IOP Publishing Journal of Neural Engineering
J. Neural Eng. **16** (2019) 026011 (14pp) https://doi.org/10.1088/1741-2552/aaf606

Closed-loop cortical control of virtual reach and posture using Cartesian and joint velocity commands

D Young[1,2], F Willett[1,2,3,4], W D Memberg[1,2], B Murphy[1], P Rezaii[3], B Walter[2,5,6], J Sweet[2,7,8], J Miller[2,7,8], K V Shenoy[4,9,10,11,12,13], L R Hochberg[14,15,16,17,18], R F Kirsch[1,2] and A B Ajiboye[1,2,19]

[1] Case Western Reserve University, Cleveland, OH, United States of America
[2] Department of VA Medical Center, FES Center of Excellence, Rehabilitation R&D Service, Louis Stokes Cleveland, Cleveland, OH, United States of America
[3] Department of Neurosurgery, Stanford University, Stanford, CA, United States of America
[4] Department of Electrical Engineering, Stanford University, Stanford, CA, United States of America
[5] Neurology, UH Cleveland Medical Center, Cleveland, OH, United States of America
[6] Neurology, CWRU School of Medicine, Cleveland, OH, United States of America
[7] Neurosurgery, UH Cleveland Medical Center, Cleveland, OH, United States of America
[8] Neurological Surgery, CWRU School of Medical, Cleveland, OH, United States of America
[9] Stanford Neurosciences Institute, Stanford University, Stanford, CA, United States of America
[10] Bio-X Program, Stanford University, Stanford, CA, United States of America
[11] Department of Bioengineering, Stanford University, Stanford, CA, United States of America
[12] Department of Neurobiology, Stanford University, Stanford, CA, United States of America
[13] Howard Hughes Medical Institute, Stanford University, Stanford, CA, United States of America
[14] Center for Neurorestoration and Neurotechnology, Rehabilitation R&D Service, Department of VA Medical Center, Providence, RI, United States of America
[15] School of Engineering, Brown University, Providence, RI, United States of America
[16] Department of Neurology, Massachusetts General Hospital, Boston, MA, United States of America
[17] Department of Neurology, Harvard Medical School, Boston, MA, United States of America
[18] Carney Institute for Brain Science, Brown University, Providence, RI, United States of America

E-mail: aba20@case.edu

Received 12 July 2018, revised 3 December 2018
Accepted for publication 4 December 2018
Published 29 January 2019

Abstract

Objective. Brain–computer interfaces (BCIs) are a promising technology for the restoration of function to people with paralysis, especially for controlling coordinated reaching. Typical BCI studies decode Cartesian endpoint velocities as commands, but human arm movements might be better controlled in a joint-based coordinate frame, which may match underlying movement encoding in the motor cortex. A better understanding of BCI controlled reaching by people with paralysis may lead to performance improvements in brain-controlled assistive devices. *Approach.* Two intracortical BCI participants in the BrainGate2 pilot clinical trial performed a visual 3D endpoint virtual reality reaching task using two decoders: Cartesian and joint velocity. Task performance metrics (i.e. success rate and path efficiency) and single feature and population tuning were compared across the two decoder conditions. The participants also demonstrated the first BCI control of a fourth dimension of reaching, the arm's swivel angle, in a 4D posture matching task. *Main results.* Both users achieved significantly higher success rates using Cartesian velocity control, and joint controlled trajectories were more variable and significantly more curved. Neural tuning analyses showed that most single feature activity was

[19] Author to whom any correspondence should be addressed.

best described by a Cartesian kinematic encoding model, and population analyses revealed only slight differences in aggregate activity between the decoder conditions. Simulations of a BCI user reproduced trajectory features seen during closed-loop joint control when assuming only Cartesian-tuned features passed through a joint decoder. With minimal training, both participants controlled the virtual arm's swivel angle to complete a 4D posture matching task, and achieved significantly higher success using a Cartesian + swivel velocity decoder compared to a joint velocity decoder. *Significance.* These results suggest that Cartesian velocity command interfaces may provide better BCI control of arm movements than other kinematic variables, even in 4D posture tasks with swivel angle targets.

Keywords: BCI, intracortical brain computer interface, neural decoding, motor control

S Supplementary material for this article is available online

(Some figures may appear in colour only in the online journal)

1. Introduction

Brain–computer interfaces (BCIs) decode recorded neural activity into command signals for external effectors, such as virtual cursors [1–4], robotic arms [1, 5–8], or functional electrical stimulation (FES) systems [9, 10]. BCIs are a promising command interface for high degree of freedom tasks (e.g. reach and grasp), because they can simultaneously decode multi-dimensional commands with low cognitive burden. Additionally, BCIs are capable of extracting user intention even in the absence of voluntary movement, which is an important criteria for assistive devices aiding the millions of people with tetraplegia resulting from spinal cord injury (SCI), amyotrophic lateral sclerosis (ALS), and other chronic neurological impairment [11].

Previous BCI studies primarily decoded Cartesian endpoint velocity commands as inputs for various effectors. However, previous studies suggest that neural activity may be tuned to joint space kinematics during natural reaching movements [12–15]. Therefore, direct decoding of joint velocities could result in better closed-loop BCI performance for the control of human arm movements. Additionally, joint velocity control is also an attractive interface for current assistive devices, such as FES systems, because it simplifies the requirements of a controller for mapping commanded kinematics to electrode stimulation parameters and more closely matches existing FES command interfaces [10, 16].

