Robust neural control of virtual locomotion enabled by a novel decoding strategy

Karen E Schroeder* [1,2], Sean M Perkins* [2,3], Qi Wang[3], & Mark M Churchland[1,2,4,5, †]

* Equal contribution

[1] Department of Neuroscience, Columbia University Medical Center, New York, NY
[2] Zuckerman Institute, Columbia University, New York, NY
[3] Department of Biomedical Engineering, Columbia University, New York, NY
[4] Kavli Institute for Brain Science, Columbia University Medical Center, New York, NY
[5] Grossman Center for the Statistics of Mind, Columbia University, New York, NY

[†] Correspondence to: mc3502@columbia.edu

Abstract

Brain-machine interfaces (BMIs) for reaching have enjoyed continued performance improvements. Yet there remains significant need for locomotor BMIs (e.g., for wheelchair control), which could potentially benefit a much larger patient population. Fewer studies have addressed this need, and the most effective approach remains undetermined. Here, we develop a locomotor BMI based on cortical activity as monkeys cycle a hand-held pedal to progress along a virtual track. Unlike most reach-based BMIs, we did not directly map neural states to commanded velocity or position. Instead, we leveraged features of the neural population response that were robust during rhythmic cycling. These included an overall shift in neural state when moving, and rotational trajectories with direction-specific paths. We used nonlinear means to infer kinematics from these features. Online BMI-control success rates approached those during manual control. Our results illustrate that different use-cases can require very different approaches to guiding a prosthetic via neural activity.

Introduction

Brain-machine interfaces (BMIs) interpret neural activity and provide control signals to external devices such as computers and prosthetic limbs. Intracortical BMIs for reach-like tasks have proved successful in primates and human clinical trials[1–8]. More widespread use appears imminent. Yet at the same time, there exist non-reach-like movements whose restoration is valuable to patients. For example, many patients could benefit from a BMI that controls locomotion through their environment (e.g., movement of a wheelchair). Recent work has demonstrated that this is feasible[9,10]. While locomotor BMIs can be guided by reach-inspired decoding approaches, other viable strategies exist and remain unexplored. For example, it may be more natural or robust to decode locomotor commands from continuous rhythmic neural activity. The specific case of locomotor BMIs highlights a broader issue: a focus on reaching and grasping has produced state-of-the-art decode algorithms that may not generalize well to non-reaching applications.

The early success of reach-based BMIs[11–16] was built upon a decoding strategy that sought to invert the ostensible cortical encoding of kinematic variables such as hand velocity / direction. Multiple lines of evidence now argue against the hypothesis that activity in motor cortex literally encodes kinematic variables[17–22]. Yet within the confines of a given task, robust correlations between neural activity and reach direction allow excellent BMI performance[11–16]. This core strategy has thus endured even as decoder sophistication has increased; improvements have derived largely from better estimating the neural state, thus improving the reach-velocity decode[1,23–26]. Indeed, this strategy is employed even when the hypothesis of literal kinematic encoding is explicitly rejected[23]. Such decoders are thus 'opportunistic': they seek to optimally leverage robust relationships between neural activity and behavior, regardless of whether those relationships are fundamental.

Opportunistic decode strategies possess two large advantages over biomimetic strategies (defined as strategies that attempt to decode the true output signals). First, opportunistic strategies do not require knowledge of the true relationship between neural activity and motor output, only an accurate characterization of activity for the behaviors one wishes to decode. Second, opportunistic strategies are potentially more noise robust than a truly biomimetic strategy. We recently argued that the dominant signals in motor cortex – i.e., those with the greatest influence on firing rates – exist to ensure noise-robust dynamics and do not encode any external quantity[18]. In contrast, neural signals that relate directly to outgoing commands (e.g., downstream muscle activity) are small. Opportunistic strategies can leverage the dominant signals while biomimetic strategies are (by definition) limited to the smaller output signals. Yet

opportunistic decoding carries an obvious disadvantage: different tasks may necessitate very different decode strategies[27].

Here we explore strategies appropriate for decoding virtual motion during a task in which monkeys rotate a hand-held pedal to move along a virtual track. Rather than emulate natural locomotion, this cycling task provides a view of cortical activity during a rhythmic voluntary movement. The resulting rhythmic neural responses represent a class of activity with which a patient might wish to control prosthetic locomotion. In agreement with recent results, neural population activity during cycling differed from that during reaching in fundamental ways. It was not simply that activity was sustained rather than transient. More critically, 'directional' signals related to kinematics (hand velocity or position) were only weakly reflected in neural firing rates. Signals related to muscle activity were of similarly small magnitude.

Thus, rather than directly map neural activity to decoded limb kinematics, we sought to leverage features of the neural response that had robust (but not necessarily linear) relationships with the variable we most wished to decode: self-motion through the environment. We identified three such features. The first was a translation of the neural state that rapidly and robustly indicated whether the monkey was moving. The second was an elliptical neural-trajectory while cycling. That trajectory did not reverse with cycling direction (as would a representation of hand velocity), but instead occupied direction-dependent dimensions. As a result, the direction and magnitude of self-motion could be estimated by comparing the angular momentum of the neural state between pairs of dimensions. This feature was robust during ongoing cycling but not at movement initiation. We thus leveraged a third feature: at movement initiation, neural activity corresponding to forward and backward cycling was briefly linearly separable.

A decoder that leveraged these dominant features provided excellent online control of virtual locomotion. Success rates and acquisition times were very close to those achieved under manual control. Almost no training or adaptation time was needed; the low-latency and accuracy of the decoder were such that monkeys appeared to barely notice transitions from manual control to BMI control. These results demonstrate the feasibility of BMI locomotion based on rhythmic neural activity. More broadly, they establish that opportunistic decode strategies can work well in non-reach-based scenarios, but that new applications require novel decode approaches that respect the dominant structure of neural activity.

Results

Behavior

We trained two monkeys (G and E) to rotate a hand-held pedal to move through a virtual environment (**Fig. 1**). All motion was along a linear track – no steering was necessary. Consistent with this, a single pedal was cycled with the right arm only. Our goal when decoding was to reconstruct the virtual motion produced by that single pedal. On each trial, a target appeared in the distance. To acquire that target, monkeys produced virtual velocity in proportion to the rotational velocity of the pedal. The color of the environment (lush and green versus desert-like and tan) instructed cycling direction. When the environment was green (**Fig. 1a**, *left*) forward virtual motion was produced by cycling 'forward' (i.e., with the hand moving away from the body at the top of the cycle). When the environment was tan (**Fig. 1a**, *right*) forward virtual motion was produced by cycling 'backward' (the hand moving toward the body at the top of the cycle). Cycling in the wrong direction produced motion away from the target. Trials were presented in blocks of forward or backward trials. Within each block, targets were separated by a randomized distance of 2, 4 or 7 cycles. Acquisition of a target was achieved by stopping and remaining stationary 'on top' of the virtual target for a specified time. Reward was then given and the next target appeared.

Monkeys performed the task well, moving swiftly between targets, stopping accurately on each target, and remaining stationary until the next target was shown. Monkeys cycled at a pace that yielded nearly linear progress through the virtual environment (**Fig. 1b**). Although not instructed to cycle at any particular angular velocity, monkeys adopted a brisk ~2 Hz rhythm (**Fig. 1c**). Small ripples in angular velocity were present during steady-state cycling; when cycling with one hand it is natural for velocity to increase on the downstroke and decrease on the upstroke. Success rates were high, exceeding 95% in every session (failures typically involved over- or under-shooting the target location). This excellent performance under manual control provides a stringent bar by which to judge performance under BMI control.

BMI control was introduced after monkeys were adept at performing the task under manual control. Task structure and the parameters for success were unchanged under BMI control, and no cue was given regarding the change from manual to BMI control. The switch to BMI control was made at the beginning of the session, after completion of a block of manual-control trials (25 forward and 25 backward 7-cycle trials). These manual-control trials were used to train the decoder. The switch was then made to BMI control for the remainder of the session. For monkey G, we occasionally included blocks of manual-control trials later in the session to allow

comparison between BMI and manual performance. For Monkey E we used separate (interleaved) sessions to assess manual-control performance.

During both BMI control and manual control, the monkey's ipsilateral (non-cycling) arm was restrained. The contralateral (cycling) arm was never restrained. We intentionally did not dissuade the monkey from continuing to physically cycle during BMI control. Indeed, our goal was that the transition to BMI control would be sufficiently seamless to be unnoticed by the monkey, such that he would still believe that he was in manual control. An advantage of this strategy is that we are decoding neural activity when the subject attempts to actually move, as a patient presumably would. Had we insisted the arm remain stationary, monkeys would have needed to actively avoid patterns of neural activity that drive movement – something a patient would not have to do. Allowing the monkey to continue to move normally allowed us to extensively quantify the performance of our decoder by comparing decoded with intended (i.e., actual) movement. This is often not possible when using other designs. For example, in Rajangam et. al.[9], performance could only be assessed via indirect measures (such as time to target) because what the monkey was actually intending to do at each moment was unclear. We considered these advantages to outweigh a potential concern: a decoder could potentially 'cheat' by primarily leveraging activity driven by proprioceptive feedback (which would not be present in a paralyzed patient). This is unlikely to be a large concern. Recordings were made from motor cortex, where robust neural responses precede movement onset. Furthermore, we have documented that motor cortex population activity during cycling is quite different from that within the proprioceptive region of primary somatosensory cortex[18]. Thus, while proprioceptive activity is certainly present in motor cortex[28–31] (especially during perturbations[32]) the dominant features of M1 activity, described below, are unlikely to be primarily proprioceptive.

Given our use of healthy animals, we stress that the goal of the present study is to determine how the dominant structure of neural activity can be leveraged for accurate prosthetic decode. This follows the successful strategy of BMI studies that leveraged the known structure of activity during reaching[11,23]. Of course, the nature of the training data used to specify decode parameters (e.g., the neural dimensions to be used) will necessarily be different for a healthy animal that cannot understand verbal instructions and an impaired human that can. We thus stress that our goal is to determine a robust and successful decode strategy that works in real time during closed-loop performance. We do not attempt to determine the best approach to parameter specification, which in a patient would necessarily involve intended or imagined movement.

Neural activity and decoding strategy

We recorded motor cortical activity using 96-channel Utah arrays. For monkey G, one array was implanted in primary motor cortex (M1) and a second in dorsal premotor cortex (PMd). For monkey E, a single array was implanted in M1. For each channel we recorded times when the voltage crossed a threshold. Threshold crossings typically reflected individual spikes from a small handful of neurons (a neural 'unit'). Spikes from individual neurons could be clearly seen on many channels, but no attempt was made to spike-sort, as the benefit of doing so is typically modest when controlling a prosthetic device[33]. Unit activity was strongly modulated during cycling (**Fig. 1d**). The phase, magnitude, and temporal pattern of activity depended on whether cycling was forward (*green* traces) or backward (*red* traces). A key question is how these unit-level features translate into population-level features that might be leveraged to estimate intended motion through the virtual environment.

In traditional decoding approaches (**Fig. 2a**, *top*) neural activity is hypothesized (usefully if not literally) to encode kinematic signals, which can be decoded by inverting the encoding scheme. Although nonlinear methods (including variations of Kalman filtering) are often used to estimate the neural state, the final conversion to a kinematic command is typically linear or roughly so. To explore kinematic encoding in the present task, we used linear regression to identify neural dimensions where activity correlated well with kinematics (including hand velocity and position). Regression was performed using single trials. Use of single trials provides a large quantity of training data and is implicitly regularizing: regression must find signals that are robust in the face of single-trial spiking variability. The regression weights for a given kinematic parameter define a neural dimension where activity correlates strongly with that parameter. We computed the neural variance captured by each such dimension. Variance captured was computed using trial-averaged data, to ensure that values were not diluted by noise. Despite this, the neural dimensions that best captured kinematic signals captured little population response variance (**Fig. 2b**, green bars). This was also true of neural dimensions that captured muscle activity (**Fig. 2b**, yellow bar). This was initially surprising: single-neuron responses were robustly sinusoidally modulated, as were many kinematic variables. Yet sinusoidal response features were often superimposed upon other response features (e.g., overall shifts in rate when moving versus not moving). Sinusoidal features also did not display phase relationships, across forward and backward cycling, that were consistent with kinematic encoding[18]. As a result, the dimensions where activity correlated strongly with kinematics captured relatively little response variance.

