. (2009)]. With these metrics, one compares the true, natural reach to the offline decode of that same reach. Unfortunately, these sensible offline metrics are not useful online, as there is no correct "true reach" that can be compared.

To meaningfully compare offline analysis with online analysis (the first major point of this work), we require a metric that is suitable both in the offline and the online context. As the subject was always motivated to move closer to the target during a reach (whether in OPS, BMI, or real reaching mode), we can consider the distance to the target. We define mean-integrated-distance-to-target as the average distance to the target during the time course of the subject's reach. On trial i, if we call the target position $p^{(i)}_{targ}$ and the position at time t during the trial $p^{(i)}_t$, the error $E^{(i)}$ for trial i can be written mathematically as:

$$E^{(i)} = \frac{1}{T^{(i)}} \sum_{t=1}^{T^{(i)}} \| p^{(i)}_t - p^{(i)}_{targ} \| \tag{5}$$

where $T^{(i)}$ is the length of trial i. Using the mean distance instead of total distance is sensible because offline and online trials have characteristically different temporal lengths (since offline trials are based on real reaches, these are generally quicker than online, noisy prosthetic trials). It is important to note that, when comparing offline and online performance (and BMI vs. OPS mode performance), less will be made of absolute differences in performance between offline and online (and BMI vs. OPS modes); rather, we are mostly interested in the optimal parameter settings that are suggested by different prosthetic modes.

Mean-integrated-distance-to-target is by no means the only metric possible for comparing offline and online trials, so we here give a context for why we believe this metric to be appropriate. First, since the actual task rewards movement of the cursor to the target both in offline and online settings, having a metric that considers the target is sensible. One might reasonably consider other metrics that compare the prosthetic reach to some true arm kinematics such as an "ideal" real arm reach (perhaps gleaned from real reaching trials). However, the subject is not rewarded for having a particular kinematic profile, the reward is for task success, i.e., acquiring the target. Quantifying prosthetic reach error based on a quantity that is unknown to the subject and only somewhat related to task success does not seem appropriate. Put another way, a subject using a BMI might develop a different success strategy (changing his/her kinematics, for example), and "ideal reach" metrics could arbitrarily penalize such a strategy. On the other hand, the distance to target is sensible because, by task design, we know that the subject is always trying to reduce this distance, and thus motivation and performance are aligned.

Finally, we note that each of these metrics was calculated on a per trial basis, and thus we could calculate average statistics and confidence intervals. Throughout the results we use 95% confidence intervals, calculated via the binomial distribution for failure-rate analyses and Gaussian distributions for time-to-target and mean-integrated-distance-to-target analyses (Zar 1999). These performance analyses give us all the quantification necessary to analyze the results gathered via previous methods.

RESULTS

The results are composed of three pieces, which we describe here in brief to help the reader navigate the section. We begin by showing the raw performance data across subjects. These data will demonstrate that all subjects using the OPS and BMI modes show significant performance differences when using different bin widths of a Kalman filter decode algorithm. In other words, the raw data show that the bin width does have meaningful performance implications and further that this prosthetic setup is a meaningful way to investigate those implications. Next, we make the first major point of this study: offline analysis does not provide an accurate picture of online performance. The data show significant differences in the shape of these performance curves based on offline or online (BMI/OPS) modes, thereby suggesting very different algorithmic parameter choices. Finally, we make the second point of this study: the monkey BMI data, the human OPS data, and the monkey OPS data all show similar performance trends. This importantly indicates that, at least when it comes to this one critical algorithm parameter, the OPS framework can stand as a reasonable proxy to real BMI mode.

Online prosthetic data. We begin by showing the performance for both humans and monkey in the online prosthetic modes, both BMI and OPS. In Fig. 4, we show the error for the human subjects; *left* shows the OPS continuous task variant, and *right* shows the human OPS interleaved variant. Within a given column, $A-C$ shows the same data analyzed with a different error metric: failure rate (A), time to target (B), and mean integrated distance to target (C). Each of these metrics is described in METHODS. As a reminder, this third metric, mean integrated distance to target, is required to compare offline and online analyses. In $A-C$, each faint line denotes the average error of a particular human subject over the course of that subject's full experiment. For example, in Fig. 4*A*, *right*, the light green trace shows the average failure rate of *subject JH* at each of the bin widths. We see that, at a bin width of 200 ms (*subject JH* did 200 reaches at this bin width, per the block structure protocol described above), *subject JH* failed to acquire and hold the target on roughly 60% of trials. Since we are particularly interested in average error at a bin width (and less interested in intersubject differences other than to confirm that the trend is the same in all subjects), in dark blue we plot the average across all subjects, along with 95% confidence intervals (as described in METHODS). This color scheme is consistent in Fig. 4, *A*−*C*.

First and foremost, it is immediately apparent in Fig. 4, *A*−*C*, that in the human OPS, shorter integration bin widths lead to lower error in the online setting. The raw data give confidence that there are meaningful performance differences across this algorithm parameter. By performing a linear regression to the data in Fig. 4, *A*−*C* (binomial noise model for failure rates in *A*; gaussian noise model for other error metrics in *B* and *C*), we can calculate the confidence level that each panel is indeed a positive sloping line (i.e., shorter is better). For the human OPS continuous, those values are as follows: *A*: $P < 10-4$; *B*: $P < 10-4$; *C*: $P < 10-4$. For the human OPS interleaved, those values are as follows: *A*: $P < 10-4$; *B*: $P < 10-4$; *C*: $P < 10-4$. Thus with high confidence we can say that the human OPS indicates that shorter integration bin widths will lead to higher online performance.

There are a few salient features to point out in this data. First, it is encouraging that all error metrics suggest a similar trend in the data, giving us confidence that we are looking at a real trend and not an artifact of the summary metric. This fact will also become useful when we compare offline to online OPS, which we can only do with the metric of Fig. 4*C* (since failure rate and time to target are not meaningful in offline analysis, as previously described). Second, it is encouraging that the data are rather insensitive to the human OPS continuous or OPS interleaved task variants (Fig. 4, *left* and *right*). Recall that the interleaved variant was introduced to prevent user frustration. The data robustness to that frustration indicates that it was not critical to subject performance and did not invalidate the data. Third, it is encouraging to see that subjects performed rather similarly compared with one another. If there were substantial intersubject differences, more investigation might be required. However, in both OPS continuous and OPS interleaved, we had a few experienced subjects (*subjects JC*,