Many studies have attempted to characterize tuning properties of the motor cortex during non-human primate able-bodied reaching experiments. Across a variety of tasks and paradigms, results show that multiple movement parameters can be decoded from motor cortical areas, including Cartesian arm endpoint position, movement direction, movement speed, and joint velocities [13, 17–19]. These results may be expected, as coordinating limb movements involves a number of sensorimotor transformations (i.e. from Cartesian to joint kinematic coordinates, joint kinematics to arm dynamics, and from dynamics to muscle activations) [20]. Additionally, evidence shows that population and single feature activity tends to be tuned to multiple degrees of freedom across coordinate systems, rather than having information limited to a single movement parameter [13, 17–19, 21–23]. However, fMRI comparisons of motor imagery evoked neural activity suggest that tuning to attempted movements may change after paralysis, possibly due to network reorganization in the absence of sensory input [24]. Single-joint movements have been reliably decoded from human intracortical recordings years after paralysis, but only using offline decoding of open-loop movements about a single joint at a time [25]. It remains to be shown if persons with tetraplegia are capable of closed-loop BCI control of reaching movements using joint-based commands, and how joint-based performance would compare to standard Cartesian velocity control.

In addition, typical Cartesian velocity decoders manipulate the 3D endpoint of the arm, but constrain the fourth dimension of arm position, the swivel angle [26]. The swivel angle describes the rotation of the arm around a virtual axis that connects the shoulder and the wrist, and is used to specify the position of the elbow for a given arm endpoint location [27, 28]. Arm control using Cartesian coordinates therefore needs to either set the swivel angle to a constant value, or estimate a likely natural angle based on the current arm endpoint. However, giving users control over the full posture of the arm (rather than constraining the swivel angle) could enable more natural movements and improve obstacle avoidance compared to typical endpoint-only control. Closed-loop control of the swivel angle is achieved with 4D joint-velocity control, but can also be accomplished through combining 3D endpoint control with direct decoding of the swivel angle in a Cartesian + swivel velocity decoder. It is unknown whether 4D postural control of the arm endpoint and swivel angle would be superior using a joint-based or Cartesian + swivel based decoder.

In this study, we directly compare closed-loop BCI control of reaching using decoded Cartesian and joint velocities. Two participants enrolled in the BrainGate2 pilot clinical trial performed anthropomorphic virtual reaching tasks using two kinematic coordinate frames. Task performance metrics, including success rate and path efficiency, as well as single feature and population level neural tuning results were compared between the two decoder conditions. This study also demonstrates the first closed-loop BCI control of arm swivel

angle in a 4D posture matching task, using both joint and Cartesian + swivel angle decoders.

2. Methods

2.1. Permissions and participants

Permission for these studies was granted by the US Food and Drug Administration (Investigational Device Exemption #G090003) and the Institutional Review Board of University Hospitals Cleveland Medical Center (#04-12-17), Massachusetts General Hospital (2011P001036), and Stanford University (#20804). The participants were enrolled in a pilot clinical trial of the BrainGate Neural Interface System (ClinicalTrials.gov, NCT00912041). (Caution: Investigational device. Limited by federal law to investigational use.) Informed consent, including consent to publish, was obtained from the participants prior to their enrollment in the study.

This study includes data from two participants with chronic tetraplegia: T8, a 55 year-old right-handed male with C4 level AIS-A SCI and T5, a 63 year-old right-handed male with C4 level AIS-C SCI. Both participants received two 96-channel intracortical microelectrode arrays (1.5 mm electrode length, Blackrock Microsystems, Salt Lake City, UT, USA) placed in the hand/arm area of the dominant precentral gyrus [10, 29].

2.2. Neural signal processing

Each intracortical array was attached to a percutaneous pedestal connector on the head of the participant. Patient cables connected the pedestals to amplifiers (Blackrock Microsystems), which filtered (0.3 Hz–7.5 kHz bandpass) and digitized (30 kHz) the neural signals. Real time processing was performed in Simulink on the xPC real-time operating system (The Mathworks Inc., Natick, MA, USA). The digitized signals were downsampled to 15 kHz, then filtered (8th order non-causal Butterworth filter) between 250 and 5000 Hz. From this spike band signal, two features per electrode were extracted: threshold crossing rate (TX) and high-frequency spike power (HFSP). TX was computed by counting the number of non-spike sorted negative threshold crossings per time bin, where the threshold was set as −4.0 times the RMS amplitude of the channel during a one-minute reference block collected at the beginning of the session. HFSP was computed by taking the root mean square of the filtered spike band (250–5000 Hz) voltages. These features were computed in 20 ms non-overlapping bins (50 Hz update rate), then passed to a decoder to generate the commanded velocities. Offline analyses were performed with the same features that were computed online (2 arrays/participant $*$ 96 channels/array $*$ 2 features/channel = 384 features/participant).

For decoder building, a variant of optimal linear estimation (OLE) was used to produce a decoding matrix D that mapped the neural features to intended velocities [10]. The decoder was applied at each timestep using the equation: $u_t = Df_t$, where the decoder matrix D (size $M \times 384$) was applied to a vector f_t of neural features (size 384×1), yielding the commanded velocity u_t (size $M \times 1$, where M is the number of decoded dimensions: 3 for Cartesian or 4 for joint/Cartesian + swivel). For online control, this command vector was then smoothed with a first-order low-pass filter [10]; for offline analysis, only the unsmoothed vector u_t was considered. The command vector smoothing filter was of the form $v_{t+1} = \alpha v_t + (1 - \alpha)$ Bu_t, where v_t is the smoothed control vector at time step t, α is a scalar smoothing parameter (set to 0.96), and B is a diagonal gain matrix that specifies the gain of each dimension. A 400 ms reaction time interval after target presentation was excluded from the training data.