Low-variance signals are a poor candidate for decoding intended action; they are likely to be non-robust with respect to multiple challenges. Some of these challenges (e.g., spiking noise) can be anticipated and estimated in advance, but others cannot. They include recording instabilities, changes in strategy or behavior with time, and (outside the laboratory setting) external sources of noise or variability. Given these challenges, it is worth stressing that there were two practical reasons why reach-focused BMIs leveraged signals that correlate with hand velocity. First, such signals are high-variance during reaching – so much so that M1 responses have often been summarized in terms of a preferred direction[34,35]. Second, movement direction / velocity are the variables one wishes to decode during reaching. In the present case neither motivation holds. Signals related to hand kinematics are low-variance, and we most wish to decode self-motion through the virtual environment. This suggests an alternative strategy (**Fig. 2a**, *bottom*): identifying neural response features that are both robust (high-variance) and relate reliably to the presence and direction of self-motion.

To pursue this strategy, we considered three sets of high-variance dimensions. The first set included four 'rotational dimensions' (two each for forward and backward cycling) which captured elliptical trajectories present during steady-state cycling[18]. The second set included a single 'moving-sensitive' dimension, in which the neural state distinguished whether the monkey was stopped or moving regardless of movement direction[36]. The third set was a triplet of 'initial-direction' dimensions. In these dimensions, cycling direction could be transiently but readily distinguished in the moments after cycling began.

In subsequent sections we document the specific features present in these high-variance dimensions. Here we concentrate on the finding that the space spanned by these eight dimensions captured 70.9% ± 2.3% of the firing-rate variance (**Fig. 2c**). This was only modestly less than that captured by the top eight PCs (which capture the most variance possible), and much more than that captured by spaces spanned by dimensions where activity correlated with kinematics and/or muscle activity (**Fig. 2b**). We thus based our BMI decode entirely on activity in these eight high-variance dimensions. Before describing how this was accomplished, we document the resulting performance.

Performance

Monkeys performed the task very well under closed-loop BMI control (**Fig. 3** and **Supp. Movie 1**). Monkeys continued to cycle as normal, presumably not realizing that the pedal had been disconnected from the control system. The illusion that the pedal still controlled the task was supported by a high similarity between decoded virtual velocity and intended virtual velocity (i.e., what would have been produced by the pedal were it still controlling the task). The cross-

correlation between these peaked at 0.93 ± .02 and 0.81 ± .03 (monkey G and E, mean ± SD) at a short lag: 76 ± 4 ms and 102 ± 7 ms (**Fig. 3a**). That illusion was also aided by a low rate of false starts; it was exceedingly rare for decoded motion to be non-zero when the monkey was attempting to remain stationary on top of a target. False starts occurred on 0.29% and 0.09% of trials (monkeys G and E), yielding an average of 1.9 and 0.12 occurrences per day. This is notable because combatting unintended movement is a key challenge for BMI decoding[2,37,38]. The above features – high correlation with intended movement, low latency, and few false starts – led to near-normal performance under BMI control (**Fig. 3b,c**). Success rates under BMI control (**Fig. 3d**, *magenta symbols*) were almost as high as under manual control (*open symbols*), and the time to move from target to target was only slightly longer under BMI control (**Fig. 3e**).

The only respect in which BMI control suffered noticeably was accuracy in stopping on the middle of the target. Under manual control, monkeys stopped very close to the target center (**Fig. 3f,** *gray histogram*), which always corresponded to the 'pedal-straight-down' position. Stopping was less accurate under BMI control (*magenta histogram*). This was partly due to the fact that because virtual motion was swift, even small errors in decoded stopping time become relevant: e.g., a 100 ms error corresponds to ~0.2 cycles of physical motion. The average standard deviation of decoded stopping time (relative to actual stopping time) was 133 (monkey G) and 99 ms (monkey E). Increased stopping error in BMI-control trials was also due to an incidental advantage of manual control: the target center was aligned with the pedal-straight-down position, a fact which monkeys leveraged to stop very accurately in that position. This strategy was not available during BMI control because the correct time to stop rarely aligned perfectly with the pedal-straight-down position (this occurred only if decoded and intended virtual velocity matched perfectly when averaged across the cycling bout).

Performance was overall modestly better for monkey G versus E. This was likely due to the implantation of two arrays rather than one. Work ethic may also have been a factor; monkey E performed fewer trials under both BMI and manual control. Still, both monkeys could use the BMI successfully starting on the first day, with success rates of 0.87 and 0.74 (monkey G and E). Monkey G's performance rapidly approached his manual-control success rate within a few sessions. Monkey E's performance also improved quickly, although his manual-control and BMI-control success rates were mostly lower than Monkey G's. The last five sessions involved BMI success rates of 0.97 and 0.96 for the two monkeys. This compares favorably with the overall averages of 0.98 and 0.95 under manual control. Although this performance improvement with time may relate to adaptation, the more likely explanation is simply that monkeys learned to not be annoyed or discouraged by the small differences in decoded and intended velocity.

State machine

The performance documented above was achieved using a state-dependent decode (**Fig. 4**). Features of the neural activity (described more fully in subsequent sections) determined state transitions and what was decoded in each state. Briefly, state transitions were governed by activity in the moving-sensitive dimension, which was translated into a probability of moving, p_{move}. If p_{move} was low, the STOP state was active and decoded virtual velocity was zero. When p_{move} became high, the INIT state was entered but decoded velocity remained zero. After 175 ms, the EARLY state was entered and velocity was decoded using the initial-direction dimensions. After an additional 200 ms, the STEADY state was entered and virtual velocity depended only on the neural state in the rotational dimensions. Decoded velocity was filtered to smooth fluctuations during STEADY.

Values of $p_{move} < 0.1$ always produced a transition back to STOP. This typically occurred from STEADY to STOP, as the movement was successfully ending. However, it could also occur from the other two states. This was especially helpful if p_{move} became high very briefly (and presumably erroneously). In such cases the state could transition from INIT back to STOP with the decoded velocity never departing from zero. Below we describe how virtual velocity was estimated while in STEADY, how p_{move} was derived, and how we decoded the early direction of movement during EARLY.

Direction of steady-state movement inferred from rotational structure

The dominant feature of the neural response during steady-state cycling was a repeating elliptical trajectory[18]. Our decoder leveraged the fact that forward-cycling and backward-cycling trajectories occurred in non-identical dimensions. We employed an optimization procedure to find a two-dimensional 'forward plane' that maximized the size of the forward trajectory relative to the backward trajectory. We similarly found an analogous 'backward plane'. These planes were identified based on trial-averaged responses from the 50 trials of training data collected under manual control (**Fig. 5a**). With the aid of filtering (*Methods*), these planes continued to capture rotational features on individual trials (**Fig. 5b**). Although forward and backward trajectories were not orthogonal to one another, the above procedure was still able to find planes where strongly elliptical trajectories were present for only one cycling direction.

A common strategy for reaching prostheses is to linearly transform the neural state into a hand-velocity command; e.g. a state consistently to the right of zero would result in a consistently high rightwards velocity. In a given plane (e.g., backwards) the neural state traces a circle, and a plot of horizontal versus vertical hand velocity would also trace a circle. However, it would not

be desirable to attempt to directly decode velocity. Not only would this require somehow choosing between planes, but a neural state consistently to the right of zero should not result in a consistent decode of rightwards hand velocity. A decode of hand position would be somewhat more natural but still awkward (there are four dimensions rather than two, and positions near zero are difficult to interpret). We thus chose the strategy of comparing angular momentum (the cross product of the state vector with its derivative) between the two planes. When moving backward (first three cycling bouts in **Fig. 5c**) angular momentum was sizeable in the backward plane (*dark blue*) but not the forward plane (*bright blue*). The opposite was true when moving forward (subsequent three bouts).

Based on training data, we considered the joint distribution of forward-plane and backward-plane angular momentum. We computed distributions when stopped (**Fig. 5d**, *orange*), when cycling forward (*green*) and when cycling backward (*red*). These distributions overlapped little, and we fit a Gaussian to each. During BMI control, we computed the likelihood of the observed angular momentums under each of the three distributions. If likelihood under the stopped distribution was high, decoded velocity was zero. Otherwise, decoded velocity was determined by the relative likelihoods under the forward and backward distributions. These likelihoods were converted into a virtual velocity that was maximal when one likelihood was much higher (which was typically the case) and slower when likelihoods were more similar. The maximum decoded virtual velocity was set to approximate the typical virtual velocity under manual control, when cycling at ~2 Hz.

The above steps were performed when in the STEADY state. Distributions of decoded velocity under BMI control (**Fig. 5e**, *bottom*) were similar to the distributions of velocity that would have resulted were the pedal still operative (**Fig. 5e**, *top*). Importantly, distributions overlapped very little; the direction of decoded motion was almost always correct. Decoded velocity was near maximal at most times, especially for monkey G. High accuracy and brisk velocities were responsible for the ability to move between targets almost as rapidly under BMI control as under manual control.

Inferring the probability of moving

Decoders that directly translate neural state to cursor velocity have historically had difficulty remaining stationary when there is no intended movement. The ability to do so is of even greater importance for a locomotor prosthetic. To meet this challenge, we adopted the strategy of a state machine with distinct stopped and moving states[37–39]. Transitions between these states were governed by a probability of moving, p_{move}, derived from the neural state in the moving-sensitive dimension.

We identified the moving-sensitive dimension by applying linear discriminant analysis to the 50 training-data trials, and finding the direction that best discriminated whether the monkey was moving versus stopped. Projecting trial-averaged data onto that dimension (**Fig. 6a**) revealed that activity transitioned suddenly from low to high just before movement onset, and back to low around the time movement ended. This pattern was remarkably similar regardless of cycling direction (*red* and *green* traces largely overlap). Activity in this dimension behaved similarly for single trials (**Fig. 6b**).

We used a Hidden Markov Model (HMM)[37,38] to estimate p_{move}, which allows the current estimate to depend on all prior observations. Because those observations must be independent, we did not use filtered rates (which were used for all other aspects of the decode) but instead considered spike counts in non-overlapping bins, projected onto the moving-sensitive dimension. Figure 6c plots the resulting distributions when stopped (*orange*) and moving (*blue*). These overlapped modestly, a result of the narrow (10 ms) bin. The estimate of p_{move} is robust to this overlap because the HMM leverages the full history of spike counts; it can ignore brief weak evidence for moving while still transitioning swiftly given strong evidence. During BMI control, p_{move} (**Fig. 6d**, *blue*) was near typically unity during intended movement (i.e., when the monkey was actually cycling, *black*) and near zero otherwise.

State transitions were determined by p_{move} (**Fig. 3**). Entering a state that produced virtual movement (EARLY or STEADY) required that p_{move} exceed 0.9 and remain consistently above 0.1 for 175 ms. This conservative strategy led to a very low rate of false starts (~2 per day for monkey G and ~1 every ten days for monkey E). The transition to EARLY (**Fig. 6d**, left edge of *gray regions*) occurred on average 117 and 194 ms after physical movement onset (monkeys G and E). Trial-to-trial variability around these mean values was modest: standard deviations were 93 and 138 ms (computed within session and averaged across sessions). As discussed above, estimated stopping time (when p_{move} dropped below 0.1) was also decoded with only modest trial-to-trial variability.

Inferring initial movement direction

Angular momentum of the neural state in the forward and backward planes became substantial a few hundred milliseconds after p_{move} became high. Thus, the EARLY state became active before the direction of movement could be inferred from the elliptical trajectories. To overcome this problem, we identified three dimensions in which the neural state, around the time of movement onset, distinguished between forward and backward movement. The neural state in these dimensions (two of which are shown) differed between forward and backward

movements (*green* and *red* traces) both in the average response (**Fig. 7a**) and on individual trials (**Fig. 7b**). This difference began to grow just prior to physical movement onset (*dark portion* of trajectory shows -200 to +175 ms relative to detected movement onset) and became less prominent later in the movement (*light portion* of trajectory). We found these dimensions by performing PCA on training data (*Methods*). For each of the 50 training trials, we considered the neural state in these dimensions, measured 175 ms after decoded movement onset. We fit Gaussian distributions separately for forward (**Fig. 7c**, *green*) and backward (*red*) trials. These had largely non-overlapping distributions.

During BMI control, upon transition from INIT to EARLY, we computed the likelihood of the neural state under each distribution. A simple winner-take-all computation determined the direction of virtual velocity during the EARLY state. The inference of movement direction during EARLY was correct on 94% and 82% of trials (monkeys G and E). After 200 ms, the STEADY state was entered and virtual velocity was controlled thereafter by activity in the rotational dimensions. **Figure 7d** illustrates moments (*colored regions*) where the EARLY state was active and the above strategy was used to decode virtual motion (physical pedal velocity is shown for reference). These moments were brief, and had a very modest effect on the overall time to reach the target. However, we still employed this strategy because our goal was to build a BMI decode that closely tracked intended movement and felt responsive to the subject.