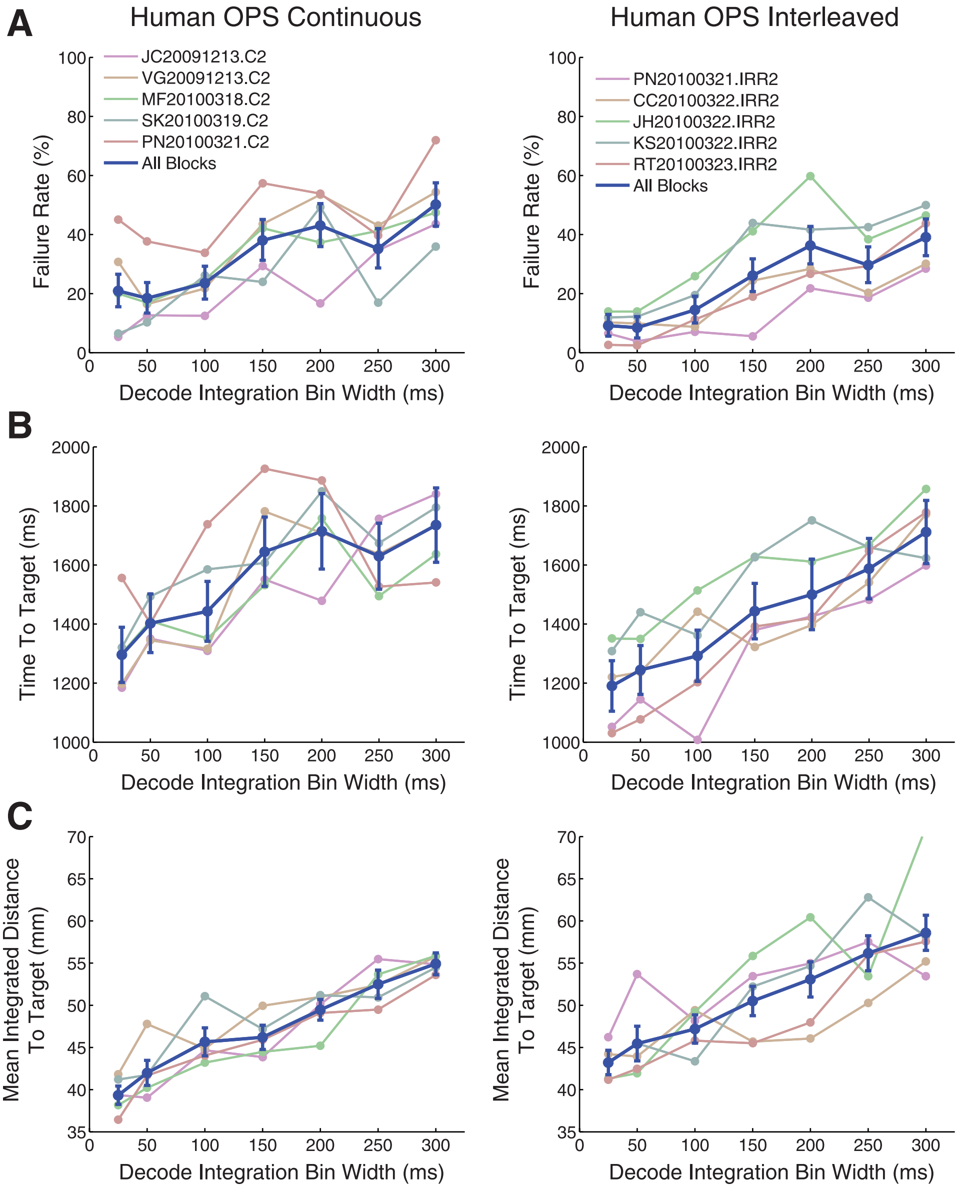

Fig. 4. Performance metrics for human OPS decode trials. *Left*: continuous task variant. *A*: failure rate, the percentage of trials where the subject did not successfully acquire and hold the reach target. *B*: time to target, the time required to reach the target for successful trials. In *A* and *B*, data from 5 subjects are shown in light colors. Average of all trials is shown in dark blue. This average shows in both cases a significant linear trend indicating that smaller bin widths will lead to better performance. *C*: to compare offline and online data, we use a third metric: integrated distance to target (failure rate and time to target cannot be calculated for offline data, as offline failure rate approaches 100% without online feedback, which is again telling of the inappropriateness of offline analysis). This metric is normalized by trial length (total trial time). *Right*: interleaved task variant. These show the same metrics as the *left*, reinforcing with another task variant these trends. In *A*−*C*, 95% confidence intervals (*A*: binomial distribution; *B* and *C*: Gaussian) are shown as error bars.

PN, and *VG*, who had run several thousand trials in this task on days prior to the data collection) and several naive subjects (*subjects MF*, *SK*, *CC*, *JH*, *KS*, and *RT*, who had not run the task before their full experiment). As no difference between these groups is apparent, we are also given confidence that there are not significant learning/training effects that need to be controlled in this data. Thus in summary, Fig. 4 tells us that shorter bin widths will lead to lower error rates and further that this is a highly robust effect in the human OPS.

In Fig. 5, we repeat the same presentation as in Fig. 4, but for the monkey subject instead of the human subjects. As discussed in METHODS, here there is only one subject who performed many full experiments (as opposed to the human subjects, who each did one full experiment). Accordingly, Fig. 5 shows faint lines corresponding to individual full experiments, and the dark blue corresponds to the average error across all experiments. Note also that there are now three columns: monkey BMI continuous, monkey BMI interleaved, and also monkey OPS interleaved (these choices discussed in METHODS). As in the human data, the findings are largely the same. In online prosthetic mode, across two task variants, BMI/OPS mode, and three error metrics, this data indicate that shorter integration bin widths will lead to higher prosthetic performance. As previously noted in the METHODS, we note that less will be made of absolute differences in performance between offline vs. online and OPS vs. BMI; rather, we are mostly interested in the optimal parameter settings that are suggested by different prosthetic modes.

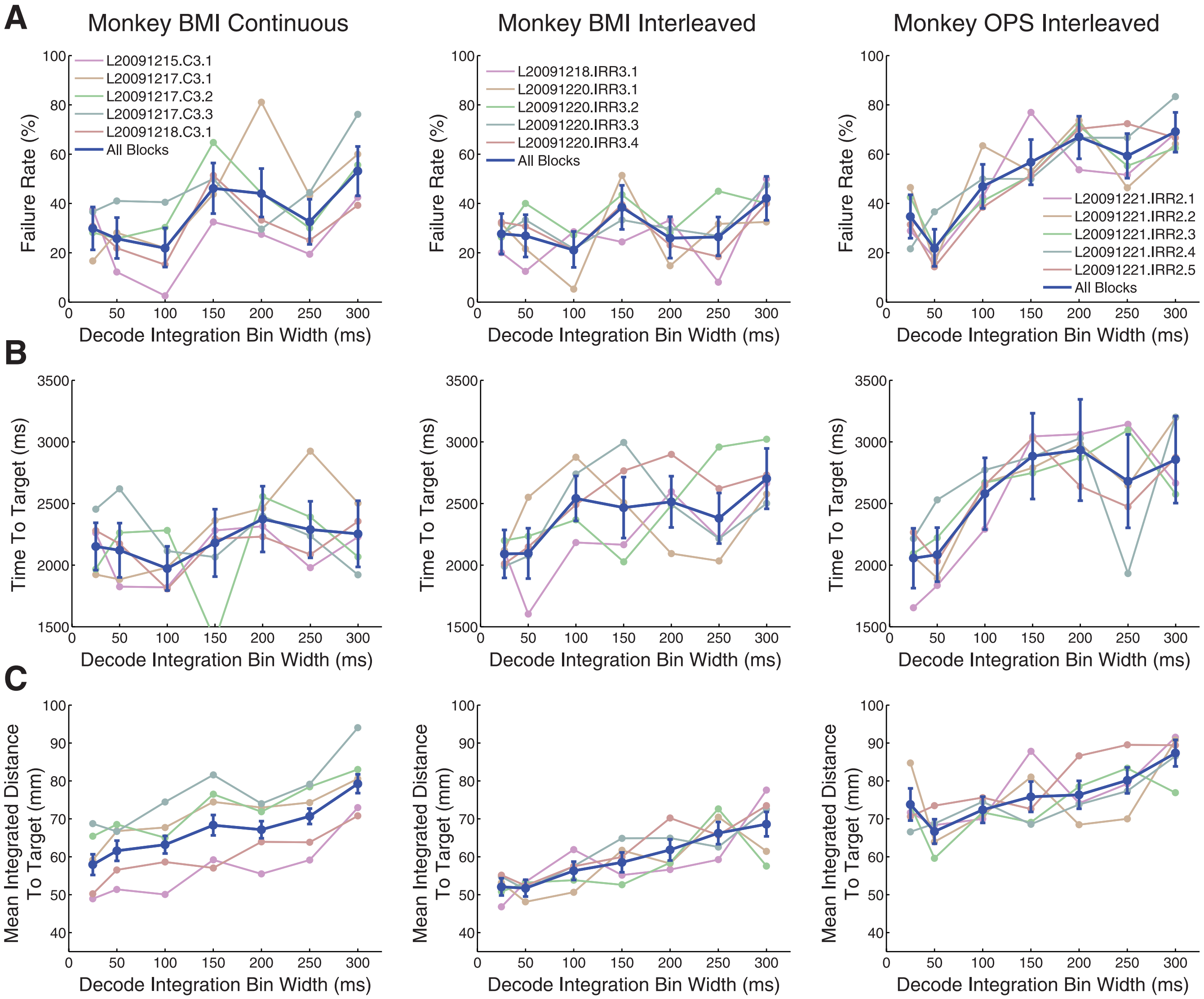

Fig. 5. Performance metrics for monkey BMI and human OPS decode trials. *Left*: BMI continuous task variant. *A*: failure rate, the percentage of trials where the subject did not successfully acquire and hold the reach target. *B*: time to target, the time required to reach the target for successful trials. In *A* and *B*, data from 5 subjects are shown in light colors. Average of all trials is shown in dark blue. This average shows in both cases a significant linear trend indicating that smaller bin widths will lead to better performance. *C*: to compare offline and online data, we use a third metric: integrated distance to target (failure rate and time to target cannot be calculated for offline data, as offline failure rate approaches 100% without online feedback, which is again telling of the inappropriateness of offline analysis). This metric is normalized by trial length (total trial time). *Middle*: BMI interleaved task variant. These show the same metrics as the *left*, reinforcing with another task variant these trends. *Right*: OPS interleaved task variant. The monkey can also run the OPS task with synthetic neural data. *Right* shows the same metrics as *left* and *middle*, and these metrics all show similar trends. By comparing BMI and OPS within subject on the same task (*middle* and *right*), we have an indication that the OPS is providing a valuable proxy to real neural prosthetic systems. In *A*−*C*, 95% confidence intervals (*A*: binomial distribution; *B* and *C*: Gaussian) are shown as error bars.