2.3. BCI tasks

Participants used a 3D monitor to complete virtual reality reaching tasks with a first-person view of right-arm movements. Participant T5 wore active 3D glasses while performing tasks; participant T8 was more comfortable without them, but had trained with the visualization enough to be familiar with its 2D depth cues (i.e. shadows). The main task used for evaluation of Cartesian and joint velocity control of BCI arm movements was a standard center-out reaching task. In this 3D task, the user controlled virtual reality arm movements to reach targets located in the workspace. Endpoint target regions were visible as a colored ball located at the wrist of a translucent 'ghost' arm (figure 1(A)) and were defined by a 3D endpoint and a target radius (22% target distance) without any enforced postural requirements (i.e. any swivel angle/joint angle combination that kept the endpoint within the target was allowed). When only Cartesian endpoints were specified, the virtual arm's swivel angle was solved according to the methods in (Tolani, Goswani, and Badler, 2000) [30]. Trials began with a random instructed delay period (0.25–0.75 s) where the arm was held frozen in place (no decoded velocities were applied to the arm) but the target was identified. A go cue then signaled the start of the reach period with an audible sound and a change in the color of the ball at the wrist of the controlled arm. Trials were considered successful if the user-controlled arm overlapped with the target region for a continuous 500 ms 'hold period', and unsuccessful if the hold period was not maintained before an imposed maximum trial time of 10–12 s. Regardless of trial success, the virtual arm was reset to the target location before the next trial began.

Two additional tasks were used for the evaluation of the participants' ability to control the arm's swivel angle (figure 1(B)). First, swivel control was tested in a 1D task, where the endpoint of the arm was fixed while the user controlled only the swivel angle to reach posture targets designated by the forearm angle of the ghost arm (figure 1(C)). Second, participants completed a 4D postural task which combined simultaneous endpoint and swivel angle control with sequential, independent (translation-only and swivel-only) targets (supplementary video 1 (stacks.iop.org/JNE/16/026011/mmedia)). In this task, a trial was only considered successful if both the endpoint and the swivel angle were within a designated target region, defined by an endpoint (with the same dimensions as the as in the 3D endpoint task) and a swivel target region (±9 degrees centered on swivel angle targets of 15, 50, and 85 degrees). However, a new trial's target posture

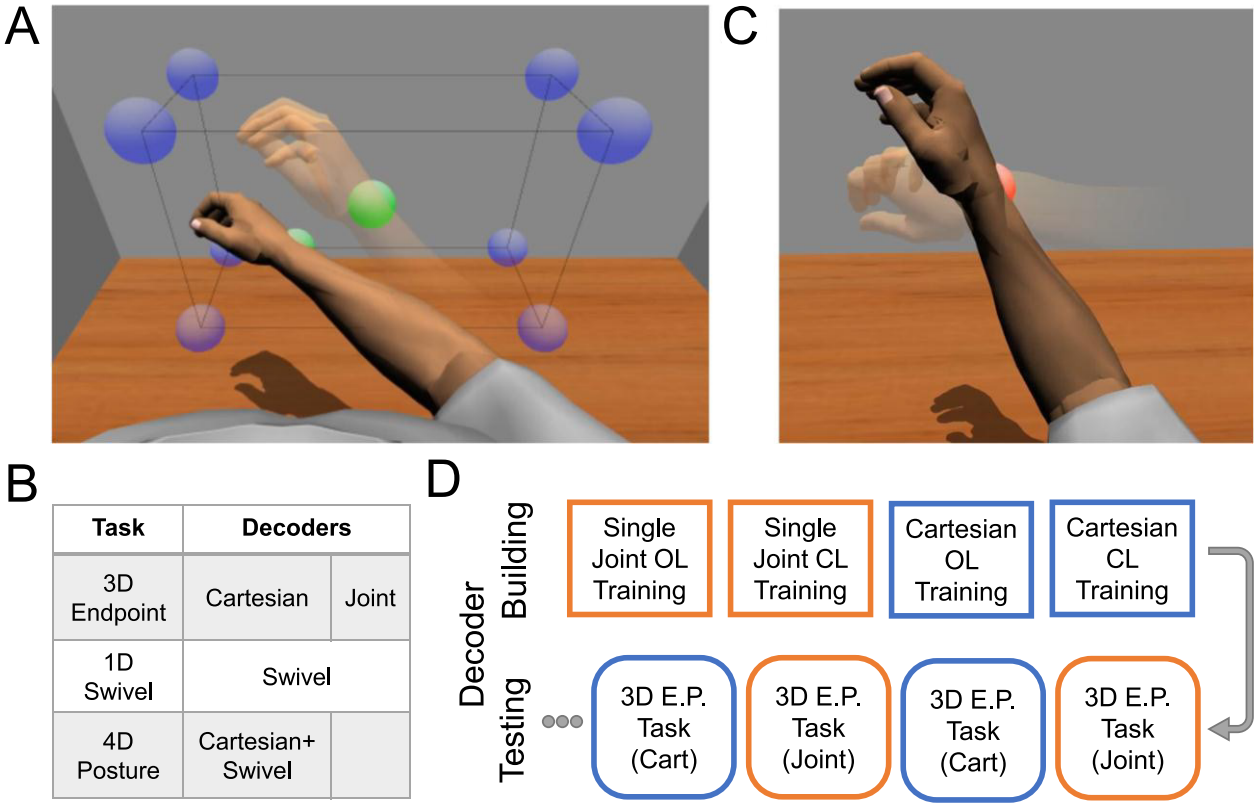

Figure 1. Virtual reality task visualization. (A) Participants used a first-person 3D virtual reality game to perform reaches by commanding either the Cartesian endpoint velocities or joint velocities of the opaque avatar arm. (B) A table of the decoders used for each of the three tasks. Primary comparisons between Cartesian and joint decoders were conducted in the 3D endpoint task. Control of the swivel angle was trained and tested in a 1D swivel game, then compared to joint control in a 4D posture matching task. (C) In two tasks, participants controlled the swivel angle of the arm by matching the opaque forearm angle to the posture of the translucent ghost arm. (D) A typical session protocol, starting with calibration blocks to 'build' both a joint and Cartesian decoder, then performing interleaved 'testing' with those decoders in a 3D endpoint task.

shared with the previous trials posture either a swivel angle (for translation-only trials) or endpoint (for swivel-only trials). This task structure allowed analysis of the independence of control between, and relative performance of, the translation and swivel dimensions.