Speed control

The excellent performance of the decoder was aided by the relative simplicity of behavior: when monkeys moved, they did so at a stereotyped speed. This allowed us to concentrate on building a decode algorithm that decoded intended direction with accurate timing, and remained stationary if movement was not intended. However, that decode provided only limited control of movement speed. An obvious extension is to allow finer-grained speed control. This would presumably be desired by users of a locomotor prosthetic. Furthermore, speed control provides one possible way of steering: e.g., by decoding the relative intensity of intended movement on the two sides of the body. While we do not attempt that here, we still considered it important to determine whether the neural features we identified could support speed control.

That assessment required a task where speed control is necessary for success. We thus trained one monkey to track various speed profiles as he progressed through the virtual environment. Two floating targets were rendered in the foreground as the monkey cycled. The distance between them reflected the difference between actual and instructed speed. Obtaining juice required aligning the two floating targets while progressing towards a final target, on which he stopped to obtain additional reward. The task was divided into trials, each of which required

moving a distance equivalent to twenty cycles under manual control. We used eight trial-types, four each for forward and backward cycling. Two of these employed a constant target speed (equivalent to 1 or 2 Hz cycling) and two involved a ramping speed (from 1 Hz to 2 Hz or vice versa). As above, the decoder was trained based on a small number of manual-control trials performed at the beginning of each session. Blocks of manual-control trials were also included for comparisons between manual and BMI-based performance.

Our decode strategy was largely preserved from that described above. However, we used a modified state machine (**Supp. Fig. 2**) and a slightly different algorithm for transforming rotations of the neural state into decoded virtual velocity. Direction was determined based on which distribution (forward or backward) produced the higher likelihood of observing the measured angular momentums (as in **Fig. 5d**). Once that choice was made, speed was determined by the angular velocity of the neural state in that plane. Thus, faster rotational trajectories led to faster decoded virtual velocity. We chose a scaling factor so that a given neural angular velocity produced the speed that would have been produced by physical cycling at that angular velocity. Neural angular velocity was exponentially filtered with a time constant of 500 ms. The filter memory was erased on entry into a movement state (EARLY or STEADY) from a stopped state (INIT or EXIT) to allow brisk movement onset (see *Methods)*.

The above strategy allowed smooth BMI control of movement speed. In fact, it tended to give BMI control an intrinsic advantage over manual control. In manual control, the angular velocity of the pedal was naturally modulated within each cycle (being higher on the downstroke), resulting in a fluctuating virtual velocity. Such fluctuations mildly impaired the ability to match target speed under manual control. To allow a fair comparison, we thus also applied an exponential filter to virtual velocity under manual control. Filters were chosen separately for BMI ($\tau = 500$ ms) and manual control ($\tau = 1000$ ms) to maximize performance. This was done informally, in the earliest session, by lengthening the filter until success rate roughly plateaued. The filter then remained fixed for all further sessions.

Under BMI control, decoded virtual speed closely tracked instructed speed. This was true across trials with different constant speeds, and within trials where speed modulated with time (**Fig. 8a,b**). To compare BMI with manual control (which were performed on separate days) we considered all trials where the monkey completed the portion of the trial that required matching speed (87% of trials in arm control, and 79% in BMI control). The monkey was able to match instructed speed nearly as accurately under BMI control as under manual control. This was true judged both by time within the rewarded speed window (**Fig. 8c**) and by the error between virtual and instructed velocity (**Fig. 8d**).

Discussion

We have argued that the largest signals in motor cortex are not 'representational' – they do not encode variables but are instead essential for noise-robust dynamics. Those dynamics produce outgoing commands that *are* representational (they covary with the variables they control) but are low-variance. This perspective argues that decoders should not attempt to invert encoding unless a great many neurons can be recorded. Instead, decoders should opportunistically leverage whichever high-variance response features have a robust relationship with the variables one wishes to decode. In retrospect, traditional reach-based prosthetics can be seen as taking exactly this approach; during reaching there exist high-variance neural signals that correlate (linearly) with a projection of two-dimensional reach velocity onto a 'preferred direction'. Similarly, decoding of muscle activity for prosthetic control[6] likely leverages signals that coincidentally but usefully correlate with muscle force during the task of interest.

During cycling, the neural signals that correlate strongly with hand position and velocity are low variance. Yet there exist different high-variance response features that possess robust relationships with intended movement. Those relationships may be nonlinear, but can nevertheless be decoded using straightforward means (e.g., computing angular momentum in state space). Doing so yielded BMI control that was sufficiently natural that monkeys appeared not to notice that the task was no longer under manual control. By most measures (success rate, time to target) performance under BMI control was remarkably close to that under manual control. The main limitation of BMI control was stopping accuracy. Although our algorithm detected stopping with ~ 0.1 second precision, even small discrepancies could lead to the target being over or undershot by a noticeable amount. A beneficial feature of our BMI decode is that it almost never produced movement when it was not intended. With rare exceptions, truly zero velocity was decoded when the monkey was intending to remain stopped on the target. We consider this a particularly important attribute of any locomotor-BMI algorithm, due to the potentially large consequences of unintended movement of the whole body.

The present approach relates to recent studies that modeled neural dynamics to improve online[23] or offline[24,40,41] decodes of movement kinematics. A key insight of those studies is that signals that do not correlate directly with kinematics can be used to infer those that do. For linear decoding, the value of given variable depends upon the neural state in one dimension: the dimension defined by the regression weights. Nevertheless, inferring the neural state in that dimension may benefit from a dynamical model that spans multiple dimensions. Much like the present approach, this allows the decode to leverage features that are robust, even if they do not directly correlate with the kinematic parameters of interest. The present approach extends this

idea to situations where there may be no high-variance dimensions that can be linearly decoded, and/or where the most prominent features are not well-described by linear dynamics.

We had three motivations for developing a BMI algorithm for virtual locomotion. First, our recently developed cycling task naturally lends itself to this application. Second, BMI-controlled locomotion is likely to be desired by a large patient population (potentially much larger than the population that desires BMI-controlled cursors or robot arms). Third, prior work has demonstrated that BMI control of locomotion is viable[9,10]. In particular, Rajangam et al.[9] demonstrated BMI control of a physical wheelchair based on neural activity recorded from monkey motor and somatosensory cortex. Our work supports their conclusion that BMI-controlled locomotion is possible, and demonstrates the feasibility of an alternative decode strategy. Rajangam et al. employed a traditional decode strategy: linear filters transformed neural activity into the key variables: translational and angular velocity. That strategy allowed monkeys to navigate ~2 meters to a target (which had to be approached with an accuracy of +/- ~0.2 meters, or 10% of the distance traveled) in an average of 27-49 seconds (depending on the monkey and degree of practice). In our task, monkeys had to stop with similar relative accuracy: +/- 0.5 cycles, or 7% of the distance traveled for a seven-cycle movement. They traversed those seven cycles in ~4 seconds under BMI control (averages of 4.3 and 3.7 seconds for monkey G and E). While this is roughly tenfold faster, we stress that movement durations are not directly comparable between our task and theirs. Success requirements differed in multiple ways. For example, Rajangam et al. required that monkeys turn *en route* (which adds considerable challenge) but did not require them to stop on the target location. Yet while direct comparison is not possible, a tenfold improvement in time-to-target argues that non-traditional decode strategies can be effective and should be explored further.

An obvious limitation of the current experiments is that we did not explore strategies for steering, which would be essential to a real-world locomotor prosthetic. There exist multiple candidate strategies for enabling steering. Rajangam et al. used a Wiener filter to decode angular velocity of the body. While straightforward, this strategy appears to have had limited success: even during training, the R^2 of their angular velocity decode was 0.16 and 0.12 for the two monkeys. During online performance, the considerable time to reach the target argues that steering was not accurate. One alternative strategy would be to apply our decode strategy bilaterally, and employ a comparison (e.g., between left and right cycling speed) to control angular velocity. Another strategy would be to control translational velocity using the strategies developed here, but use a reach-like decode for steering (rather like pedaling a bicycle while also steering). Which (if any) of these three strategies is preferable remains a question for future experiments.

For convenience, monkeys were trained to control the pedal with their forelimb (allowing them to be seated in a traditional primate chair) and we thus recorded from the forelimb region of motor cortex. Recordings during natural locomotion in monkeys reveal broadly similar signals in both the forelimb[27] and hindlimb[42] regions of motor cortex. These signals are dominated, as in our task, by elliptical neural trajectories during ongoing locomotion. It is thus likely that prosthetic locomotion could be driven by signals derived from either region.

Like many proof-of-concept prosthetic systems developed in primates[1,16,43–45], decoder training depended on observations of neural activity under manual control. Whether this approach translates depends upon the assumption that useful patterns of neural activity will emerge when a paralyzed patient tries to move but can't. A number of existing studies indicate that motor cortex is active, in reasonably normal ways, when paralyzed patients attempt to move. Importantly, decode strategies based on a characterization of population activity during normal reaching in primates have provided successful directional control of a cursor in human patients[1,2,43]. The approach to specifying parameters was of course tailored to the needs of the patients, but the class of decoder did not need to be altered. It seems likely that the same will be true of rhythmic neural activity and decoding of intended movement. More broadly, a key point of the present study is that a fixed decode strategy is unlikely to work well across the different classes of movement that patients are likely to desire. Instead, decode strategies should leverage population-level response features that relate robustly to the variables one wishes to control. Such features are presumably present in paralyzed patients, and will almost certainly be task dependent.

Our results indicate that a nonlinear, yet relatively simple, decode strategy can afford excellent one-dimensional control of locomotion. Although other approaches remain possible[46], our findings support the idea that cortical control of prosthetic locomotion is viable and should be explored further. More broadly, the present results argue that many of the decode strategies that proved effective for reach-based prostheses are unlikely to generalize across tasks. An alternative approach is to identify, for each task, the dominant features and determine how they might be usefully translated into decoded movement. While this approach abandons the elegant idea of inverting a literal encoding of kinematics, it opens up possibilities for improved prosthetic control across a variety of contexts.

Methods

Subjects and primary task

All procedures were approved by the Columbia University Institutional Animal Care and Use Committee. Subjects G and E were two adult male macaque monkeys (*Macaca mulatta*). Monkeys sat in a primate chair facing an LCD monitor (144 Hz refresh rate) that displayed a virtual environment generated by the Unity engine (Unity Technologies, San Francisco, CA). The head was restrained via a titanium surgical implant. While the monkey's left arm was comfortably restrained, the right arm grasped a hand pedal. Cloth tape was used to ensure consistent placement of the hand on the pedal. The pedal connected via a shaft to a motor (Applied Motion Products, Watsonville, CA), which contained a rotary encoder that measured the position of the pedal with a precision of 1/10,000 of the cycle. The motor was also used to apply forces to the pedal, endowing it with virtual mass and viscosity.

Manual-control sessions for the primary cycling task required that the monkey cycle the pedal in the instructed direction to move through the virtual environment, and stop on top of a lighted target to collect juice reward. The color of the landscape indicated whether cycling must be 'forward' (green landscape, the hand moved away from the body at the top of the cycle) or 'backward' (tan landscape, the hand moved toward the body at the top of the cycle). There were 6 total conditions, defined by cycling direction (forward or backward) and target distance (2, 4, or 7 cycles). Distance conditions were randomized within same-direction blocks (3 trials of each distance per block), and directional blocks were randomized over the course of each experiment. Trials began with the monkey stationary on a target. A second target appeared in the future. To obtain reward, the monkey had to cycle to that target, come to a halt 'on top' of it (in the first-person perspective of the task) and remain stationary for a hold period of 1000-1500 ms (randomized). A trial was aborted without reward if the monkey began moving before target onset (or in the 170 ms after, which would indicate attempted anticipation), if the monkey moved past the target without stopping, or if the monkey moved while awaiting reward. The next trial began 100 ms after the variable hold period. Monkeys performed until they received enough liquid reward that they chose to desist. As their motivation waned, they would at times take short breaks. For both manual-control and BMI-control sessions, we discarded any trials in which monkeys made no attempt to initiate the trial, and did not count them as 'failed'. These trials occurred 2 ± 2 times per session (mean and standard deviation, Monkey G, maximum 10) and 3 ± 3 times per session (Monkey E, maximum 11).

In BMI control, trial parameters and failure conditions were the same as in manual control, for purposes of comparison. The only difference between manual and BMI control was that, in the latter, position in the virtual environment was controlled by the output of a decoder rather than the pedal. We did not prevent or discourage the monkey from cycling during BMI-control blocks, and he continued to do so as normal. In BMI control, monkey G performed an average of 654 trials/session over 20 sessions and monkey E performed an average of 137 trials/session over 17 sessions. Manual-control data for monkey G (average of 229 trials/session over 8 sessions) were collected during sessions in which BMI-control data sets were also collected.