By performing a linear regression to the data in Fig. 5, *A*−*C* (binomial noise model for failure rates in panels Fig. 5*A*; gaussian noise model for other error metrics in Fig. 5, *B* and *C*), we can calculate the confidence level that each panel is indeed a positive sloping line (i.e., shorter is better). For the monkey BMI continuous, those values are as follows: *A*: $P < 10-4$; *B*: $P = 0.07$; *C*: $P < 10-4$. For the monkey BMI interleaved, those values are *A*: $P < 0.01$; *B*: $P < 10-3$; *C*: $P < 10-4$. For the monkey OPS interleaved, those values are as follows *A*: $P < 10-4$; *B*: $P < 10-4$; *C*: $P < 10-4$. Thus with high confidence we can say that the monkey BMI and OPS, like the human OPS, indicate that shorter integration bin widths will lead to higher online performance.

Comparing online analysis to offline analysis. With the several different variants of online prosthetic tasks (OPS vs. BMI, monkey vs. human, continuous vs. interleaved), and with the real reach trials that each subject performed as part of the experiment, we now demonstrate the difference between offline and online analysis of neural prosthetic systems. We noted in the previous section that the online data indicate strongly that shorter bin widths will lead to lower error. As a reminder, Figs. 4*C* and 5*C* show an error metric that can be calculated for both offline and online data. Accordingly, we took the real reaches from each subject during these experiments, and we ran offline analysis as has been previously seen in much literature (Artemiadis et al. 2007; Brockwell et al. 2004; Brown et al. 1998; Carmena et al. 2005; Chestek et al. 2007; Eden et al. 2004; Gao et al. 2002; Hatsopoulos et al. 2004; Kim et al. 2006; Lebedev et al. 2005; Mulliken et al. 2008; Paninski et al. 2004; Serruya et al. 2002; Shakhnarovich et al. 2006; Shoham et al. 2005; Taylor et al. 2002; Wessberg et al. 2000; Wu et al. 2004, 2006; Wu and Hatsopoulos 2008; Yu et al. 2007). The offline analysis consists of taking the neural data from a real reaching trial and running the Kalman filter with a given bin width, thereby producing a decoded reach from the recorded neural data alone. Because these data are offline, the same neural activity can be decoded at each bin width; that is, only one block is needed for every seven that were needed for the online analysis.

In Fig. 6, we show this offline analysis in red. Again, *left* is the human OPS continuous task, and *right* is the OPS interleaved. For clarity of illustration, we only show the average data (not the individual experiments, which were shown as faint traces in Figs. 4 and 5). The dark red points and confidence intervals (95%) represent the distribution of the error at each bin width. In dark red, we fit a quadratic polynomial to the data. In faint red, we show the linear regression to that same data. In blue, we show in the same way the distribution of the online data analysis. Note that these points and confidence intervals are exactly the same as the points and intervals from Fig. 4*C* (without the connecting line segments). In dark blue, we show the linear regression to the data, and in faint blue we show the quadratic fit. In both columns, the dark blue line entirely obscures the quadratic fit (the quadratic term is nearly 0).

Again, a few salient features appear. First, online (blue) has consistently lower error than offline (red). This is not surprising, as the subjects' access to feedback information in the online case should unambiguously improve performance. Nonetheless, it is a sanity check for the data and error metric. Second, and more importantly, we see that the offline error analysis has a characteristic "U-shape" indicating a performance optimum at 100- to 150-ms bin widths, adding more evidence that offline and online analyses may not agree. We note that the U-shape in offline errors makes intuitive sense [and has been shown elsewhere (Wu et al. 2006)]: integrate too little neural data (short bin widths), and the estimate is swamped by noise; integrate too much neural data (long bin widths), and the system updates the decoded cursor too slowly. Some optimal tradeoff between these two extremes (i.e., a U-shape) seems reasonable. Contrasting these U-shapes to the blue lines in Fig. 6 (online, where shorter bin widths give lower error), the online case may also make intuitive sense: indeed shorter bin widths lead to a noisier estimate, but the critical presence of feedback control allows the user to compensate for that noise, resulting in lower error. On the other hand, longer bin widths still result in a slowly updating prosthesis, so this

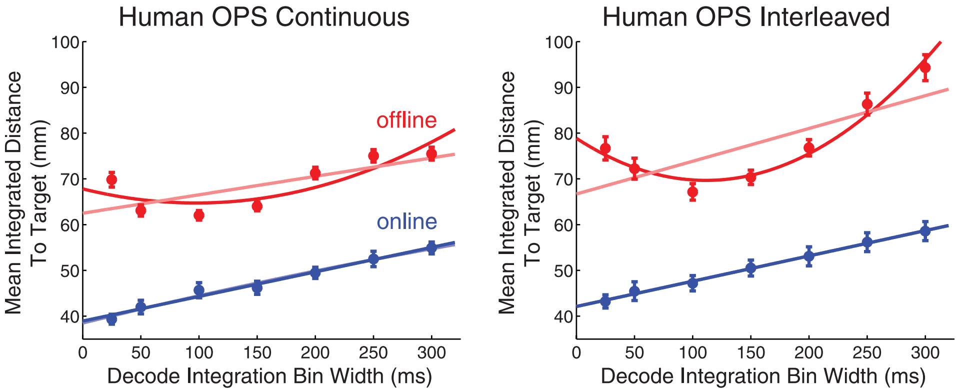

Fig. 6. Comparing offline analysis to online analysis in human OPS mode. *Left*: continuous task variant. Blue error bars are replicated from Fig. 4*C*. Dark blue line is a linear fit of that data, showing a significant performance trend indicating that shorter bin widths imply better performance. A light blue quadratic fit (not visibly or statistically different from a line) to the same data is obscured by the linear fit. In red, we perform the same analysis offline, using the real reaching sets from these users. We perform offline decodes at each bin width, allowing us to generate the same mean integrated distance to target metric. These error bars are the offline analogs to the blue error bars. Not surprisingly, offline analysis has worse performance than online, since there was no benefit of feedback. More significantly, however, is the shape of the data. This offline analysis suggests statistically significant performance optima of roughly 100–150 ms. This can be seen in the significant quadratic fit to the data (dark red). Linear fit (light red) does a clearly poorer job of fitting the data. Note that the characteristic "U-shape" of the offline data tells a very different story than the online analysis, indicating the important differences between these 2 testing paradigms. *Right*: interleaved task variant. Same analysis and implication holds true as in the *left*.

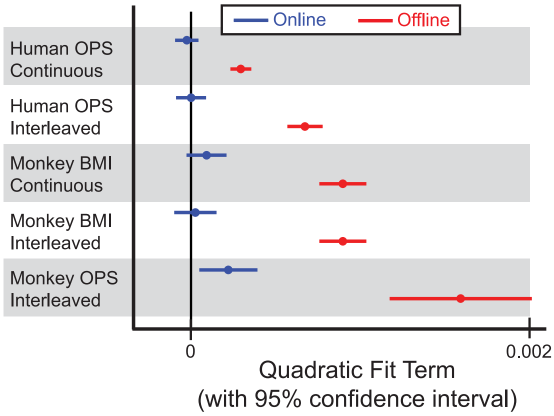

Fig. 8. Summary of online/offline differences; summary of OPS/BMI similarities. In the previous figures, we fit quadratic curves to both the offline and online performance data. Here, we plot those quadratic regression coefficients and their 95% confidence intervals. Any confidence interval overlapping 0 should be read as, "these data are not significantly different than a linear trend." This figure shows that nearly all online analyses in monkey and human do not have significant quadratic terms, agreeing with the implication from all previous data that shorter bin widths lead to better performance. Comparing that to the red error bars, again the central point is reiterated: online analysis and offline analysis (whether in real neural data or in synthetic neural data) tell very different stories. Furthermore, in addition to online and offline being very different, we see that OPS and BMI are in fact very similar. OPS gives a valuable means by which to sweep performance based on this algorithmic parameter.

slow "hopping" behavior makes feedback control more difficult and results in relatively higher error.