2.4. Decoder calibration procedure

Each session began by training decoders using an open-loop and closed-loop recalibration protocol figure 1(D)). For Cartesian filter building, a typical 3D center-out task was used, but the targets were placed at the vertices of a cube so that equal durations of training data would be collected in each of the three Cartesian dimensions. For joint velocity decoders, training data was obtained in a sequential single-joint task, where movements were made about a single degree of freedom at a time through its range of motion (four total dimensions were trained: shoulder plane of elevation, shoulder angle of elevation, shoulder rotation, and elbow flexion). Each decoder building process began with a single block of open-loop training, where the user watched and attempted to mimic the scripted movements on the screen. An initial decoder was built from the open-loop data and then used to start the second phase of training, closed-loop recalibration [31]. In this phase, 2–4 blocks of closed-loop neural control were performed, with the decoders updated to include the new training data after each block. Additionally, to keep success rates at a user-acceptable level and improve the quality of control during training, computer assistance was used where movements perpendicular to the target direction were attenuated by decreasing amounts from 80% to 40% [31]. Computer assistance was only used during decoder calibration blocks and was removed after final decoder building. In sessions where the 4D posture matching task was performed, Cartesian + swivel decoders were built using the standard open-loop then closed-loop recalibration procedure in the 4D posture matching task.

2.5. Experimental design

After completing decoder training, the users performed decoder comparison blocks in either the 3D endpoint or 4D posture matching task. Either the Cartesian or the joint velocity decoder was randomly chosen for the first block (3–4 min of closed-loop control without computer assistance), and then the decoders were alternated each block until 4–6 blocks were collected for each condition. The users were informed prior to starting a block which decoder they would be using and were allowed to take different strategies if they chose. However, both participants reported attempting the same control strategy, regardless of the decoder condition. In

all tasks, the users were instructed to try and make straight movements to the targets, but no path metrics were enforced for trial success. T8 completed 3 sessions of interleaved decoder comparison ($n = 32$ blocks), while T5 completed 2 sessions ($n = 18$ blocks).

Additionally, participant T8 conducted three sessions in which he only used a joint velocity decoder. The goal of these sessions was to test if T8 was able to improve his performance (success rate and path efficiency) through focused short-term practice with the joint velocity decoder. These sessions similarly began with the typical joint velocity decoder building paradigm, followed by 6–15 blocks of joint control in the 3D endpoint task.

2.6. Reach performance metrics

Performance in the virtual reality tasks was primarily measured as success rate, or the percentage of trials that were successfully completed within the block. A trial was considered successful if the user held all controlled dimensions within the target intervals for a hold-time of 500 ms, and unsuccessful if the arm did not complete the hold-time before a timeout period (10–12 s). Success rates were compared between conditions using the Fisher exact test with an alpha of 0.05.

Trajectories were compared with two quantitative metrics, path efficiency and path curvature [32]. Path efficiency, a standard BCI metric, captures inefficiencies in closed-loop control that can result from indirect or noisy movements, while path curvature directly measures how far the actual path deviated from the straightest possible path. Path efficiency was calculated as the ratio of shortest path length possible to the path length actually taken (with a maximum of 1, and decreasing values for less efficient paths). Path curvature was calculated as the maximum distance of the actual path from the straightest path (distance calculated at each timestep, maximum chosen per trial), divided by the path length of a straight-line trajectory (with a minimum of 0, and increasing values for more curved paths). Path efficiency and curvature were compared across all trials between the two decoder conditions using two-sample t-tests for equal means with alpha $= 0.05$.

Average trajectories were calculated per target direction by oversampling each trajectory (described by the position of the endpoint through time) to 200 data points. Trajectories which ended within 1.3× the target radius were then averaged to obtain an average trajectory, which was then downsampled to 30 equally spaced points. Additional processing was performed to calculate trajectory statistics (e.g. confidence intervals) as a function of space rather than of time, thus accounting for reaches with similar paths but varying speed or durations [33]. At each point of the average trajectory, a plane perpendicular to the movement was constructed by defining a normal vector that points along the trajectory path. For each trial, all intersection points were found between the trial trajectory and the average trajectory normal planes. Within each plane, spatial variability across trials could then be described by the eigenvalues of the covariance matrix of the intersection points [34]. Thus, this procedure yields an average trajectory with error described in 3D spatial coordinates, which more accurately accounts for path variability when individual trials have varying path lengths or temporal durations [33]. For 2D plotting, the 3D spatial-errors were projected onto the 2D viewing plane and connected to form a shaded confidence region.

2.7. Neural information analyses

Neural activity was analyzed in two main ways: fitting single feature encoding models, and modeling population activity using demixed principal components analysis (dPCA) [35, 36].