Manual-control sessions for monkey E (average of 231 trials/session over 5 sessions) were interleaved with BMI-control sessions on different days. For monkey G, an additional three manual-control sessions (189, 407, and 394 trials) were employed to record EMG, which was used for the variance captured analysis (**Fig. 2b,c**). We recorded from 5-7 muscles per session, yielding a total of 19 recordings. We made one or more recordings from the three heads of the *deltoid*, the lateral and long heads of *triceps brachii*, the *biceps brachii*, *trapezius*, and *latissimus dorsi*. These muscles were selected due to their clear activations during the cycling task.

Surgery and neural/muscle recordings

Neural activity was recorded using chronic 96-channel Utah arrays (Blackrock Microsystems, Salt Lake City, UT), implanted in the left hemisphere using standard surgical techniques. In each monkey, an array was placed in the region of primary motor cortex (M1) corresponding to the upper arm. In monkey G, a second array was placed in dorsal premotor cortex (PMd), just anterior to the first array. Array locations were selected based on MRI scans and anatomical landmarks observed during surgery. Experiments were performed 1-8 months (monkey G) and 3-4 months (monkey E) after surgical implantation. Neural responses both during the task and during palpation confirmed that arrays were in the proximal-arm region of cortex.

Electrode voltages were filtered (band-pass 0.3 Hz – 7.5 kHz) and digitized at 30 kHz using Digital Headstages, Digital Hubs, and Cerebus Neural Signal Processors from Blackrock Microsystems. Digitized voltages were high-pass filtered (250 Hz) and spike events were detected based on threshold crossings. Thresholds were set to between -4.5 and -3 times the RMS voltage on each channel, depending on the array quality on a given day. On most channels, threshold crossings included clear action-potential waveforms from one or more neurons, but no attempt was made to sort action potentials.

Intra-muscular EMG recordings were made using pairs of hook-wire electrodes inserted with 30 mm x 27 gauge needles (Natus Neurology, Middleton, WI). Raw voltages were amplified and filtered (band-pass 10 Hz – 10 kHz) with ISO-DAM 8A modules (World Precision Instruments, Sarasota, FL), and digitized at 30 kHz with the Cerebus Neural Signal Processors. EMG was then digitally band-pass filtered (50 Hz – 5 kHz) prior to saving for offline analysis. Offline, EMG recordings were rectified, low-pass filtered by convolving with a Gaussian (standard deviation: 25 ms), downsampled to 1 kHz, and then fully normalized such that the maximum value achieved on each EMG channel was 1.

A real-time target computer (Speedgoat, Bern, CH) running Simulink Real-Time environment (MathWorks, Natick, MA) processed behavioral and neural data and controlled the decoder output in online experiments. It also streamed variables of interest to another computer that saved these variables for offline analysis. Stateflow charts were implemented in the Simulink model to control task state flow as well as the decoder state machine. Real-time control had millisecond precision.

Spike trains were causally converted to firing rates by convolving each spike with a beta kernel. The beta kernel was defined by temporally scaling a beta distribution (shape parameters: $\alpha = 3$

and $\beta = 5$) to be defined over the interval [0, 275] ms and normalizing the kernel such that the firing rates would be in units of spikes/second. The same filtering was applied for online decoding and offline analyses. Firing rates were also mean centered (subtracting the mean rate across all times and conditions) and normalized. During online decoding, the mean and normalization factor were values that had been computed from the training data. We used soft normalization[18]: the normalization factor was the firing rate range plus a constant (5 spikes/s).

Computing trial-averaged firing rates

Analyses of BMI performance are based on real-time decoding during online performance, with no need to consider trial-averaged firing rates. However, we still wished to compute trial-averaged traces of neural activity and kinematics for two purposes. First, some aspects of decoder training benefited from analyzing trial-averaged firing rates. Second, we employ analyses that document basic features of single-neuron responses and of the population response (e.g., **Fig. 1d**, **Fig. 2**, **Fig. 5a**, **Fig. 6a**). These analyses benefit from the denoising that comes from computing a time-varying firing rate across many trials. Due to the nature of the task, trials could be quite long (up to 20 cycles in the speed-tracking task), rendering the traditional approach of aligning all trials to movement onset insufficient for preserving alignment across all subsequent cycles. It was thus necessary to modestly adjust the time-base of each individual trial (e.g., stretching time slightly for a trial where cycling was faster than typical). We employed two alignment methods. Method A is a simplified procedure that was used prior to parameter fitting when training the decoder before online BMI control. This method aligns only times during the movement. Method B is a more sophisticated alignment procedure that was utilized for all offline analyses. This method aligns the entire trial, including pre- and post-movement data. For visualization, conditions with the same target distance (e.g., 7 cycles), but different directions, were also aligned to the same time base. Critically, any data processing that relied on temporal structure was completed in the original, unstretched time base prior to alignment.

Method A*:* The world position for each trial resembles a ramp between movement onset and offset (**Fig. 1a**). First, we identify the portion of each trial starting ¼ cycle into the movement and ending ¼ cycle before the end of the movement. We fit a line to the world position in this period and then extend that line until it intercepts the starting and ending positions. The data between these two intercepts is considered the movement data for each trial and is extracted. This movement data is then uniformly stretched in time to match the average trial length for each trial's associated condition. This approach compresses slower than average movements and stretches faster than average movements within a condition, such that they can be averaged while still preserving many of the cycle-specific features of the data.

Method B: This method consists of a mild, non-uniform stretching of time in order to match each trial to a condition-specific template. For complete details, see Russo et al. 2018.[18]

Variance captured analysis

Analysis of neural variance captured (**Fig. 2**) was based on successful manual-control trials from the three sessions with simultaneous neural and muscle recordings. We considered data from the full duration of each trial, including times before movement onset and after movement offset. We analyzed the variance captured by neural dimensions of three types. First, neural dimensions where activity correlated strongly with kinematic features. Second, neural dimensions where activity correlated strongly with muscle activity. Third, neural dimensions that captured robust 'features' leveraged by our decoder.

Dimensions of the third type (were found as detailed below in a dedicated section below. Dimensions of the first two types were found using the model $z(r,t) = c + \boldsymbol{w}^{\top}\boldsymbol{y}(r,t)$, where $z(r,t)$ is the kinematic or muscle variable at time t during trial r, and $\boldsymbol{y}(r,t)$ is the corresponding N-dimensional vector of neural firing rates. The constant c and the column vector $\boldsymbol{w}$ were found via regression. The vector $\boldsymbol{w}$ defines a direction in neural space where activity correlates strongly with the variable z. We found multiple such vectors; e.g. $\boldsymbol{w}_{x-vel}$ is a dimension where neural activity correlates with horizontal velocity and $\boldsymbol{w}_{biceps}$ is a dimension where neural activity correlates with biceps activity. All such vectors were scaled to have unity norm before computing the neural variance captured by that dimension. Regression was based on single-trial responses because this was intrinsically regularizing. We wished to encourage regression to find high-variance dimensions if possible, and the use of single-trial data encouraged it to do so. Because filtering of neural activity introduces a net lag, this analysis naturally assumes a ~100 ms lag between neural activity and the variables of interest. Results were extremely similar if we considered longer or shorter lags.

We wished to compute, for each dimension, the percentage of neural variance explained – i.e., whether that dimension captured large or small signals. We were not interested in whether dimensions captured stochastic spiking variability, but in whether they captured large features that were reliable across trials. Thus, variance captured was always computed based on trial-averaged neural responses. We considered the matrix $\overline{Y} \in \mathbb{R}^{N \times T}$ where T is the total number of time points across all conditions. Each row of $\overline{Y}$ contains the trial-averaged firing rate of one neuron. We computed an $N \times N$ covariance matrix $\Sigma = \mathrm{cov}(\overline{Y})$ by treating rows of $\overline{Y}$ as random variables and columns as observations. The proportion of total neural variance captured by a given dimension, $\boldsymbol{w}$, is therefore:

$$\frac{\boldsymbol{w}^{\top}\Sigma\boldsymbol{w}}{\mathrm{tr}(\Sigma)}$$

Some analyses considered the variance captured by a subspace spanned by a set of dimensions. To do so we took the sum of the variance captured by orthonormal dimensions spanning that space.

Identifying neural dimensions

Although the response features leveraged by the decode algorithm are clearly visible in the top principal components of the data (when PCA is performed on the full trial-averaged time-series

of firing rates across conditions), we sought to choose neural dimensions that would cleanly isolate particular features. To this end, each feature was isolated using dedicated preprocessing and dimensionality reduction approaches.

We sought a moving-sensitive dimension, the projection onto which would allow an HMM to estimate the probability of moving, p_{move}, at each moment. To do so, we first computed binned spike-counts (10 ms for monkey G, 20 ms for monkey E) and applied a square-root transform to these counts as this has been shown to improve the Gaussian fit for Poisson data with small counts[47]. We then aggregated all of these square-rooted binned counts from the training set (25 forward trials, 25 backward trials) and separated them into two classes based on pedaling speed: 'moving' (speed > 1 Hz) and 'stopped' (speed < .05 Hz). Samples that didn't fall into either of these two classes were discarded. We applied linear discriminant analysis to these two labeled sets, which yielded a discriminating hyperplane that best separated the two classes. We defined the moving-sensitive dimension, $\boldsymbol{w}_{move}$, as the vector normal to this hyperplane.

In order to decode direction, we sought to isolate four neural dimensions that captured rotational trajectories during steady-state cycling. Spike time-series were filtered to yield firing rates (as described above), and then futher high-pass filtered (2nd order Butterworth, cutoff frequency: 1 Hz). This removed drift or other low-frequency signals. Single-trial movement-period responses were then aligned (Method A) and averaged within conditions to generate $N \times T_c$ matrices $\overline{Y}_f$ and $\overline{Y}_b$. We sought a 4-dimensional projection of these trial-averaged responses that would maximally capture rotational trajectories while segregating forward and backward data into different planes. Whereas the standard PCA cost function finds dimensions that maximize variance captured, we opted instead for a cost function that would maximize the difference in variance captured between the two conditions:

$$J(W) = \mathrm{tr}(W^\top \Sigma_f W) - \mathrm{tr}(W^\top \Sigma_b W)$$

where $\Sigma_f = \mathrm{cov}(\overline{Y}_f)$, $\Sigma_b = \mathrm{cov}(\overline{Y}_b)$, W is constrained to be orthonormal. Note that this cost function will be maximized when the projection of the data captures a great deal of variance for forward trials and very little variance for backward trials. Conversely, this cost function will be minimized when the projection favors large variances for backward trials and small variances for forward trials. We thus chose to define our forward rotational plane by the 2D matrix $W_f = \left[\boldsymbol{w}_f^{(1)} \quad \boldsymbol{w}_f^{(2)}\right]$ that maximizes $J(W)$ and our backward rotational plane by the 2D matrix $W_b = \left[\boldsymbol{w}_b^{(1)} \quad \boldsymbol{w}_b^{(2)}\right]$ that minimizes $J(W)$. An iterative optimization procedure was used to find W_f and W_b; full details of this in[48].

To decode direction during the EARLY state, we found a set of initial-direction dimensions. We used activity in the moving-sensitive dimension to determine the time, t_{init}, at which the state machine would have entered the INIT state during online control. We then considered trial-averaged neural activity, for each condition. from t_{init} through t_{init} + 200 ms. We applied PCA and retained the top three dimensions: $\boldsymbol{w}_{dir}^{(1)}$, $\boldsymbol{w}_{dir}^{(2)}$, and $\boldsymbol{w}_{dir}^{(3)}$. Such dimensions capture how

activity evolves both across that timespan, and how it differs across forwards and backwards cycling conditions.

Computing probability of moving (p_{move})

To compute p_{move} based on neural activity in the moving-sensitive dimension, an HMM was used to track two states: 'moving' or 'stopped'[38]. Square-rooted spike counts in the training data were already separated into 'moving' and 'stopped' sets for the purposes of identifying $\boldsymbol{w}_{move}$. We projected those counts onto $\boldsymbol{w}_{move}$ and a fit Gaussian distribution for each state. The probability, p_{move}, of being in the 'moving' state, given the entire sequence of current and previously observed square-rooted spike counts, was computed efficiently with a recursive algorithm that uses the state transition matrix

$$\Phi = \begin{bmatrix} p_{move|move} & p_{move|stop} \\ p_{stop|move} & p_{stop|stop} \end{bmatrix}$$

and knowledge of the Gaussian distributions. Φ encodes prior assumptions about the probability of transitioning from one state to the next at any given bin. We used a benchmark set of manual-control training data from each monkey to determine reasonable values for Φ, which were then used in all experiments. For monkey G, we set $p_{move|stop} = .0001$ and $p_{stop|move} = .002$; for monkey E, we set $p_{move|stop} = .0002$ and $p_{stop|move} = .004$. The value p_{move} was used throughout the decoder state machine to control transitions between various states, effectively dictating the movement onset and offset behavior of the decoder (**Fig. 6d**).