In Fig. 7, we repeat the presentation of Fig. 6 but for the monkey instead of humans. Offline analyses are shown in red, and online analyses are shown in blue. The raw data are represented with points and confidence intervals, and linear and quadratic fits are shown. In all online cases, the linear fit (dark blue) highly overlaps the quadratic fit, although the quadratic can just be seen in the rightmost column. Again, the monkey results are consistent with the human results. Online analysis in blue produces linear fits suggesting that shorter bin widths produce lower error, whereas the offline analyses produce characteristic U-shapes indicating a performance optimum around a bin width of 150–200 ms. Note that the monkey BMI continuous and BMI interleaved offline data (red data at *left* and *middle*) are identical: in real reaching trials, there is no difference between continuous and interleaved cases, as all reaches are real. Accordingly, we used the same reaching data for the *left* and *center* panels of Fig. 7. Another interesting note is that, even disregarding the U-shapes, the linear fits to offline data are inconsistent: human offline linear analysis says that shorter bin widths are better (red data in Fig. 6), but monkey offline linear analysis says that performance either is insensitive to bin width or improves with larger bin widths (red data in Fig. 7). Contrast that to the simple and consistent story that is seen across all online (blue) data in Figs. 6 and 7 (and so too Figs. 4 and 5): shorter bin widths will improve prosthetic performance. Thus these data strongly suggest that offline analysis gives an inconsistent and different answer than online analysis. Since the eventual user mode of neural prostheses is fundamentally online, this first finding calls into question the validity of offline analysis in informing prosthetic design choices.

Comparing OPS to BMI. The previous section showed that offline analysis may often be a poor proxy to online use of a prosthetic system. This effect was apparent in both the OPS and BMI data across Figs. 6 and 7. Thus the OPS, alongside BMI mode, has already made clear the importance of feedback-control considerations in prosthetic design. While this first finding is of scientific value, it does not clarify that the OPS is a strong proxy to BMI mode. We investigate that question here.

Figure 8 demonstrates the similarity between OPS and BMI modes in an online setting. In previous analyses, we fit both lines and parabolas to the collected online and offline performance data. Some of those regressions appeared linear, others "U-shaped." By investigating the quadratic fit term, we can see which data support a U-shape conclusion and which support a linear. Again, this distinction is of critical importance because

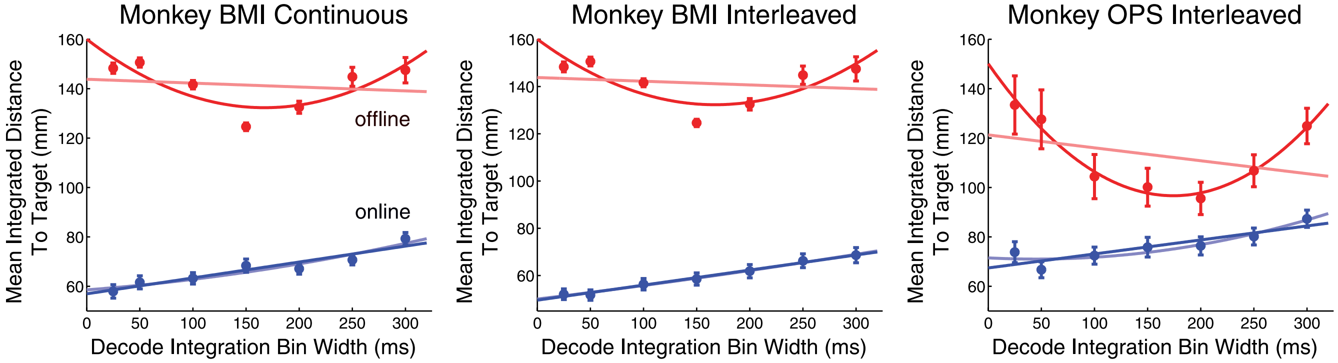

Fig. 7. Comparing offline analysis to online analysis in monkey BMI and monkey OPS modes. *Left*: BMI continuous task variant. Blue error bars are replicated from Fig. 4*C*. Dark blue line is a linear fit of that data, showing a significant performance trend indicating that shorter bin widths imply better performance. A light blue quadratic fit (not visibly or statistically different from a line) to the same data is obscured by the linear fit. In red, we perform the same analysis offline, using the real reaching sets from these users. We perform offline decodes at each bin width, allowing us to generate the same mean integrated distance to target metric. These error bars are the offline analogs to the blue error bars. Not surprisingly, offline analysis has worse performance than online, since there was no benefit of feedback. More significantly, however, is the shape of the data. This offline analysis suggests statistically significant performance optima of roughly 150–200 ms. This can be seen in the significant quadratic fit to the data (dark red). The linear fit (light red) does a clearly poorer job of fitting the data. Note that the characteristic "U-shape" of the offline data tells a very different story than the online analysis, indicating the important differences between these two testing paradigms. *Middle*: BMI interleaved task variant. Same analysis and implication holds true as in the *left*. *Right*: OPS interleaved rask variant. Again, the same analysis and implication holds true as in the *left*. Taken together, *left*, *middle*, and *right* show that both the BMI and OPS modes tell the same story: shorter bin widths imply better performance, whereas the offline analyses indicate an incorrect trend that leads to varying and misleading performance optima.

the shape of the performance curve makes fundamentally different statements about how to design a prosthetic system. We calculate the 95% confidence intervals on the quadratic regression coefficient for each of the 10 data sets, and we plot that data in Fig. 8. The blue points and intervals again represent the online data, and the red represent offline data. Each row indicates the prosthetic mode (human or monkey, OPS or BMI, continuous or interleaved). The gray and white stripes serve only to visually distinguish the rows. The black line denoting 0 allows us to draw the key conclusion from Fig. 8. If a confidence interval includes 0, then we cannot say with 95% confidence that the corresponding data is U-shaped; rather we conclude that shorter bin widths will indeed lead to better performance.

By considering only the blue data, we see importantly that all OPS and BMI modes indicate that shorter bin widths are indeed better. The consistency of these findings indicates that OPS and BMI modes are indeed valid proxies for one another. Only one case, the monkey OPS interleaved, does not have a confidence interval overlapping 0, although it does overlap the intervals of all other monkey data. In other words, that OPS quadratic fit term is not statistically significantly different from the BMI regression terms. Furthermore, if we instead plotted 99% confidence intervals, the interval of this monkey OPS interleaved case would indeed include 0. Thus this discrepancy is slight and of minor concern to the interpretation of the data. Figure 8 restates what was already visible in Figs. 6 and 7: blue online data across OPS and BMI modes agree.

In addition, this figure emphasizes the prior finding that offline analysis gives a different answer than online analysis: the red offline data are all U-shaped with 95% (and indeed 99%) confidence, but the blue online data do not support that claim, and only show that shorter bin widths are better. This figure also conveys another troubling aspect of offline analysis, namely the inconsistency of the findings across different subjects and prosthetic modes (i.e., the red intervals are quite varied amongst themselves).

This second finding suggests that the OPS may allow rapid and simpler testing of many algorithmic choices and may be a better proxy to clinical use than offline data analyses. In this work, we carefully tested that finding by gathering real BMI experimental data also, but in the future, careful experimental design with the OPS may allow many algorithms, parameters, and indeed prosthetic systems to be optimized without the risk and difficulty of implanted BMI subjects. Offering the OPS as an important part of the neural prosthesis design toolkit is the broad aim of this study. In the DISCUSSION, we discuss other uses of the OPS, and we give caveats about ways in which the OPS may not be a valuable proxy to BMI mode. These discussion points are critical to understanding the role of the OPS in future neural prosthesis research.