For single feature analyses, neural tuning assumptions were compared by fitting encoding models that predicted feature activity using intention vectors constructed from kinematics, and evaluating model fit using R^2. Three different kinematic models were considered: (1) a 'Cartesian position error' model in which the assumed command vector is the difference between current endpoint and target endpoint in 3D Cartesian space, (2) a 'joint position error' model in which the assumed command vector is the difference between current joint angles and target joint angles in 4D joint space, and (3) a 'coordinated joint model' in which the assumed command vector represents 4D joint angles which produce coordinated movements that are straight in 3D Cartesian space. For the coordinated joint model, an inverse kinematic model was used to calculate the desired joint angles at a waypoint that lies a small distance further on the Cartesian straight-line path towards the target. The model's final command vector was the difference between the current joint angles and the calculated joint angles at the straight-line waypoint. 10-fold cross-validated encoding models were built for each kinematic assumption and single feature. Encoding model R^2 was averaged separately for each decoder condition (closed-loop Cartesian or joint velocity decoder) within each session.

For population analyses, a dimensionality reduction technique, demixed principle component analysis (dPCA) recently developed by Kobak *et al*, was used to investigate neural activity differences between decoder conditions [35]. dPCA is a linear dimensionality reduction technique that optimizes each dimension to explain neural variance that is specifically related to one or more task 'factors'. Two primary task-related factors were examined: target direction and decoder condition. The goal of this analysis was to investigate whether there were large decoder-dependent or direction-and-decoder dependent dimensions of neural activity that would be indicative of the user adapting their closed-loop command signals to the decoder type used during that block. Within each session, the neural responses were averaged according to target direction separately for blocks where the participant was using Cartesian and joint velocity control. dPCA then generated the top 20 dimensions which best separated neural activity according to the task factors: target direction, decoder condition, direction-decoder interaction, and variance that was consistent across all trials (condition-independent). To improve dPCA interpretability, neural features were standardized across blocks through z-scoring, then Gaussian kernel smoothed (50 ms), and finally reduced to include only the top

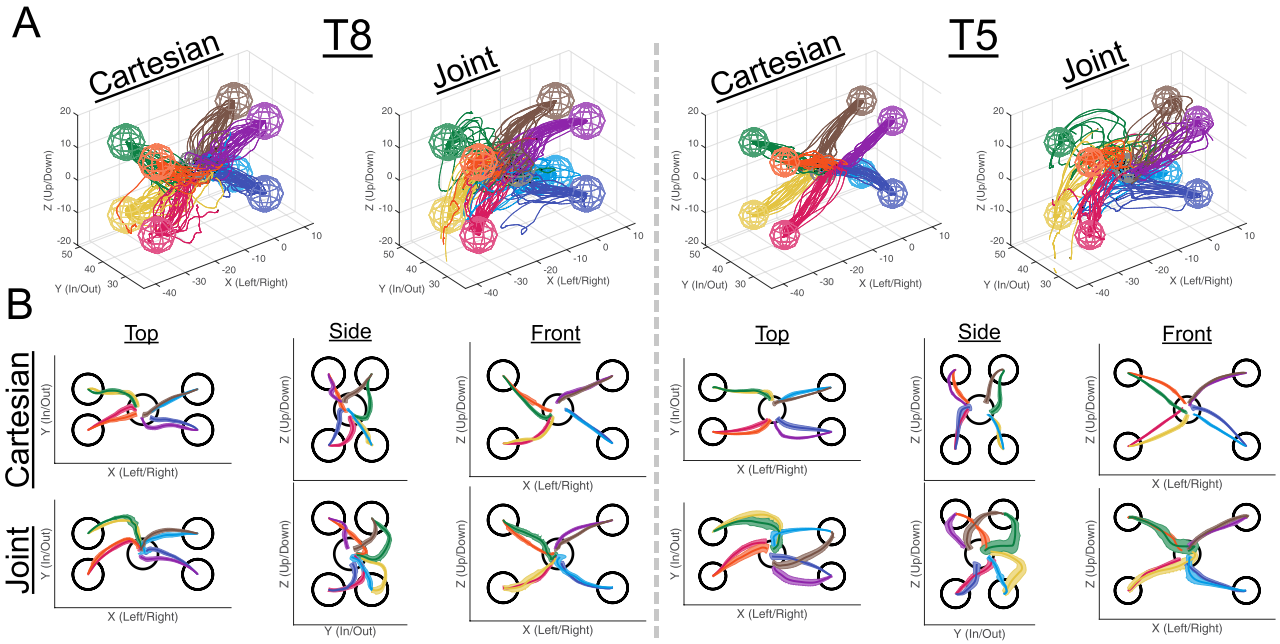

Figure 2. 3D Endpoint trajectories using Cartesian and joint velocity decoders. (A) Closed-loop 3D endpoint trajectories from the 8 outer targets to the center target for both decoder conditions and participants (T8: Cartesian $n = 196$, Joint $n = 199$. T5: Cartesian $n = 157$, Joint $n = 120$). (B) 2D projections of the average trajectories for each direction (solid lines) and 95% confidence interval (transparent patches). Target regions are shown as black circles.