Computing steady-state direction and speed

Projecting single-trial, high-pass filtered firing rates onto the rotational planes spanned by W_f and W_b yielded trajectories that differed considerably between forward and backward conditions. To further denoise these state trajectories we applied a Kalman filter of the form

$$x_t = Ax_{t-1} + q_t$$
$$y_t = Cx_t + r_t$$

where $q_t \in \mathcal{N}(0, Q)$, and $r_t \in \mathcal{N}(0, R)$. In these equations, x_t represents the true underlying neural state in the rotational dimensions and y_t are the high-pass filtered firing rates, which we treat as noisy measurements of that underlying state. We chose to let our measurements be smooth firing rates, rather than use non-overlapping bins of spikes, for purely opportunistic reasons: it consistently yielded better performance by our decoder. The parameters of the Kalman filter were fit to the training data as follows:

$$A = \overline{X}_2 \overline{X}_1^\top \left(\overline{X}_1 X_1^\top\right)^{-1}$$

$$C = \begin{bmatrix} W_f^\top \\ W_b^\top \end{bmatrix}^\dagger$$

$$Q = \mathrm{cov}(\overline{X}_2 - A\overline{X}_1)$$

$$R = \mathrm{cov}(Y - C\overline{X})$$

where

$$\overline{X}_1 = \begin{bmatrix} W_f^\top \\ W_b^\top \end{bmatrix} [\overline{Y}_f(:,1{:}T_f - 1), \overline{Y}_b(:,1{:}T_b - 1)]$$

$$\overline{X}_2 = \begin{bmatrix} W_f^\top \\ W_b^\top \end{bmatrix} [\overline{Y}_f(:,2{:}T_f), \overline{Y}_b(:,2{:}T_b)]$$

$$\overline{X} = \begin{bmatrix} W_f^\top \\ W_b^\top \end{bmatrix} [\overline{Y}_1, \overline{Y}_2, \dots, \overline{Y}_{50}]$$

$$Y = [Y_1, Y_2, \dots, Y_{50}]$$

with Y_i denoting the neural activity (high-pass filtered firing rates) for the i-th trial in the training set, $\overline{Y}_i$ denoting the trial-averaged activity for the condition that the i-th trial is an instantiation of, † denoting the Moore-Penrose pseudoinverse, and the colon symbol designating how to index matrices (e.g., $M(:,a{:}b)$ refers to the submatrix of M including all rows of M, but only the columns a through b). Lastly, the initial state parameter x_0 was computed by taking the average value of the trial-averaged projections over all times and conditions. Online inference of the underlying neural state, which yields an estimate $\hat{x}_t$ at each millisecond t, was computed recursively using the steady-state form of the Kalman filter[49].

After denoising the neural state in the rotational dimensions via the Kalman filter, angular momentum was computed in each plane as the cross product between the estimated neural state and its derivative, which (up to a constant scaling) can be written

$$L(t) = \begin{bmatrix} L_f(t) \\ L_b(t) \end{bmatrix} = \begin{bmatrix} \hat{x}_{t-1}^{(1)}\hat{x}_t^{(2)} - \hat{x}_t^{(1)}\hat{x}_{t-1}^{(2)} \\ \hat{x}_{t-1}^{(3)}\hat{x}_t^{(4)} - \hat{x}_t^{(3)}\hat{x}_{t-1}^{(4)} \end{bmatrix}$$

where the superscript indexes the elements of $\hat{x}_t$. We fit 2D Gaussian distributions to these angular momentums for each of three behaviors in the training data: 'stopped' (speed < .05 Hz), 'pedaling forward' (velocity > 1 Hz), and 'pedaling backward' (velocity < -1 Hz) (**Fig. 5d**). Online, the likelihood of the observed angular momentums with respect to each of these three distributions dictated the steady-state estimates of direction and speed. We'll denote these three likelihoods f_{stop}, $f_{forward}$, and $f_{backward}$.

In general, one can compute which of these three distributions is most likely by choosing the maximizing likelihood and assess confidence in that choice by comparing the relative values of the three likelihoods. However, we wanted the decoder to err on the side of withholding movement. We therefore set a conservative threshold on f_{stop} corresponding to the point at which L would have a Mahalanobis distance of 3 to the stopped distribution of angular

momentums. If f_{stop} ever exceeded this threshold, we set $speed_{steady}$ to zero. If this condition was not met, we decoded direction and speed as follows:

$$direction_{steady}(t) = \text{sgn}\left(f_{forward}(t) - f_{backward}(t)\right)$$

$$speed_{steady}(t) = \left|2 \cdot \frac{f_{forward}(t)}{f_{forward}(t) + f_{backward}(t)} - 1\right| \beta$$

where $\left|2 \cdot \frac{f_{forward}(t)}{f_{forward}(t)+f_{backward}(t)} - 1\right|$ varies between 0 and 1 depending on the relative sizes of the likelihoods (yielding a slower velocity if the direction decode is uncertain) and β is a direction-specific constant learned from the training data whose purpose is simply to scale up the result to match steady-state cycling speed. In practice, $speed_{steady}$ was frequently very close to the monkeys' steady-state cycling speeds (**Fig. 5e**).

Computing initial direction and speed

Initial direction and speed were always computed at the moment the EARLY state was entered, t_{early}. These values then persisted throughout the remainder of the EARLY state. Given that the decoder state machine doesn't make use of the initial-direction dimensions prior to entering the EARLY state, t_{early} can be computed for the training trials. Single-trial firing rates from the training set were then projected onto the initial-direction dimensions at t_{early} and 3D Gaussian distributions were fit to the resulting sets of forward and backward neural states. Online, firing rates were projected onto the initial-direction dimensions at t_{early} and likelihoods $g_{forward}$ and $g_{backward}$ were computed with respect to each the learned distributions. If the observed neural state in the initial-direction subspace was not an outlier (>10 Mahalanobis distance units) with respect to both distributions, then the initial direction and speed were computed as follows:

$$direction_{initial}(t_{early}) = \text{sgn}\left(g_{forward}(t_{early}) - g_{backward}(t_{early})\right)$$

$$speed_{initial}(t_{early}) = \left|2 \cdot \frac{g_{forward}(t_{early})}{g_{forward}(t_{early}) + g_{backward}(t_{early})} - 1\right| \beta$$

If the observed neural state was an outlier, initial direction and speed were computed in the same manner as is done in the STEADY state.

Smoothing of decoded velocity

In the primary experiment, the decoder state machine produced an estimate of velocity, v_{dec}, at every millisecond. During the STOP and INIT states, this estimate was zero and the monkey's position in the virtual environment was held constant. During the EARLY and STEADY states, this estimate was smoothed with a trailing average:

$$v'_{dec}(t) = \frac{1}{T_{smooth}+1} \sum_{i=0}^{T_{smooth}} v_{dec}(t-i)$$

where $T_{smooth} = \min(500, t - t_{early})$, i.e., the trailing average extended in history up to 500 ms or to the moment the EARLY state was entered, whichever was shorter. v'_{dec} was integrated every millisecond to yield decoded position in the virtual environment. In the speed-tracking experiment (described below) there was no need to smooth of v_{dec} prior to integration because the speed estimate had already been smoothed.

Speed-tracking task

In addition to the primary task (where the monkey traveled 2-7 cycles between stationary targets) we employed a speed-tracking task, in which the monkey was required to match his virtual speed to an instructed speed. Speed was instructed implicitly, via the relative position of two moving targets. The primary target was located a fixed distance in front of the monkey's location in virtual space: the secondary target fell 'behind' the first target when cycling was too slow, and pulled 'ahead' if cycling was too fast. This separation saturated for large errors, but for small errors was proportional to the difference between the actual and instructed speed. This provided sufficient feedback to allow the monkey to track the instructed speed even when it was changing. Because there was no explicit cue regarding the absolute instructed speed, monkeys began cycling on each trial unaware of the true instructed speed profile and 'honed in' on that speed over the first ~2 cycles.

We quantify instructed speed not in terms of the speed of translation through the virtual environment (which has arbitrary units) but in terms of the physical cycling velocity necessary to achieve the desired virtual speed. E.g., an instructed speed of 2 Hz necessitated cycling at an angular velocity of 2 Hz to ensure maximal reward. Under BMI control, the output of the decoder had corresponding units. For example, a 2 Hz angular velocity of the neural trajectory produced movement at the same speed as 2 Hz physical cycling (see *'Neural features for speed-tracking'* for details of decoder). Reward was given throughout the trial so long as the monkey's speed was within 0.2 Hz of the instructed speed. We employed both constant and ramping instructed-speed profiles.

Constant profiles were at either 1 Hz or 2 Hz. Trials lasted 20 cycles. After 18 cycles, the primary and secondary targets (described above) disappeared and were replaced by a final stationary target two cycles in front of the current position. Speed was not instructed during these last two cycles; the monkey simply had to continue cycling and stop on the final target to receive a large reward. Analyses of performance (e.g., **Fig. 8c,d**) were based on the ~16 cycle period starting when the monkey first honed in on the correct speed (within 0.2 Hz of the instructed speed) and ending when the speed-instructing cues disappeared 2 cycles before the trial's end.

Ramping profiles began with three seconds of constant instructed speed to allow the monkey to hone in on the correct initial speed. Instructed speed then ramped, over 8 seconds, to a new value, and remained constant thereafter. As for constant profiles, speed-instructing cues

disappeared after 18 cycles and the monkey cycled two further cycles before stopping on a final target. Again, analyses of performance were based on the period from when the monkey first honed in on the correct speed, to when the speed-instructing cues disappeared. There were two ramping profiles: one ramping up from 1 to 2 Hz, and one ramping down from 2 to 1 Hz. There were thus four total speed profiles (two constant and two ramping). These were performed for both cycling directions (presented in blocks and instructed by color as in the primary task) yielding eight total conditions. This task was only performed by monkey G, who completed an average of 166 trials/session over 2 sessions in manual control and an average of 116 trials/session over 3 sessions in BMI control.

As will be described below, the speed decoded during BMI control was low-pass filtered to remove fluctuations due to noise. This had the potential to actually make the task easier under BMI control, given that changes in instructed speed were slow within a trial (excepting the onset and offset of movement). We did not wish to provide BMI control with an 'unfair' advantage in comparisons with manual control. We therefore also low-pass filtered virtual speed while under manual control. Filtering (exponential, $\tau = 1$ second) was applied only when speed was above 0.2 Hz, so that movement onset and offset could remain brisk. This aided the monkey's efforts to track slowly changing speeds under manual control.

During training and while under manual control, trials were failed if there was ever a large discrepancy between actual and instructed speed. This ensured that monkeys tried their best to consistently match speed at all times. We relaxed this failure mode under BMI control because we did not wish to mask large failures in decoded speed. Over the course of single sessions, this did not discourage monkeys from trying their best, but simply allowed us to observe and quantify decode failures that would otherwise have resulted in aborted trials. This potentially puts BMI performance – quantified as in **Figure 8c,d** – at a disadvantage relative to manual control, where large errors could not persist. In practice this was not an issue as large errors were rare.

Neural features for speed-tracking

Although the speed-tracking experiment leveraged the same dominant neural responses that were used in the primary experiment, the specific features calculated for the decoder state machines differed. Details on how the relevant features were calculated in the speed-tracking experiment are presented in this section.

The probability of moving, p_{move}, was calculated using a different set of parameters for speed-tracking, largely due to changes in recording quality in the intervening time between data collection from the primary experiment and data collection for the speed-tracking experiment. The bin size was increased to 100 ms and the following state transition values were used: $p_{move|stop} = .0005$ and $p_{stop|move} = .0005$. In addition, we observed that the square-root transform seemed to be having a negligible impact on the quality of the decoder at this bin size, so we removed it for this task.

Several features used in the speed-tracking state machine rely on neural activity in the rotational dimensions. In the primary experiment, this activity was high-pass filtered (cutoff frequency: 1 Hz) prior to projection into these dimensions, which helped isolate the rotational neural trajectories during ~2 Hz cycling. For speed-tracking, we wanted to accommodate a broader range of cycling speeds (which corresponded to a broader range of periodicities in the rotational neural trajectories). Thus, we dropped the cutoff frequency from 1 Hz to 0.75 Hz for this experiment.