DISCUSSION

At a simple level, our results make three specific scientific points about optimizing the performance of a neural prosthetic system by choosing the bin width of a Kalman filter. The first point, seen in the raw data of Figs. 4 and 5, is that the bin width of the Kalman filter does indeed have meaningful performance implications; that is, it is a critical parameter that should be optimized in a neural prosthetic system. Second, the results show that offline analysis is a poor representative of online, closed-loop performance. Figures 6 and 7 show, across a variety of paradigms, that offline analysis indicates a very different trend in the data than the actual online usage mode of a prosthetic system. Third, we see that both the OPS and BMI suggest that shorter integration bin widths of 25–50 ms will lead to better performance, presumably reflecting the increased ability of the subject to incorporate feedback-control strategies.

At a deeper level, our results imply two fundamental findings about the design of neural prosthetic systems. First, we claim that feedback control is an essential consideration in the design of these systems, as demonstrated by the fact that offline analysis of algorithms is an inconsistent and problematic testing scenario. Across Figs. 6, 7, and 8, we see that offline analyses of various experimental paradigms indicate parabolic performance curves with error minima anywhere from 100–200 ms [and previous literature has found even higher optima, 200–300 ms (Wu et al. 2006), although the experimental paradigm and evaluation metrics were different in that study]. Furthermore, the curvature of these parabolas, in other words how costly it would be to have a suboptimal parameter setting, is inconsistent across paradigms, as seen in Fig. 8. Most troubling, however, is that these varied analyses all disagree fundamentally with online, closed-loop prosthetic use, which is in fact the true usage mode of these systems. All online analyses suggest that performance curves are linear and that shorter parameter settings are better, which stands in stark contrast to the parabolic offline implication. Thus the first finding of this work is to say that offline and online analyses are very different for neural prosthetic systems and further that offline is perhaps not adequately realistic to justify its continued use in the design of prostheses.

The second broad implication of this work is to present the OPS as a valuable proxy to a real neural prosthetic system. By replacing recorded neural activity with synthetic neural activity, the OPS enables analysis of human subjects using an online, closed-loop prosthetic system. By reproducing the qualitative behavior of real prosthesis use, we hypothesized that the OPS would produce similar performance curves as the real BMI mode. We tested this hypothesis by comparing humans and a monkey using the OPS to a monkey using a real BMI, and we found that all reproduce similar performance trends in the bin-width parameter sweep (Fig. 8 summarizes this effect). Thus in terms of this specific design question for decoding algorithms like the Kalman filter, the OPS appears to be a high-quality approximation to real online BMI use. However, a bigger question is the extent to which the OPS can be relied on to generalize to different neural prosthetic design questions in humans. As is portrayed in Fig. 1, having human results here (in addition to monkey results) is important to present a full human BMI simulator that does not require the resources of invasive human or monkey experiments. Human results are also important for generalizing the findings of the OPS to areas that will not necessarily be addressable in monkeys. The following points discuss in depth a number of ways in which the OPS will generalize well and a few ways in which it may not. These points of generalization are natural steps for future work as well.

OPS for general system design questions. With our specific finding that online and offline analyses are fundamentally different for the Kalman filter bin width, we have demonstrated that feedback control and subject interaction are significant contributors to eventual system performance. This general fact is a key principle of the field of human/computer interaction (Adler and Winograd 1992; Winograd and Flores 1987). However, the role of humans in human/neural-prosthesis interaction remains largely unexplored [although see the Introduction where we discuss the relevance of Danziger et al. (2009) to this question]. The OPS framework allows systematic study of this important question. Here we describe a few examples of general design questions that can be studied with the OPS.

First, user interface design for neural prostheses will require systematic study. For example, in this study, the subject reached to one of eight targets on a single ring. In a clinical application, where presumably the goal is to maximize information throughput or user experience, how many targets should be placed on the screen and in what configuration? Our previous work (Cunningham et al. 2008) addressed this point algorithmically but that was in an offline setting. Testing this online in the presence of feedback control is important for user interface design, but it may also be overly time consuming for a clinical trial subject. However, the OPS, insomuch as it involves a human in closed-loop and has similar qualitative and quantitative performance to a real BMI, can readily be used to investigate this and similar user interface questions.

Second, offline analyses do not explore the sensitivity of prosthesis users to noise (neural spiking variability, for example). Different systems and algorithms will cope with this noise in different ways, and it is critical to understand how these different algorithmic features are managed by human subjects. In an extreme case, even with an optimal decoder of neural activity, it is not clear at what level of noise a neural prosthetic device becomes unusable. For example, an important clinical question is, "how many recorded neurons are needed to control a prosthesis?" Such a question has major bearing on translational efforts. While offline studies often perform "neuron-dropping" analyses [e.g., Carmena et al. (2003); Li et al. (2009)] to test the sensitivity of error to neural population size, these studies cannot investigate the neuron count at which a user can no longer control the prosthesis in a satisfactory way. To our knowledge, only one study (Ganguly and Carmena 2009) has done a similar analysis online, and it suggests that dropping a meaningful percentage of a small number of stable neurons (that study recorded from 10 and 15 neurons) can have a highly detrimental effect on relative performance. For larger populations of neurons and different experimental contexts, however, this question remains unanswered. The OPS allows this analysis: we can vary the number of synthetic neurons used in the decode to find the point at which the human subject can no longer successfully complete the task. This and the above features cannot be tested in offline data, when the user cannot interact with the prosthesis. Furthermore, rigorously testing this feature in an online, real BMI may be quite difficult given the rarity of human clinical trials and the frustration that such a study would cause a monkey subject.

Third, in terms of the clinical utility of a BMI, the OPS allows customization of the controller based on the specific user application. For example, BMI signals demonstrated to date would have difficulty controlling a computer mouse on a normal operating system without the use of specialized accessibility software [see Hochberg et al. (2006)]. This software could involve nonlinear mouse acceleration curves (Jellinek and Card 1990) or multiple-level mouse selections [selecting once to zoom in and a second time to click (Kumar et al. 2007)]. Such software may be difficult to test in an animal model due to having many free parameters to sweep and abstract selection mechanisms that a monkey will not readily learn. Before using such as system in a human clinical trial, the OPS can serve as a test bench for developing these interfaces and testing them at a wide range of potential signal quality.

More generally, again by analogy to human/computer interaction, the design of neural prostheses will require systematic study. In addition, as the field moves towards clinically viable prostheses, the role of a human in a prosthetic system is increasingly important. Indeed, recent years have seen the increased importance of human studies for neural prostheses (Hochberg et al. 2006; Kim et al. 2008; Leuthardt et al. 2004). The OPS should help systematize this design process with a human in closed-loop control. We now discuss more detailed algorithmic features that the OPS can address.

OPS for specific algorithmic questions. Generating synthetic neural activity is the key enabler of the OPS, as it allows many human subjects to be tested without neural implants. However, generating synthetic neural data is also the principal concern with the OPS, as this choice calls into question the relevance of OPS findings for real BMI systems. In RESULTS, we demonstrated that OPS and BMI modes are consistent for optimizing the bin width of the Kalman filter, showing that indeed the OPS can be used as a valuable proxy. Nonetheless, for other algorithmic questions, there remains the risk that the resulting OPS effects will not translate well into a real neural system. To address this risk and to help ensure legitimacy of the result, we here identify four areas in which the OPS will continue to allow meaningful investigation of algorithmic design.

First, algorithmic models for arm reaching can be meaningfully studied with the OPS. The human motor plant imposes significant constraints on the frequency content, speed, and extent of a reach. For example, it is known that the motor system naturally produces smooth movements (Shadmehr and Wise 2005). This and other constraints have been well studied in human behavioral studies (Ghahramani and Wolpert 1997; Shadmehr andWise 2005; Wolpert and Ghahramani 2000; Wolpert et al. 1995), but they have been largely neglected in the design of neural prosthetic decode algorithms. Steps in this direction have been taken to acknowledge the point-to-point nature of reaches (Kulkarni and Paninski 2008; Mulliken et al. 2008; Srinivasan and Brown 2007; Srinivasan et al. 2007, 2006; Wu and Hatsopoulos 2008; Yu et al. 2007), but the field has not produced a model for arm reaching that will generalize to the eventual prosthesis user mode of unconstrained natural reaching. The Kalman filter used in this study stipulates a linear prior model for arm reaching (*Eq. 3*), but this model appears only in the decode algorithm, not in the generation of synthetic neural activity. Critically, the generation of synthetic neural data has no notion of a model of arm reaching. Accordingly, the OPS can be legitimately used to study the online effect of different algorithmic models for arm reaching, and performance improvements discovered here should port reliably to real BMI mode.