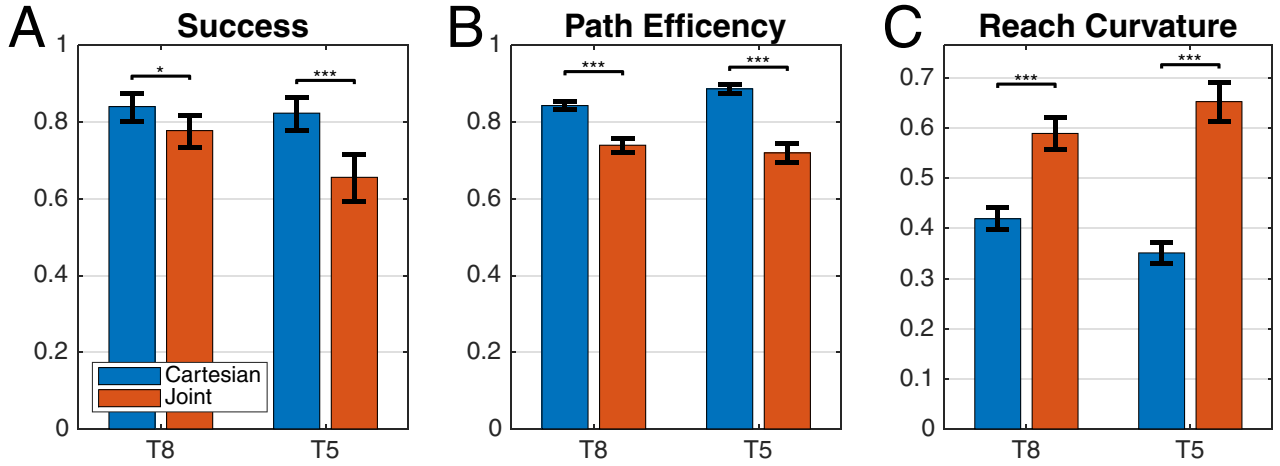

Figure 3. 3D Endpoint task performance metrics. Task performance metrics ((A) success, (B) path efficiency, (C) reach curvature) for both participants in the 3D endpoint task using Cartesian and joint velocity decoders. Significance levels: $^{*}\ p < 0.05$, $^{***}\ p < 0.001$.

140 features (sorted by highest mean trial-averaged-response across all conditions). Additionally, the analyses were limited to the time period starting 1 s prior to the go-cue (therefore including delay period activity) and ending 2 s into the reach period.

A second population analysis method, decoder signal-to-noise ratio (SNR), was used to measure changes in decoder accuracy within and across the three joint-velocity only sessions performed by T8. First, the decoded velocity was modeled as the sum of an 'intended velocity' and noise, where the intended velocity is derived from a user-specific model of closed-loop BCI control described in [37, 38]. The model assumed the intended velocity, c_t, was a position error vector scaled by a non-linear function of target distance:

$$c_t = \frac{g_t - p_t}{||g_t - p_t||} f_{\text{targ}}\left(||g_t - p_t||\right),$$

where g_t is the target position, p_t is the cursor position, and f_{targ} is a nonlinear, scalar weighting function that is fit empirically to each user's closed-loop kinematic control using 4-fold cross validation. A full model of user intention was first developed in [39], which included additional terms for user feedback delay and cursor velocity. The simplified model used for this analysis (including only f_{targ}) was also used for decoder SNR calculations in [37, 38], and is appropriate whenever the smoothing dynamics are moderate and the cursor velocity term is not needed. Decoder SNR was calculated as the magnitude of the intended velocity divided by

the standard deviation of the decoder noise and averaged over all trial timesteps between a 400 ms reaction time and the time of first target touch.

2.8. BCI user simulator

A feedback control model of a BCI user was used to understand how task performance would be expected to change under different neural tuning assumptions. To determine whether the curvature observed during joint-control blocks was due to the user's neural features being tuned to Cartesian velocity, the model was used to simulate trajectories in the 3D center-out task that would result when a joint decoder was applied to completely Cartesian-tuned neural activity. First, we used the methods in [39] to estimate decoding noise and control policy parameters from real data recorded from participant T5 in a typical closed-loop block. This step resulted in a model of T5 that outputs a simulated Cartesian velocity command at each time step with realistic decoding noise. Then, we built a linear decoder to convert the Cartesian velocity commands to joint velocity commands with minimum squared error by using linear regression to predict the joint velocity kinematics observed on a typical decoder training block from the Cartesian velocity commands given by the feedback control model. Doing so simulates decoding joint velocities from purely Cartesian-tuned neural activity. Finally, the model and decoder were then used to simulate closed-loop trajectories in the 3D endpoint task. The simulated trajectories were compared to actual closed-loop trajectories performed by T5 during joint-control blocks. The simulated trajectories provide an estimate of the curvature and variability that would be expected when using a mismatched decoder in closed-loop, helping to interpret the curvature we observed.

3. Results

3.1. Task performance

Both participants were able to perform the 3D endpoint task using the Cartesian and the joint velocity decoders. Successful trial times (mean $\pm$ standard deviation, beginning at the 'go cue' and ending after the 500 ms 'hold time') were 4.7 ± 1.6 s for participant T8 and 5.5 ± 2.2 s for participant T5. Figure 2(A) shows all of the inner (towards the center target) trajectories for each decoder condition and participant (supplementary figure 1 shows the reach trajectories towards outer targets). Both participants achieved excellent performance using the Cartesian velocity decoder, denoted by straight, consistent reaches towards the target. However, the trajectories during joint velocity decoding blocks were visibly curved and less consistent. Figure 2(B) shows the 2D projections of the target-averaged trajectories and 95% confidence interval of the mean. The trajectory curvature during joint velocity control is especially evident in the 'Top' and 'Side' views, where the reaches tend to deviate from a straight path in the Y dimension. Figure 2(B) also shows that joint-controlled trajectories are less consistent, as the confidence intervals are much larger