In computing $direction_{steady}$, the same computations were performed as for the primary-experiment, with one exception: a new direction was not necessarily decoded every millisecond. In order to decode a new direction, the follow conditions needed to be met: 1) the observed angular momentums had a Mahalanobis distance of less than 4 to the distribution corresponding to the decoded direction, 2) the observed angular momentums had a Mahalanobis distance of greater than 6 to the distribution corresponding to the opposite direction. These criteria ensured that a new steady-state direction was only decoded when the angular momentums were highly consistent with a particular direction. When these criteria were not met, the decoder continued to decode the same direction from the previous time step.

Speed was computed identically in the EARLY and STEADY states by decoding directly from the rotational plane corresponding to the decoded direction. A coarse estimate of speed was calculated as the derivative of the phase of rotation follows:

$$\theta'(t) = \begin{cases} \frac{d\theta_f}{dt}, & direction(t) = +1 \\ \frac{d\theta_b}{dt}, & direction(t) = -1 \end{cases}$$

where $\theta_f(t)$ and $\theta_b(t)$ are the phases of the two planes in the neural state estimate $\hat{x}_t$, $direction$ corresponds to $direction_{early}$ while in the EARLY state and $direction_{steady}$ while in the STEADY state, and the derivative θ' is computed in units of Hz. The coarse speed estimate, θ', was then smoothed with an exponential moving average (τ = 500 ms) to generate $speed$, the variable that gets used in the decoder state machine. Additional saturation limits were set such that $speed$ never dropped below 0.5 Hz or exceeded 3.5 Hz, so as to remain in the range typically seen during pedaling. On entry into EARLY or STEADY from either INIT or EXIT, when $speed$ gets initialized, the output of this exponential moving average was reset to an initial value of 1.5 Hz, which was the average starting speed across conditions.

Lastly, there were two new conditions for decoder state transitions in the speed-tracking experiment (**Supp. Fig. 1**). First, transitions from INIT to EARLY required that a condition termed "confident initial direction decode" was obtained. This condition was met when the Mahalanobis distance from the neural state in the initial-direction subspace to either the forward or backward distributions dropped below 4. Second, transitions into the EXIT state required (in addition to a drop in p_{move}) that the observed angular momentums, L, belong to a

set termed 'Stationary'. This set was defined as all L with a Mahalanobis distance of less than 4 to the 'stopped' distribution of angular momentums, which was learned from the training set.

Acknowledgements

We thank Y. Pavlova for expert animal care, A. Russo for sharing code and data for preliminary analysis, and E. Oby for surgical expertise and assistance. This work was supported by NINDS 1DP2NS083037, NIH CRCNS R01NS100066, NINDS 1U19NS104649, the Simons Foundation (SCGB#325233 and SCGB#542957), the Grossman Center for the Statistics of Mind, the McKnight Foundation, P30 EY019007, a Klingenstein-Simons Fellowship, and the Searle Scholars Program.

Author Contributions

M.M.C. conceived the study. K.E.S., S.M.P., and M.M.C. designed experiments. S.M.P., K.E.S., M.M.C., and Q.W. created the decoding algorithms. K.E.S. and S.M.P. collected and analyzed datasets. K.E.S., S.M.P., and M.M.C. wrote the paper. All authors contributed to editing.

References

1. Gilja, V. *et al.* A high-performance neural prosthesis enabled by control algorithm design. *Nat. Neurosci.* **15**, 1752–1757 (2012).
2. Gilja, V. *et al.* Clinical translation of a high-performance neural prosthesis. *Nat. Med.* **21**, 1142–1145 (2015).
3. Shanechi, M. M. *et al.* Rapid control and feedback rates enhance neuroprosthetic control. *Nat. Commun.* **8**, 13825 (2017).
4. Shenoy, K. V. & Carmena, J. M. Combining Decoder Design and Neural Adaptation in Brain-Machine Interfaces. *Neuron* **84**, 665–680 (2014).
5. Ajiboye, A. B. *et al.* Restoration of reaching and grasping movements through brain-controlled muscle stimulation in a person with tetraplegia: a proof-of-concept demonstration. *The Lancet* **0**, (2017).
6. Ethier, C., Oby, E. R., Bauman, M. J. & Miller, L. E. Restoration of grasp following paralysis through brain-controlled stimulation of muscles. *Nature* **485**, 368–371 (2012).
7. Collinger, J. L. *et al.* 7 degree-of-freedom neuroprosthetic control by an individual with tetraplegia. *Lancet* **381**, 557–564 (2013).
8. Wodlinger, B. *et al.* Ten-dimensional anthropomorphic arm control in a human brain–machine interface: difficulties, solutions, and limitations. *J. Neural Eng.* **12**, 016011 (2015).
9. Rajangam, S. *et al.* Wireless Cortical Brain-Machine Interface for Whole-Body Navigation in Primates. *Sci. Rep.* **6**, (2016).
10. Libedinsky, C. *et al.* Independent Mobility Achieved through a Wireless Brain-Machine Interface. *PLOS ONE* **11**, e0165773 (2016).
11. Velliste, M., Perel, S., Spalding, M. C., Whitford, A. S. & Schwartz, A. B. Cortical control of a prosthetic arm for self-feeding. *Nature* **453**, 1098–1101 (2008).
12. Chapin, J. K., Moxon, K. A., Markowitz, R. S. & Nicolelis, M. A. L. Real-time control of a robot arm using simultaneously recorded neurons in the motor cortex. *Nat. Neurosci.* **2**, 664–670 (1999).
13. Wessberg, J. *et al.* Real-time prediction of hand trajectory by ensembles of cortical neurons in primates. *Nature* **408**, 361–365 (2000).

14. Carmena, J. M. *et al.* Learning to Control a Brain–Machine Interface for Reaching and Grasping by Primates. *PLoS Biol* **1**, e42 (2003).
15. Taylor, D. M., Tillery, S. I. H. & Schwartz, A. B. Direct Cortical Control of 3D Neuroprosthetic Devices. *Science* **296**, 1829–1832 (2002).
16. Serruya, M. D., Hatsopoulos, N. G., Paninski, L., Fellows, M. R. & Donoghue, J. P. Brain-machine interface: Instant neural control of a movement signal. *Nature* **416**, 141–142 (2002).
17. Scott, S. H., Gribble, P. L., Graham, K. M. & Cabel, D. W. Dissociation between hand motion and population vectors from neural activity in motor cortex. *Nature* **413**, 161–165 (2001).
18. Russo, A. A. *et al.* Motor Cortex Embeds Muscle-like Commands in an Untangled Population Response. *Neuron* **97**, 953-966.e8 (2018).
19. Churchland, M. M. *et al.* Neural population dynamics during reaching. *Nature* **487**, 51–56 (2012).
20. Michaels, J. A., Dann, B. & Scherberger, H. Neural Population Dynamics during Reaching Are Better Explained by a Dynamical System than Representational Tuning. *PLOS Comput. Biol.* **12**, e1005175 (2016).
21. Shenoy, K. V., Sahani, M. & Churchland, M. M. Cortical control of arm movements: A dynamical systems perspective. *Annu. Rev. Neurosci.* **36**, 337–359 (2013).
22. Sussillo, D., Churchland, M. M., Kaufman, M. T. & Shenoy, K. V. A neural network that finds a naturalistic solution for the production of muscle activity. *Nat. Neurosci.* **18**, 1025–1033 (2015).
23. Kao, J. C. *et al.* Single-trial dynamics of motor cortex and their applications to brain-machine interfaces. *Nat. Commun.* **6**, 1–12 (2015).
24. Aghagolzadeh, M. & Truccolo, W. Inference and Decoding of Motor Cortex Low-Dimensional Dynamics via Latent State-Space Models. *IEEE Trans. Neural Syst. Rehabil. Eng.* **24**, 272–282 (2016).
25. Makin, J. G., O'Doherty, J. E., Cardoso, M. M. B. & Sabes, P. N. Superior arm-movement decoding from cortex with a new, unsupervised-learning algorithm. *J. Neural Eng.* **15**, 026010 (2018).
26. Wu, W., Gao, Y., Bienenstock, E., Donoghue, J. P. & Black, M. J. Bayesian Population Decoding of Motor Cortical Activity Using a Kalman Filter. *Neural Comput.* **18**, 80–118 (2006).
27. Foster, J. D. *et al.* A freely-moving monkey treadmill model. *J. Neural Eng.* **11**, 046020 (2014).
28. Fetz, E. E., Finocchio, D. V., Baker, M. A. & Soso, M. J. Sensory and motor responses of precentral cortex cells during comparable passive and active joint movements. *J. Neurophysiol.* **43**, 1070–1089 (1980).
29. Lemon, R. N., Hanby, J. A. & Porter, R. Relationship between the activity of precentral neurones during active and passive movements in conscious monkeys. *Proc. R. Soc. Lond. B Biol. Sci.* **194**, 341–373 (1976).
30. Suminski, A. J., Tkach, D. C. & Hatsopoulos, N. G. Exploiting multiple sensory modalities in brain-machine interfaces. *Neural Netw.* **22**, 1224–1234 (2009).
31. Schroeder, K. E. *et al.* Robust tactile sensory responses in finger area of primate motor cortex relevant to prosthetic control. *J. Neural Eng.* **14**, 046016 (2017).
32. Pruszynski, J. A. *et al.* Primary motor cortex underlies multi-joint integration for fast feedback control. *Nature* **478**, 387–390 (2011).
33. Christie, B. P. *et al.* Comparison of spike sorting and thresholding of voltage waveforms for intracortical brain–machine interface performance. *J. Neural Eng.* **12**, 016009 (2014).
34. Moran, D. W. & Schwartz, A. B. Motor Cortical Representation of Speed and Direction During Reaching. *J. Neurophysiol.* **82**, 2676–2692 (1999).
35. Georgopoulos, A. P., Schwartz, A. B. & Kettner, R. E. Neuronal population coding of movement direction. *Science* **233**, 1416–1419 (1986).
36. Kaufman, M. T. *et al.* The Largest Response Component in the Motor Cortex Reflects Movement Timing but Not Movement Type. *eNeuro* **3**, (2016).

37. Kemere, C. *et al.* Detecting Neural-State Transitions Using Hidden Markov Models for Motor Cortical Prostheses. *J. Neurophysiol.* **100**, 2441–2452 (2008).
38. Kao, J. C., Nuyujukian, P., Ryu, S. I. & Shenoy, K. V. A High-Performance Neural Prosthesis Incorporating Discrete State Selection With Hidden Markov Models. *IEEE Trans. Biomed. Eng.* **64**, 935–945 (2017).
39. Ethier, C., Sachs, N. A. & Miller, L. E. Continuous state-dependent decoders for brain machine interfaces. in *2011 5th International IEEE/EMBS Conference on Neural Engineering* 473–477 (2011). doi:10.1109/NER.2011.5910589.
40. Kao, J. C., Ryu, S. I. & Shenoy, K. V. Leveraging neural dynamics to extend functional lifetime of brain-machine interfaces. *Sci. Rep.* **7**, (2017).
41. Gallego, J. A., Perich, M. G., Chowdhury, R. H., Solla, S. A. & Miller, L. E. Long-term stability of cortical population dynamics underlying consistent behavior. *Nat. Neurosci.* **23**, 260–270 (2020).
42. Xing, D., Aghagolzadeh, M., Truccolo, W. & Borton, D. Low-Dimensional Motor Cortex Dynamics Preserve Kinematics Information During Unconstrained Locomotion in Nonhuman Primates. *Front. Neurosci.* **13**, (2019).
43. Santhanam, G., Ryu, S. I., Yu, B. M., Afshar, A. & Shenoy, K. V. A high-performance brain-computer interface. *Nature* **442**, 195–198 (2006).
44. Sadtler, P. T., Ryu, S. I., Tyler-Kabara, E. C., Yu, B. M. & Batista, A. P. Brain–computer interface control along instructed paths. *J. Neural Eng.* **12**, 016015 (2015).
45. Nuyujukian, P., Kao, J. C., Ryu, S. I. & Shenoy, K. V. A Nonhuman Primate Brain–Computer Typing Interface. *Proc. IEEE* **105**, 66–72 (2017).
46. Fernández-Rodríguez, Á., Velasco-Álvarez, F. & Ron-Angevin, R. Review of real brain-controlled wheelchairs. *J. Neural Eng.* **13**, 061001 (2016).
47. Thacker, N. A. & Bromiley, P. A. *The Effects of a Square Root Transform on a Poisson Distributed Quantity.* Technical Report http://www.tina-vision.net/tina-knoppix/tina-memo/2001-010.pdf (2001).
48. Cunningham, J. P. & Ghahramani, Z. Linear Dimensionality Reduction: Survey, Insights, and Generalizations. *J Mach Learn Res* **16**, 2859–2900 (2015).
49. Malik, W. Q., Truccolo, W., Brown, E. N. & Hochberg, L. R. Efficient Decoding With Steady-State Kalman Filter in Neural Interface Systems. *IEEE Trans. Neural Syst. Rehabil. Eng.* **19**, 25–34 (2011).