Second, the OPS can be used to study the algorithmic mappings from neural activity to kinematics. Many decode algorithms like the Kalman filter stipulate a linear mapping from neural activity to kinematic reach parameters (*Eq. 4*), but there are also many approaches that use nonlinear mappings. Understanding this algorithmic mapping is often considered of critical importance, but it has not been exhaustively studied online [although see Li et al. (2009)]. The OPS of this study generates synthetic neural data via a linear mapping (*Eq. 1*), so its current use for this question would be problematic, since the mapping from kinematics to neural activity is linear by assumption. However, future work can go further and use a source of synthetic neural signal that is not directly and simply related to kinematics. We think EMG offers a nice opportunity. EMG has been shown to have a linear relationship to neural activity (Santucci et al. 2005), so synthetic motor cortical neural activity can be reasonably generated from EMG. However, the connection between kinematics and EMG is by no means simple [Radhakrishnan et al. (2008), and particularly Tian and He (2003) discuss the difficulty of predicting kinematics from recorded EMG]. Thus by using EMG to create reasonable synthetic neural activity, we can guarantee a "nontrivial" relationship between the synthetic neural activity that we record and the endpoint kinematics that we want to decode. This approach would allow us to use the OPS to study the online implications of linear and nonlinear algorithmic mappings between kinematics and neural activity.

Third, we can evaluate very basic assumptions of particular decode algorithms. In this study, we considered the decode integration bin width of the Kalman filter. This time window specifies both the unit of time for output updates (changing the cursor position) and the unit of time for input integration (integration of neural activity to use as the decode signal). To remain specifically within the Kalman filter framework as has been done in the past, this agreement between update rate and integration rate is required. However, in practice this assumption can be relaxed. It may be that the difficulty with the large integration windows has more to do with the hopping or stroboscopic behavior of the cursor (the slow update rate) than with the long integration rate. Perhaps maintaining a long integration rate but making a quicker update rate (smoothly varying the cursor between two decoded positions, for example) could improve performance. Our preliminary investigation into this question suggests that further performance gains may be achievable by optimizing jointly the cursor-update-rate/neural-integration-rate pair. Since this question investigates human/neural-prosthesis interaction and not specific neural properties, the OPS is well suited to address this optimization directly.

Fourth, another basic assumption of a decode algorithm is the time lags it assumes between recorded neural activity and movement. This parameter is often optimized in BMI applications [e.g., Wu et al. (2006); Yu et al. (2007)]. However, much like the Kalman filter bin width of this study, time lags are rarely optimized online. Since time lags would be an assumption of the synthetic neural activity in the OPS, discovering the "true" online optimal time-lag is not likely to be an appropriate question for the OPS. However, understanding the sensitivity of this choice is indeed readily testable with the OPS. One can choose a set of time lags for the synthetic neural activity and then test performance sensitivity to different optimization strategies (for example, studies have included no time lags, a single time lag across all neurons, or individual lags per neuron). The OPS thus enables us to study how important time lags are to closed-loop performance.

These four examples (by no means the only examples) show that the OPS paradigm should continue to be valuable to validate algorithmic advances and translational needs.

OPS for neuroscientific questions. While the OPS may be viewed primarily as a platform for designing closed-loop neural prosthetic systems, it can also serve as a tool for investigating motor neurophysiology. Findings in the applied BMI context can often offer insight into the normal-functioning motor system. As a specific example, we found here that shorter temporal decode updates (bin widths) lead to smaller decode error. However, as we note above (the third example in *OPS for specific algorithmic questions*), future work should aim to understand the contribution to BMI decode errors between the stroboscopic effect of the cursor updates and the integration of shorter amounts of neural data. This stroboscopic effect represents a form of noise to the visual system. By altering that noise independently of other cursor control parameters, one could study the effects of noise on learning (Radhakrishnan et al. 2008), reflex adaptation in the motor system (Franklin and Wolpert 2008), or multimodal sensory integration (separating visual and proprioceptive contributions) (Graziano 1999; Sober and Sabes 2003; van Beers et al. 1999).

More generally, the OPS can serve as an experimental system for motor control studies, and it has interesting connections to that substantial literature [e.g., Shadmehr and Wise (2005); Wolpert and Ghahramani (2000)]. There has been a great deal of research using variants of a robotic manipulandum to study perturbations of normal motor control and to introduce visuomotor discrepancies [see Howard et al. (2009) for a review]. These studies ask questions ranging from dynamic learning in the motor system, to object manipulation, to limb stiffness measurements: a classic example is subjects making reaches in the presence of an external force field (Shadmehr and Mussa-Ivaldi 1994). These and other studies point to the fact that the introduction of distortions between intended movement and perceived movement has been a fundamental aspect of motor research for many years. The OPS is a similar system that allows novel 3-D visuomotor perturbations (in particular perturbations that are relevant to the applied field of neural prostheses) and manipulations of the system's underlying dynamics at several levels (at the cursor directly, or more implicitly in the neural encodings). Thus the OPS should be applicable to studies of motor control and 3-D motor adaptation. In addition to the novel perturbations this system may allow, the OPS may also prove useful as motor control researchers move towards more freely behaving experimental paradigms [as has been happening, for example, in motor electrophysiological studies (Santhanam et al. 2007)]. For example, due to electrode array shifts and head acceleration events, recorded neural signals can change abruptly (Santhanam et al. 2007). Studying 3-D motor adaptation to such signal changes in this less constrained context would be beneficial both scientifically in increasing our understanding of motor learning and in an applied setting by informing the field which aspects of motor control are contextually relevant for BMI. Certainly much caution should be taken in interpreting the neuroscientific implications of a BMI result, but the OPS should allow investigation of both applied neural prosthesis and fundamental motor neurophysiology questions.

Cautionary notes regarding the OPS. The results and our arguments for more systematic online analysis suggest that the

OPS should be used to address a variety of neural prosthetic design choices. However, a number of questions cannot readily be asked with the OPS, and much care should be taken with any simulation environment to ensure the legitimacy of any findings. Here we describe important precautions with the OPS.

As previously noted, the biggest potential pitfall of this system is the generation of synthetic neural data. Generating synthetic neural data or asking design questions carelessly can lead to tautologies that are uninteresting and potentially misleading. In this work, we specifically optimized the bin width of the Kalman filter. Since the synthetic neural activity is generated instantaneously and according to a Poisson likelihood model, the bin-width of the Kalman filter is not related to how the neural activity is generated. Thus the bin width optimization was more a question of how the subjects interacted with systems with different noise characteristics. This question could be and was legitimately investigated with the OPS, as indeed the real BMI experiments validated. However, with the same setup, one could propose another hypothesis that would prove problematic. Consider a hypothesis to test the plasticity of the mapping between neural activity and kinematics. Certainly, this is an interesting and important question both for scientific reasons and for the design of neural prosthetic systems [e.g., Ganguly and Carmena (2009)]. However, the answer here is trivial: because we synthetically created a static mapping between kinematics and neural activity, indeed this relationship is by definition without plasticity. Therefore, questions must be asked that do not directly test an assumption of the generative model for synthetic neural data. Although this point is perhaps fairly obvious, it is the overarching caveat with the extensibility of the OPS framework.

Another possible problem with the closed-loop paradigm regards proprioceptive feedback. As previously noted, there was a sensory discrepancy between the visual feedback and the real arm's proprioception (in prosthetic trials). Whether the true arm is restrained or allowed to reach (both are regularly done in the BMI literature), there is still a proprioceptive confound. Hence, this potential limitation is not specific to the OPS or this study, but rather to all able-bodied animal or human BMI studies. Indeed, previous BMI literature such as Carmena et al. (2003), Serruya et al. (2002), Taylor et al. (2002), and Velliste et al. (2008) have not reported difficulty with this proprioceptive error signal, and there is no reason to expect that it is any more prominent in the OPS than in those other able-bodied studies. Furthermore, previous neuroscience research has suggested that this confound is minor: Radhakrishnan et al. (2008) found that proprioception did not prevent task learning but did somewhat increase task difficulty. Other work has found that vision is a stronger sensory signal than proprioception in certain contexts [e.g., Touzalin-Chretien et al. (2010)], but this question is still debated in the neuroscience community. Thus while this possible limitation is important to bear in mind for the OPS and all able-bodied BMI studies, the significant progress that has been made in this field even with this proprioceptive confound suggests that the OPS framework should continue to be useful for aiding in the design of clinically relevant prostheses.