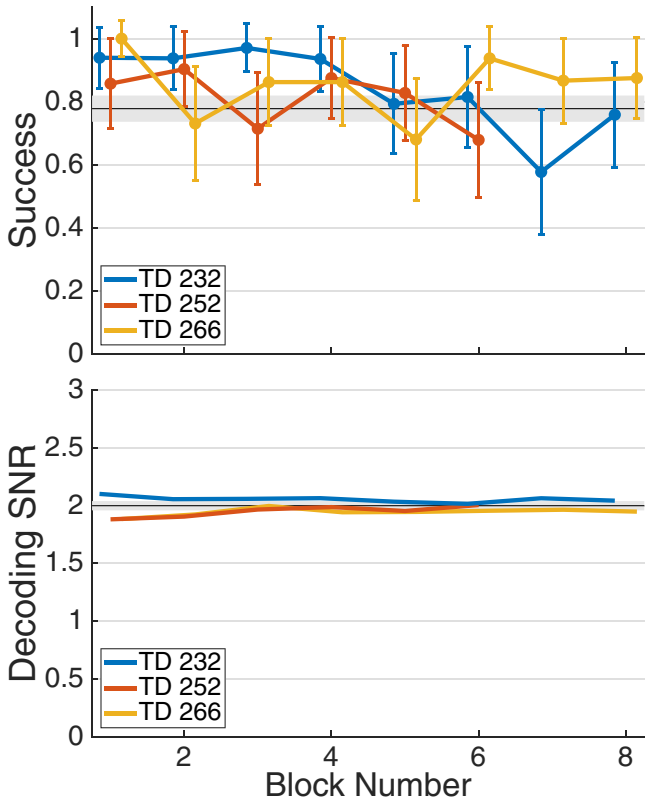

Figure 4. Joint learning session task performance. Task performance metrics (success rate and decoding SNR) are tracked across blocks during T8's three sessions of focused practice using joint control (trial days (TD): 232, 252, and 266 shown separately as colored traces). The average and 95% confidence interval for the typical value of each metric (calculated on the three sessions of interleaved joint and Cartesian control) are shown as a black line and patch.

than Cartesian-controlled reaches. Similar trends in curvature and consistency between joint and Cartesian control can be seen in the outer reaches, shown in supplementary figure 1.

Figure 3 shows the quantified 3D endpoint task performance for each condition. Both participants were able to complete the task with good performance using a Cartesian decoder (success rates $>80\%$), but achieved significantly lower success when using the joint velocity decoder (T8: $p < 0.05$, T5: $p < 0.001$). Additionally, both participants performed reaches that were significantly more efficient (figure 3(B), $p < 0.001$) and straighter (figure 3(C), $p < 0.001$) when using the Cartesian velocity decoder. Path metrics were also computed in 4D 'joint space', measuring the efficiency and curvature compared to a constant, linear path in joint space from the start of the movement to the target angles (determined at trial completion). Both metrics yielded similar results, where trajectories during Cartesian velocity control were significantly more efficient and less curved even when analyzed in 4D joint space (supplementary figure 2).

3.2. Joint-only learning sessions

Participant T8 performed three additional sessions to test if focused practice using only a joint velocity decoder would

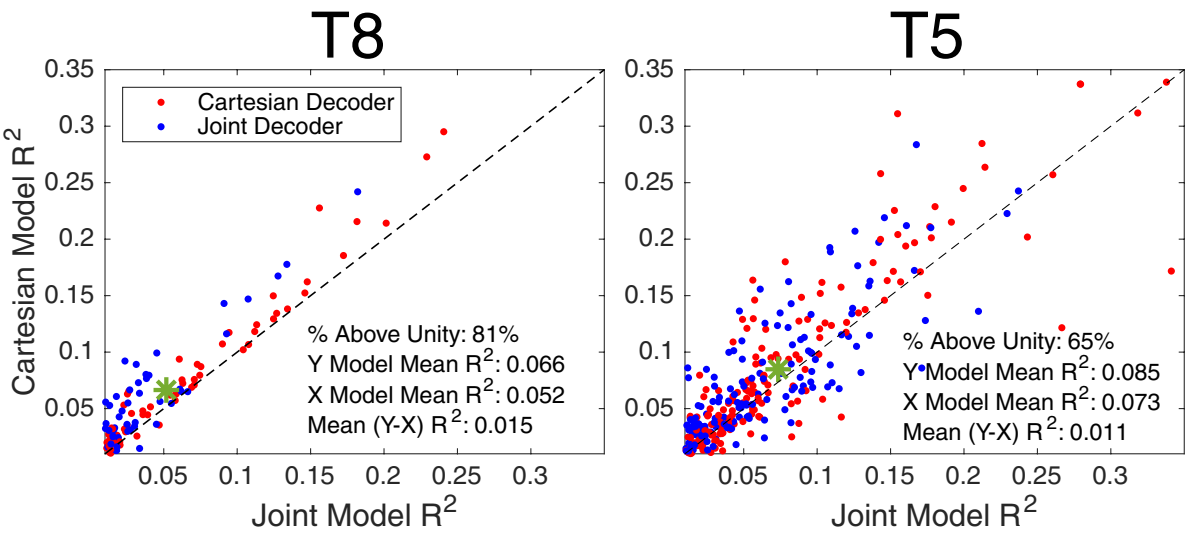

Figure 5. single feature encoding model performance. Pairwise scatterplot of encoding model R^2, averaged separately per decoder condition (red/blue) within each session, comparing the (1) 3D Cartesian position error to the (2) 4D joint position error model. Only features with model R^2 above 0.01 are included. Average model R^2, mean R^2 difference between models, and % of features which are better explained by the Cartesian model are reported for each participant. Green star shows the average performance of all plotted features.

enhance performance compared to the three sessions of interleaved joint and Cartesian velocity control. Figure 4(A) shows the success rate within and across each of the three joint-only sessions. As a baseline, typical performance during interleaved decoder sessions (average and 95% confidence interval) is shown in black. There were no significant changes in success rate within each session or across the sessions when compared to typical performance. Additionally, there were no significant differences in joint decoding SNR across sessions, indicating no change in neural information representing the desired joint velocities (figure 4(B)). Supplementary figure 3 shows two additional metrics tracked during these sessions, path efficiency and Cartesian decoder SNR. These two metrics also fail to show any significant differences in performance within these joint-only sessions as compared to the interleaved decoder testing sessions.