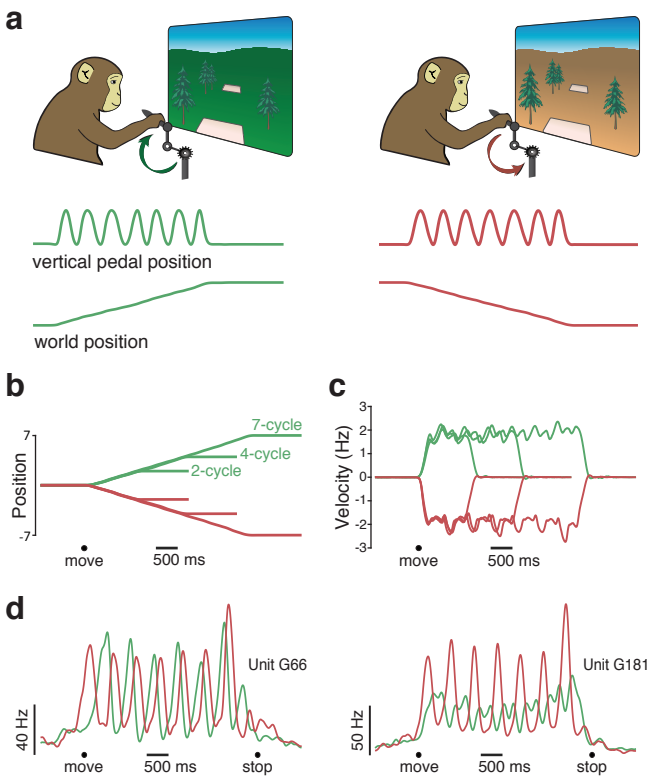

Figure 1. A cycling task that elicits rhythmic movements. (a) Monkeys rotated a hand-held pedal forward (*left*, cued by a green background) or backward (*right*, cued by a tan background) to progress through a virtual environment. Traces at bottom plot pedal kinematics (vertical position) and the resulting virtual world position for two example manual-control trials. On both of these trials (one forward and one backward) the monkey progressed from one target to another by cycling seven cycles. **(b)** Trial-averaged virtual position from a typical manual-control session. Each trace plots the change in virtual position (from a starting position of zero) for one of six conditions: forward or backward for 2, 4, or 7 cycles. Black circle indicates the time of movement onset. Trials were averaged after being aligned to movement onset, and then scaled such that the duration of each trial matched the average duration for that condition. **(c)** Trial-averaged pedal rotational velocity from the same session, for the same six conditions. **(d)** Firing rates of two example units. Trial-averaged firing rates (computed after temporally aligning trials) are shown for two conditions: forward (*green*) and backward (*red*) for seven cycles. Black circles indicate the timing of movement onset and offset.

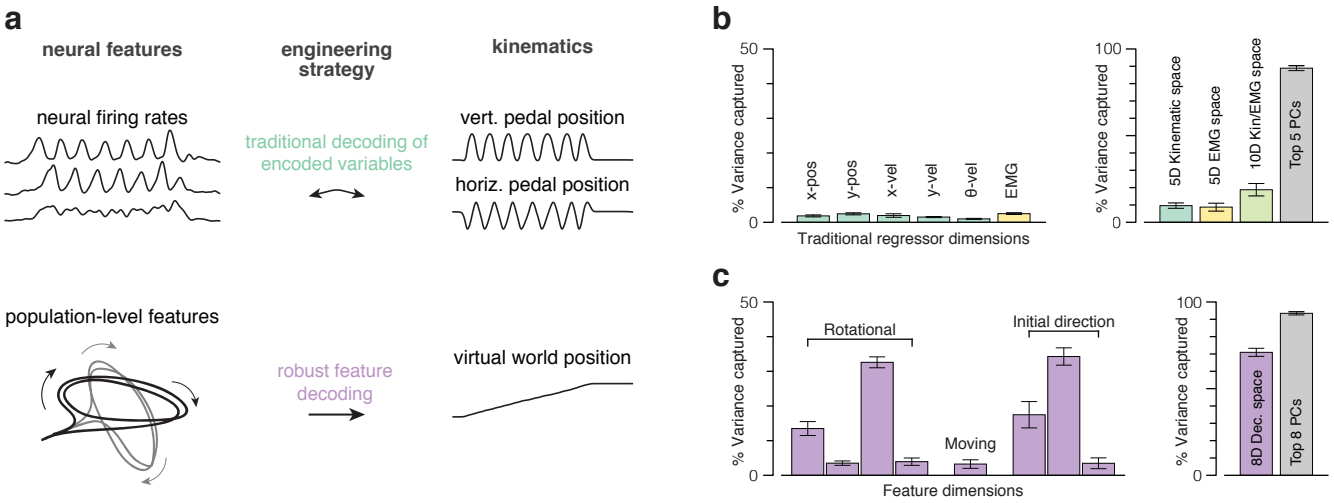

Figure 2. Different decode strategies leverage neural signals with different magnitudes. (a) Two possible decoding strategies. In the first (*top*) neural firing rates are assumed to predominantly encode the key variables. The encoding model is usually assumed to be roughly linear when variables are expressed appropriately. For example, cosine tuning for reach velocity is equivalent to a linear dependence on horizontal and vertical velocity. The goal of decoding is to invert encoding. Thus, decoding dimensions should capture the dominant signals in the neural data (because those are what is encoded). The second strategy (*bottom*) can be applied even if the dominant signals do not have the goal of encoding. This strategy seeks to find neural response features that have a robust relationship with the variable one wishes to decode. That relationship may be complex or even incidental, but is useful if it involves high-variance response features. **(b)** Variance of the neural population response captured by dimensions used to decode kinematic parameters (*green bars*) and muscle activity (*yellow bar*). Data are from three manual-control sessions where units (192 channels per day) and muscles (5-7 channels per day) were recorded simultaneously. Each bar plots the average and standard error across sessions (unless otherwise specified). *Left subpanel*: variance captured for kinematic variables (individual variables shown separately) and muscles (average across 19 recordings, standard error computed across recordings). *Right subpanel*: total variance captured by subspaces spanned by kinematic-decoding dimensions, muscle-decoding dimensions, or both. (These are not the sum of the individual variances as dimensions were not always orthogonal). We had different numbers of EMG recordings per day and thus always selected a subset of five. Variance captured by the top five principal components is shown for comparison. **(c)** Similar plot but for the dimensions upon which our decoder was built. *Left subpanel*: variance captured for each of these eight dimensions. *Right subpanel*: variance captured by the eight-dimensional subspace spanned by those dimensions. Variance captured by the top eight principal components is shown for comparison.

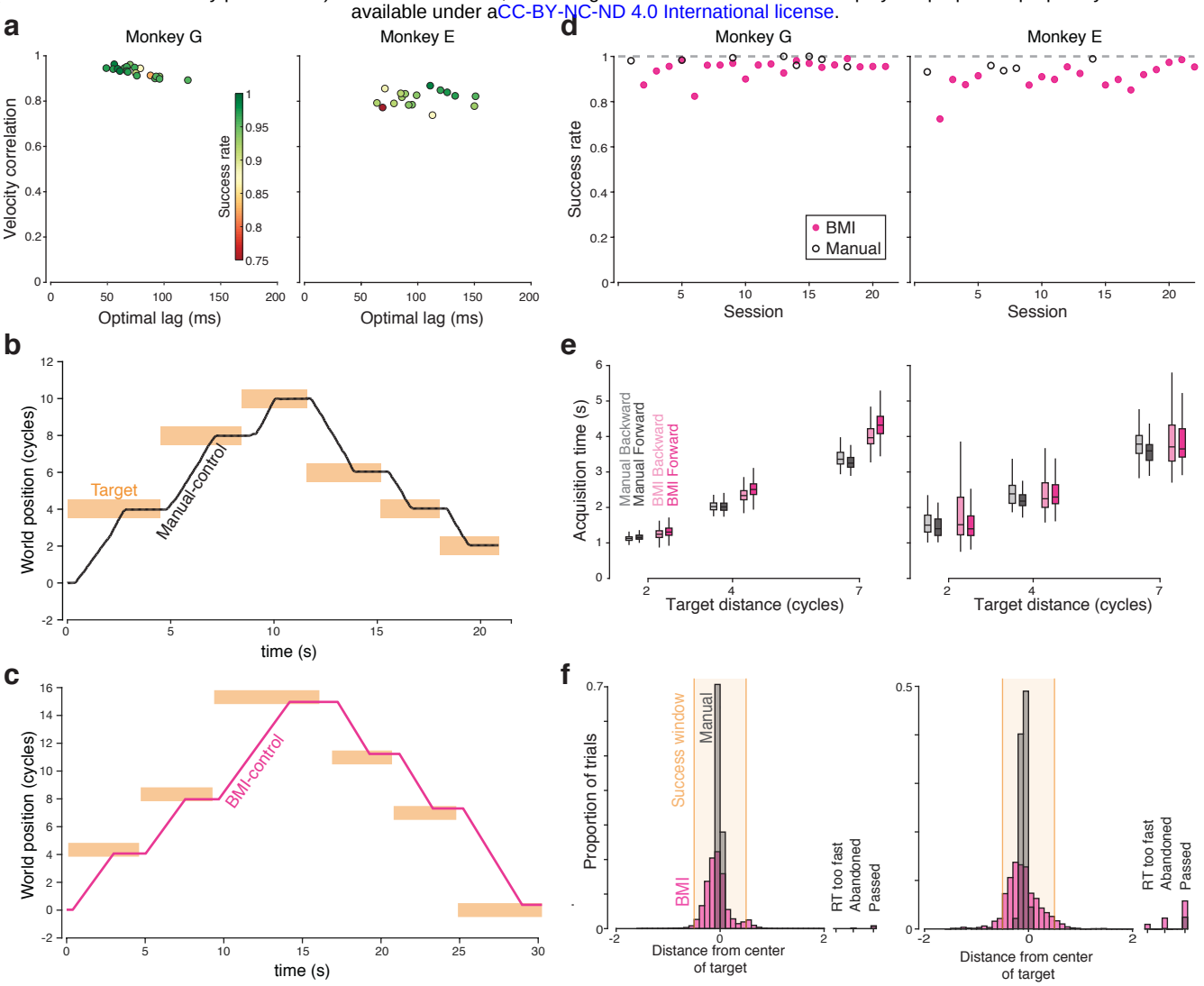

Figure 3. Decoder performance. (a) Summary of the cross-correlation between decoded virtual velocity under BMI control, and the virtual velocity that would have been produced by the pedal (which monkeys continued to manipulate normally). Each symbol corresponds to one BMI-control session, and plots the peak of the cross-correlation versus the lag where that peak occurred. Colors indicate success rate during that session. **(b)** Example manual-control performance for six consecutive trials, 3 forward and 3 backward. World position is expressed in terms of the number of cycles of the pedal needed to move that distance. For plotting purposes, the position at the beginning of this stretch of behavior was set to zero. Bars indicate the time that targets turned on and off (horizontal span) and the size of the acceptance window (vertical span). **(c)** Similar plot during BMI control. For ease of comparison, world position is still expressed in terms of the number of physical cycles that would be needed to travel that far, although physical cycling no longer had any impact on virtual velocity. **(d)** Success rate for both monkeys. Each symbol plots, for one session, the proportion of trials where the monkey successfully moved from the initial target to the final target, stopped within it, and remained stationary until reward delivery. Dashed line at 1 for reference. **(e)** Target acquisition times for successful trials. Center lines indicate median, the box edges indicate the first and third quartiles, and the whiskers include all non-outlier points (points less than 1.5 times the interquartile range from the box edges). Data are shown separately for the three target distances. **(f)** Histograms of stopping location from both monkeys. Analysis considers both successful and failed trials. The bar at far right indicates the proportion of trials where the monkey failed for reasons other than stopping accuracy per se. This included trials where monkeys disrespected the reaction time limits, abandoned the trial before approaching the target, or passed through the target without stopping.