Summary. The OPS paradigm provides a middle ground between simple (but low reality) offline testing and more realistic (but very involved) clinical trials. We showed here how the OPS can be effectively used to find a surprising result that stands in contrast to offline analysis and previous literature: the bin width of the Kalman filter should be decreased in size for online prosthetic studies. These results showed the substantial disagreement between offline and online analyses, and the results showed the substantial similarity between OPS and BMI modes by using nonhuman prosthetic experiments as validation. While these findings are valuable in their own right, the broader message of the work is that offline analysis may be a poor proxy to eventual system use, and thus the field should investigate simulation opportunities to create a principled design engineering process around neural prosthetic systems. In this example, considering human interaction with the system was of critical importance, and we speculate that such feedback control is indeed critical for many other aspects of prosthetic system design. Ideally, the OPS or similar online simulators may become a piece of the neural prosthesis researcher's toolkit to allow rigorous design before moving to the gold standards of monkey BMI experiments and human clinical trials.

ACKNOWLEDGMENTS

We thank D. Franklin for helpful conversations, M. Risch for veterinary care, D. Haven for technical support, and S. Eisensee for administrative support.

GRANTS

This work was supported by UK EPSRC EP/H019472/1, NIH Director's Pioneer Award 1DP1OD006409, Burroughs Wellcome Fund Career Award in the Biomedical Sciences, Christopher and Dana Reeve Foundation, HHMI Fellowship, DARPA Revolutionizing Prosthetics Program Contract N66001-06-C-8005, McKnight Endowment Fund for Neuroscience, NDSEG Fellowship, NIH-CRCNS-R01, NSF Graduate Research Fellowships, Soros Fellowship, Stanford's CIS, Stanford Graduate Fellowship, and Medical Scholars Program.

DISCLOSURES

No conflicts of interest, financial or otherwise, are declared by the author(s).

REFERENCES

Adler P, Winograd T. *Usability: Turning Technologies into Tools*. New York: Oxford Press, 1992.

Anderson KD. Targeting recovery: priorities of the spinal cord injured population. *J Neurotrauma* 21: 1371–1383, 2004.

Artemiadis P, Shakhnarovich G, Vargas-Irwin C, Donoghue J, Black MJ. Decoding grasp aperture from motor-cortical population activity. In: *Proc IEEE EMBS*, 2007, p. 518–521.

Brockwell A, Rojas A, Kass R. Recursive Bayesian decoding of motor cortical signals by particle filtering. *J Neurophysiol* 91: 1899–1907, 2004.

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

Carmena J, Lebedev M, Crist R, O'Doherty J, Santucci D, Dimitrov D, Patil P, Henriquez C, Nicolelis M. Learning to control a brain-machine interface for reaching and grasping by primates. *PLoS Biol* 1: 193–208, 2003.

Carmena JM, Lebedev MA, Henriquez CS, Nicolelis MA. Stable ensemble performance with single-neuron variability during reaching movements in primates. *J Neurosci* 25: 10712–10716, 2005.

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

Chestek C, Batista A, Santhanam G, Yu B, Afshar A, Cunningham J, Gilja V, Ryu S, Churchland M, Shenoy K. Single-neuron stability during repeated reaching in macaque premotor cortex. *J Neurosci* 27: 10742–10750, 2007.

Chestek CA, Cunningham JP, Gilja V, Nuyujukian P, Ryu SI, Shenoy KV. Neural prosthetic systems: current problems and future directions. In: *Proc Ann Conf IEEE (EMBS)*, 2009.

Churchland MM, Cunningham JP, Kaufman MT, Ryu SI, Shenoy KV. Cortical preparatory activity: representation of movement or first cog in a dynamical machine? *Neuron* 68: 387–400, 2010.

Churchland MM, Shenoy KV. Temporal complexity and heterogeneity of single-neuron activity in premotor and motor cortex. *J Neurophysiol* 97: 4235–4257, 2007.

Churchland MM, Yu BM, Ryu SI, Santhanam G, Shenoy KV. Neural variability in premotor cortex provides a signature of motor preparation. *J Neurosci* 26: 3697–3712, 2006.

Cunningham JP, Gilja V, Ryu SI, Shenoy KV. Methods for estimating neural firing rates, and their application to brain-machine interfaces. *Neural Netw* 22: 1235–1246, 2009.

Cunningham JP, Yu BM, Gilja V, Ryu SI, Shenoy KV. Toward optimal target placement for neural prosthetic devices. *J Neurophysiol* 100: 3445–3457, 2008.

Danziger Z, Fishbach A, Mussa-Ivaldi, F. Learning algorithms for human-machine interfaces. *IEEE Trans Biomed Eng* 56: 1502–1511, 2009.

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

Douglas SA, Kirkpatrick AE, MacKenzie IS. Testing pointing device performance and user assessment with the iso9241, part 9 standard. In: *Proc ACM Conf Human Factors Comput Syst (CHI 99)*. New York: ACM, 1999, p. 215–222.

Eden U, Frank L, Barbieri R, Solo V, Brown EN. Dynamic analysis of neural encoding by point process adaptive filtering. *Neural Comp* 16: 971–998, 2004.

Franklin DW, Wolpert DM. Specificity of reflex adaptation for task-relevant variability. *J Neurosci* 28: 14165–14175, 2008

Fraser GW, Chase SM, Whitford A, Schwartz AB. Control of a brain-computer interface without spike sorting. *J Neural Eng* 6: 1–8, 2009.

Ganguly K, Carmena JM. Emergence of a stable cortical map for neuroprosthetic control. *PLoS Biol* 7: e1000153, 2009.

Gao Y, Black M, Bienenstock E, Shoham S, Donoghue J. Probabilistic inference of hand motion from neural activity in motor cortex. *Adv NIPS*, 14, 2002.

Georgopoulos A, Schwartz A, Kettner R. Neuronal population coding of movement direction. *Science* 233: 1416–1419, 1986.

Ghahramani Z, Wolpert D. Modular decomposition in visuomotor learning. *Nature* 386: 392–395, 1997.

Gilja V, Nuyujukian P, Chestek CA, Cunningham JP, Yu BM, Ryu SI, Shenoy, KV. High-performance continuous neural control enabled by a feedback control perspective. In: *Fronteirs in Systems Neuroscience. Conference Abstract: Computational and Systems Neuroscience (COSYNE)*, 2010.

Graziano MS. Where is my arm? the relative role of vision and proprioception in the neuronal representation of limb position. *Proc Natl Acad Sci USA* 96: 10418–10421, 1999.

Hatsopoulos N, Joshi J, O'Leary JG. Decoding continuous and discrete motor behaviors using motor and premotor cortical ensembles. *J Neurophysiol* 92: 1165–1174, 2004.

Herzfeld DJ, Beardsley SA. Improved multi-unit decoding at the brain-machine interface using population temporal linear filtering. *J Neural Eng* 7: 046012, 2010.

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

Howard IS, Ingram JN, Wolpert DM. A modular planar robotic manipulandum with end-point torque control. *J Neurosci Methods* 181: 199–211, 2009.

Jellinek HD, Card SK. Powermice and user performance. In: *Proc SIGCHI Conf Human Factors Comp Syst*, 1990, p. 213–220.

Kalman RE. A new approach to linear filtering and prediction problems. *J Basic Eng* 82: 35–45, 1960.

Kemere C, Shenoy K, Meng T. Model-based neural decoding of reaching movements: a maximum likelihood approach. *IEEE Trans Biomed Eng* 51: 925–932, 2004.

Kim S, Sanchez JC, Rao YN, Erdogmus D, Carmena JM, Lebedev MA, Nicolelis MAL, Principe JC. A comparison of optimal mimo linear and nonlinear models for brain-machine interfaces. *J Neural Eng* 3: 145–161, 2006.

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

Koyama S, Chase SM, Whitford AS, Velliste M, Schwartz AB, Kass RE. Comparison of brain-computer interface decoding algorithms in open-loop and closed-loop control. *J Comput Neurosci* 29: 73–87, 2010.

Kulkarni JE, Paninski L. State-space decoding of goal-directed movements. *IEEE Sig Proc Mag* 25: 78–86, 2008.

Kumar M, Paepcke A, Winograd T. Eyepoint: practical pointing and selection using gaze and keyboard. In: *Proc of SIGCHI Conf Human Factors Comp Syst*, 2007, p. 421–430.

Lebedev M, Carmena J, O'Doherty J, Zacksenhouse M, Henriquez C, Principe J, Nicolelis M. Cortical ensemble adaptation to represent velocity of an artificial actuator controlled by a brain-machine interface. *J Neurosci* 25: 4681–4693, 2005.

Lebedev MA, Nicolelis MA. Brain-machine interfaces: past, present, and future. *Trends Neurosci* 29: 536–546, 2006.

Leuthardt E, Schalk G, Wolpaw J, Ojemann J, Moran D. A brain-computer interface using electrocorticographic signals in humans. *J Neural Eng* 1: 63–71, 2004.

Li Z, O'Doherty JE, Hanson TL, Lebedev MA, Henriquez CS, Nicolelis MA. Unscented kalman filter for brain-machine interfaces. *PLos One* 4: e624363–e6243, 2009.

Marathe AR, Ward KE, Maryampillai M, Taylor DM. Are your movement decoding methods appropriate for online control of neuroprosthetics? *Soc Neurosci Abstr*, 2009.

Moran D, Schwartz A. Motor cortical representation of speed and direction during reaching. *J Neurophysiol* 82: 2676–2692, 1999.

Moritz CT, Perlmutter SI, Fetz EE. Direct control of paralysed muscles by cortical neurons. *Nature* 456: 639–642, 2008.

Mulliken GH, Musallam S, Andersen RA. Decoding trajectories from posterior parietal cortex ensembles. *J Neurosci* 28: 12913–12926, 2008.

Musallam S, Corneil B, Greger B, Scherberger H, Andersen R. Cognitive control signals for neural prosthetics. *Science,* 305: 258–262, 2004.

Paninski L, Shoham S, Fellows M, Hatsopoulos N, Donoghue J. Superlinear population encoding of dynamic hand trajectory in primary motor cortex. *J Neurosci* 24: 8551–8561, 2004.

Radhakrishnan SM, Baker SN, Jackson A. Learning a novel myoelectric-controlled interface task. *J Neurophysiol* 100: 2397–2408, 2008.

Ryu SI, Shenoy KV. Human cortical prostheses: lost in translation? *Neurosurg Focus* 27: E5, 2009.

Sanchez JC, Principe JC, Nishida T, Bashirullah R, Harris JG, Fortes J. Technology and signal processing for brain-machine interfaces: the need for beyond the state-of-the-art tools. *IEEE Sig Proc Mag* 25: 29–40, 2008.

Santhanam G, Linderman MD, Gilja V, Afshar A, Ryu SI, Meng TH, Shenoy KV. Hermesb: a continuous neural recording system for freely behaving primates. *IEEE Trans Biomed Eng* 54: 2037–2050, 2007.

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

Santhanam G, Sahani M, Ryu SI, Shenoy KV. An extensible infrastructure for fully automated spike sorting during online experiments. In: *Proc 26th Ann Int Conf IEEE EMBS*, 2004, p. 4380–4384.

Santucci DM, Kralik JD, Lebedev MA, Nicolelis MA. Frontal and parietal cortical ensembles predict single-trial muscle activity during reaching movements in primates. *Eur J Neurosci* 22: 1529–1540, 2005.

Schwartz AB. Cortical neural prosthetics. *Ann Rev Neurosci* 27: 487–507, 2004.

Schwartz AB, Cui XT, Weber DJ, Moran DW. Brain-controlled interfaces: movement restoration with neural prosthetics. *Neuron* 52: 205–220, 2006.

Serruya M, Hatsopoulos N, Paninski L, Fellows M, Donoghue J. Instant neural control of a movement signal. *Nature* 416: 141–142, 2002.

Shadmehr R, Mussa-Ivaldi FA. Adaptive representation of dynamics during learning of a motor task. *J Neurosci,* 14: 3208–3224, 1994.

Shadmehr R, Wise SP. *The Computational Neurobiology of Pointing and Reaching*. MIT Press, Cambridge, MA: MIT Press, 2005.

Shakhnarovich G, Kim SP, Black MJ. Nonlinear physically-based models for decoding motor-cortical population activity. In *Advances in NIPS 18*. Cambridge, MA: MIT Press, 2006.

Shenoy K, Meeker D, Cao S, Kureshi S, Pesaran B, Mitra P, Buneo C, Batista A, Burdick J, Andersen R. Neural prosthetic control signals from plan activity. *NeuroReport* 14: 591–596, 2003, 2003.

Shoham S, Paninski L, Fellows M, Hatsopoulos N, Donoghue J, Normann R. Statistical encoding model for a primary motor cortical brain-machine interface. *IEEE Trans Biomed Eng* 52: 1313–1322, 2005.

Sober SJ, Sabes PN. Multisensory integration during motor planning. *J Neurosci* 23: 6982–6992, 2003.

Srinivasan L, Brown EN. A state-space framework for movement control to dynamic goals through brain-driven interfaces. *IEEE Trans Biomed Eng* 54: 526–535, 2007.

Srinivasan L, Eden UT, Mitter SJ, Brown EN. General purpose filter design for neural prosthetic systems. *J Neurophysiol* 98: 2456–2475, 2007.

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

Taylor D, Tillery SI, Schwartz A. Direct cortical control of 3-D neuroprosthetic devices. *Science* 296: 1829–1832, 2002.

Tian J, He J. Can EMG machine interface be used to model brain machine interface? In: *Proc 25th Ann Int Conf IEEE EMBS*, 2003, p. 1658–1661.

Todorov E. Direct cortical control of muscle activation in voluntary arm movements: a model. *Nat Neurosci* 3: 391–398, 2000.

Touzalin-Chretien P, Ehrler S, Dufour A. Dominance of vision over proprioception on motor programming: evidence from ERP. *Cereb Cortex* 20: 2007–2016, 2010.

van Beers RJ, Sittig AC, Denier van der Gon JJ. Integration of proprioceptive and visual position-information: an experimentally supported model. *J Neurophysiol* 81: 1355–1364, 1999.

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

Ventura V. Spike train decoding without spike sorting. *Neural Comp* 20: 923–963, 2008.

Wessberg J, Stambaugh C, Kralik J, Beck P, Laubach M, Chapin J, Kim J, Biggs S, Srinivasan M, Nicolelis M. Real-time prediction of hand trajectory by ensembles of cortical neurons in primates. *Nature* 408: 361–365, 2000.

Wheatstone C. Contributions to the physiology of vision–part the first. on some remarkable, and hitherto unobserved, phenomena of binocular vision. *Philos Trans R Soc London* 128: 371–394, 1838.

Winograd T, Flores F. *Understanding Computers and Cognition: A New Foundation for Design*. Norwood, NJ: Addison-Wesley, 1987.

Wolpert D, Ghahramani Z. Computational principles of movement neuroscience. *Nat Neurosci* 3: 1212–1217, 2000.

Wolpert D, Ghahramani Z, Jordan M. An internal model for sensorimotor integration. *Science* 269: 1880–1882, 1995.

Wu W, Black M, Gao Y, Bienenstock E, Serruya M, Donoghue J. Inferring hand motion from multi-cell recordings in motor cortex using a Kalman filter. In: *SAB'02-Workshop on Motor Control in Humans and Robots: On the Interplay of Real Brains and Artificial Devices*, 2002, p. 66–73.

Wu W, Black M, Mumford D, Gao Y, Bienenstock E, Donoghue J. Modeling and decoding motor cortical activity using a switching Kalman filter. *IEEE Trans Biomed Eng* 51: 933–942, 2004.

Wu W, Gao Y, Bienenstock E, Donoghue J, Black M. Bayesian population decoding of motor cortical activity using a Kalman filter. *Neural Comp* 18: 80–118, 2006.

Wu W, Hatsopoulos NG. Target-included model and hybrid decoding of stereotyped hand movement in the motor cortex. In: *Intl Conf Biomed Robotics Biomech, 2008 (IEEE RAS/EMBS)*, 2008, p. 55–60.

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

Zar JH. *Biostatistical Analysis*. Englewood Cliffs, NJ: Prentice Hall, 1999.