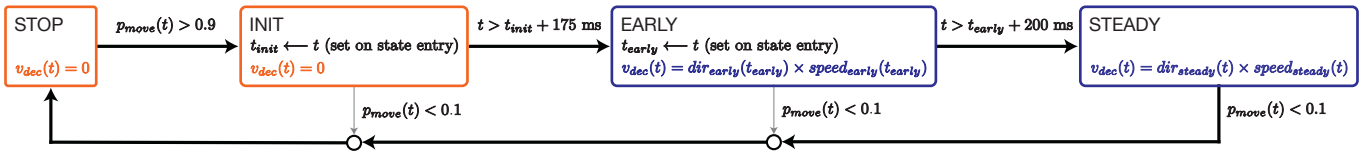

Figure 4. State machine diagram. BMI motion was determined by a state machine with four states: STOP, INIT, EARLY, and STEADY, corresponding to the different stages of a typical trial. The output of the state machine at every millisecond was an estimate of decoded velocity through the virtual environment, v_{dec}, which was then smoothed and integrated to compute virtual position. Black arrows indicate the typical path of a successful BMI trial and gray arrows indicate all other possible transitions. State transitions were governed by activity in the moving-sensitive dimension, which was translated into a probability of moving, p_{move}. While p_{move} was low, the STOP state was active and decoded velocity was set to zero. When p_{move} became high, the INIT state was entered but decoded velocity remained zero. If p_{move} remained high for 175 ms, the EARLY state was entered and velocity was decoded using the initial-direction dimensions. After another 200 ms, the STEADY state was entered and decoded velocity depended on the neural state in the rotational dimensions. If p_{move} dropped below 0.1 at any point, STOP was reentered. States in which progress is made through the virtual environment are highlighted in blue and states in which BMI motion is held at zero are highlighted in orange.

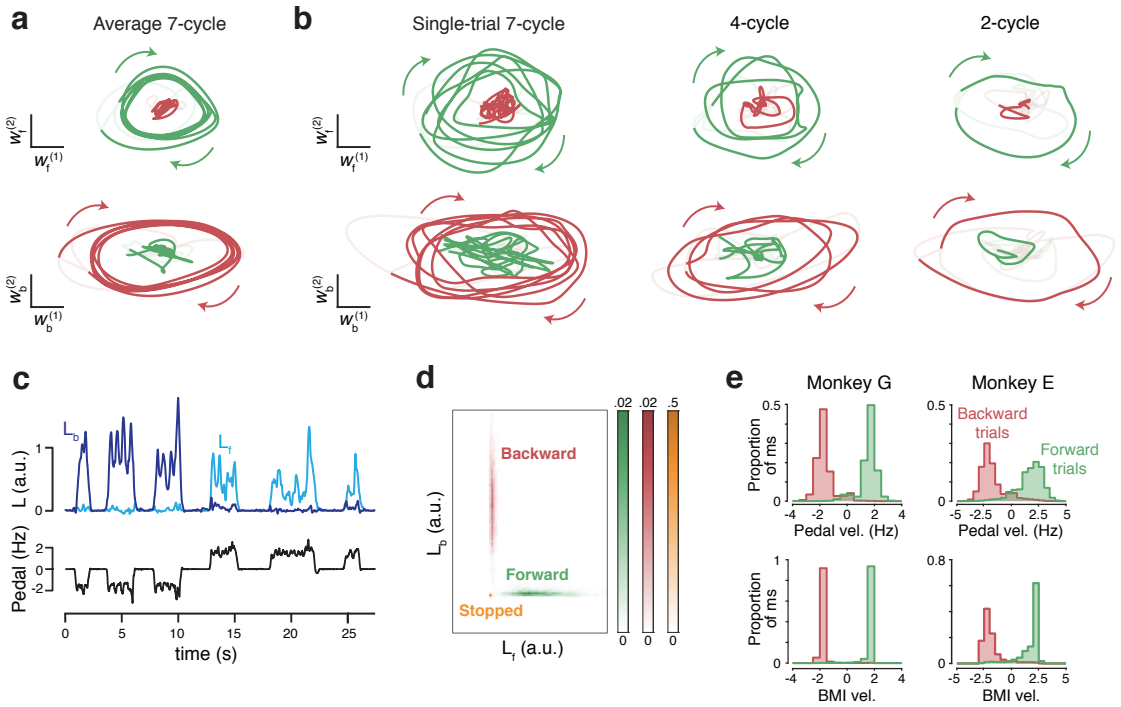

Figure 5. Leveraging rotational trajectories to decode velocity. (a) Trial-averaged population activity, during a manual-control block, projected onto the forward (*top*) and backward (*bottom*) rotational planes. Data are from seven-cycle forward (*green*) and backward (*red*) conditions. By design, the forward plane primarily captures rotational trajectories during forward cycling, and vice versa. Boldly colored portions of each trace highlight rotations during the middle cycles (a period that excludes the first and last half cycle of each movement). Colored arrows indicate rotation direction. Light portion of each trace corresponds to the rest of the trial. In addition to smoothing with a causal filter, neural data have been high-pass filtered to match what was used during BMI control. Data are from monkey G. **(b)** As in panel (a), but for three example single trials, one for each of the three distances. **(c)** Example angular momentum (L) in the backward plane (*dark blue*) and forward plane (*bright blue*) during six trials of BMI control. Velocity of the pedal is shown in black. Although the pedal was disconnected, this provides a useful indication of how the monkey was intending to move. Data are from the same day shown in panels **a** and **b**. **(d)** Probability densities of angular momentums found from the training dataset collected on the same day. **(e)** Histograms of BMI-control velocity (*bottom*) and (disconnected) pedal velocity (*top*) for all times the decoder was in the STEADY state, across all BMI-control sessions.

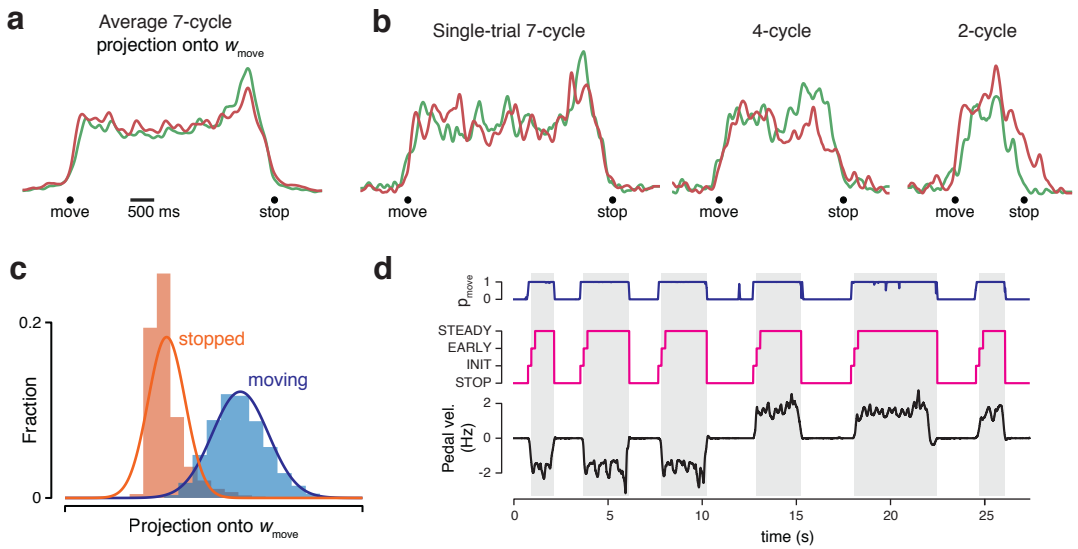

Figure 6. Leveraging the moving-sensitive dimension to infer probability of moving. (a) Trial-averaged population activity, during a manual-control block, projected onto the moving-sensitive dimension (same session and trials as Figure 5a). **(b)** As in panel (a), but for three example single trials (same trials as in Figure 5b). **(c)** Histogram of the neural state projected onto the moving-sensitive dimension for training data. The neural state was measured every ten milliseconds, at times when the monkey was stopped within a target (*orange*) or actively cycling (*blue*). Traces show Gaussian fits used to compute p_{move}. **(d)** Example time-course, during BMI control, of p_{move} (*blue*) and the active state (*magenta*). Gray regions show times when the decoder produced virtual movement (i.e., when in EARLY or STEADY). These times corresponded well to times when the monkey was intending to move, as indicated by the angular velocity of the disconnected pedal (*black*). Note also that transient inappropriate spikes in p_{move} (as seen here around 18 s) do not lead to false starts because either they don't exceed 0.9, as was the case here, or they are too brief and the EARLY state is never reached. Same example data as in Figure 5c.

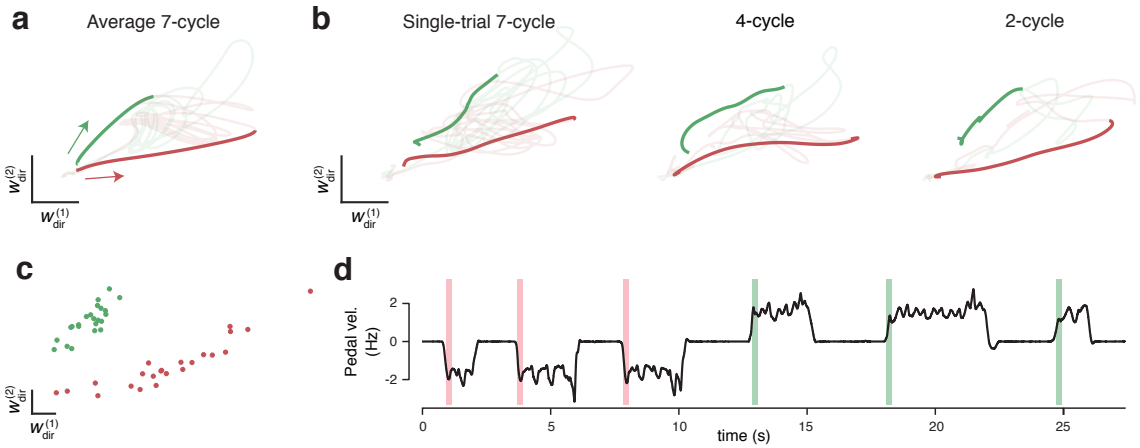

Figure 7. Leveraging initial-direction dimensions to allow low-latency decoding. (a) Trial-averaged population activity, during a manual-control block, projected onto two (of three) initial-direction dimensions (same session and trials as Figure 5a and 6a). Boldly colored portions of traces highlight -200 ms to +175 ms relative to physical move onset. Arrows indicate direction of trajectories. **(b)** As in panel (a), but for three example single trials (same trials as in Figure 5b and 6b). **(c)** The location of the neural state, for training data, at the time the state-machine (applied post-hoc to that training data) entered the EARLY state. This data (50 total trials) was used to fit two Gaussian distributions. During BMI control, when the EARLY state was entered, virtual direction was determined by which distribution maximized the data likelihood. **(d)** Example of initial-direction decoding during BMI control. Colored windows show the times in the EARLY state, with red and green indicating decoded direction. Same example data as in Figure 5c and 6d.

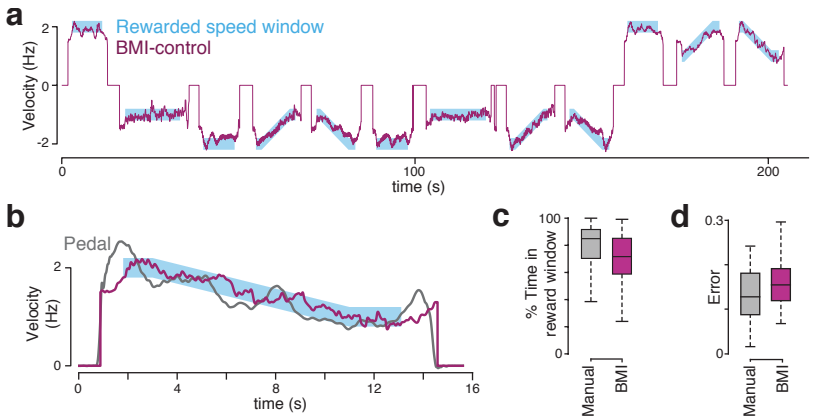

Figure 8. Performance in the modified task requiring speed tracking. (a) Instructed velocity and BMI-decoded virtual velocity during 12 contiguous trials of BMI control. **(b)** Expanded view of one example trial (the last trial from panel a). The virtual velocity that would have been produced by the pedal is shown in gray for comparison. **(c)** Percentage of time spent in rewarded velocity window for trials in manual-control (2 sessions, 333 trials) and BMI-control (3 sessions, 349 trials). Center lines indicate median, the box edges indicate the first and third quartiles, and the whiskers include all non-outlier points (points less than 1.5 times the interquartile range from the box edges). **(d)** Mean absolute error (MAE) between instructed velocity and virtual velocity for both manual control and BMI control sessions. One mean error was computed per trial. Same format as (c).