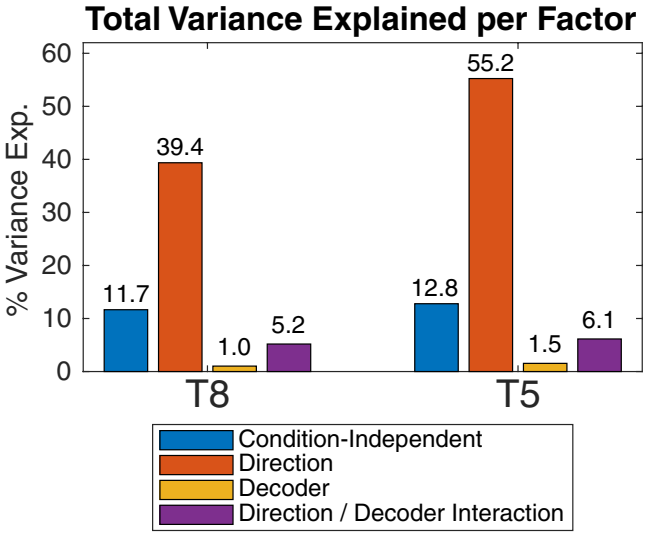

Figure 6. dPCA total variance explained per factor. dPCA results for each participant show the percent of total variance in the neural population activity explained each of the four task-specific factors summed across the top 20 dPCA dimensions and averaged across sessions (T8: $n = 3$, T5: $n = 2$).

3.3. Neural analyses

Figure 5 shows a pairwise comparison of the Cartesian and joint position error single-feature encoding models. Encoding model performance was averaged across blocks, but separated by decoder condition (closed-loop Cartesian velocity or joint velocity decoder) and session day. Both participants had more features which were explained better by the Cartesian model than the joint model (T8: 81%, T5: 65%), and had higher overall mean R^2 values with the Cartesian model. Additionally, there was no discernable pattern between decoder condition (color), indicating that single feature encoding was mostly similar regardless of the current method of control. Supplementary figure 4 shows the pairwise comparisons between (A) the Cartesian and coordinated joint models, and (B) the comparison between the joint position error and coordinated joint models. Similarly, the Cartesian model better explained the activity of more features than the coordinated joint model (T8: 85%, T5: 72%). For both participants, there were only slight differences in performance between the two joint models (supplementary figure 4(B)).

To investigate how much the participants varied their population-level neural activity between the two decoder conditions, a dimensionality reduction technique was used to find dimensions that best explain variance related to the target direction or decoder condition. dPCA analysis linearly mapped neural population activity into 20 dimensions which maximized the variance accounted for in four task-related conditions: direction, decoder, direction-decoder interaction, and condition-independent. Figure 6 shows the total variance (in trial-averaged neural activity) explained by each factor, averaged across sessions within each participant. The variance explained per factor was consistent between T8 and T5, with

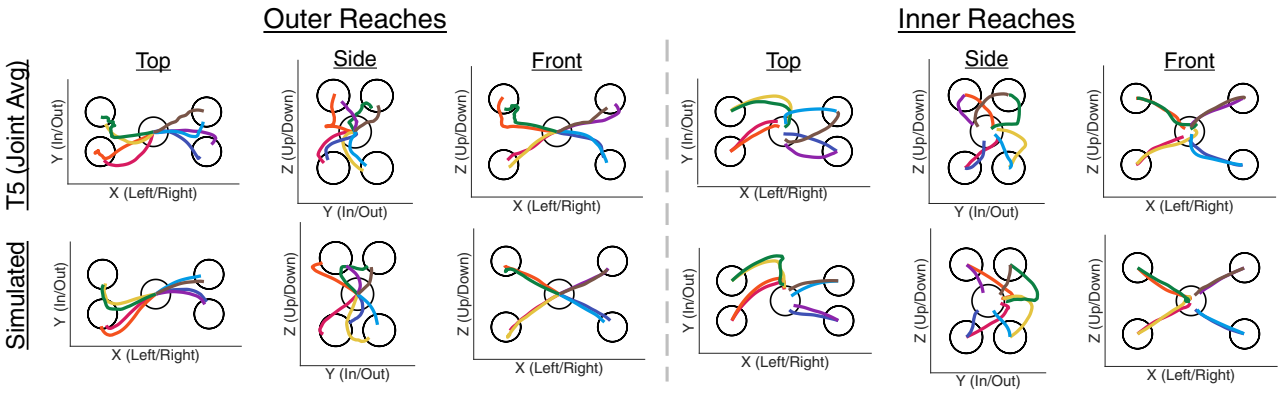

Figure 7. T5 Trajectories during joint control and simulated trajectories. 2D views of the average trajectories completed by T5 during joint velocity control (top row) and simulated trajectories using a joint decoder applied to a Cartesian tuned neural population (bottom row). Target regions are shown as black circles, trajectories are colored per target, and separated based on reaches originating at (outer) or concluding at (inner) the center target.

the direction components explaining the most neural population variance (T8: 39%, T5: 55%), the condition-independent components explaining nearly 15%, the decoder condition components explaining <2%, and the direction-decoder interaction components explaining <7%. Supplementary figure 5 plots the trial-averaged neural activity in the four largest dimensions corresponding to each factor from a sample session from each participant. The condition-independent components show typical rises in activity in accord with the task epochs, the direction components show clear separation based on target direction (color) but not decoder (line style), and the decoder components show very little modulation to the type of decoder used. The direction-decoder interaction components show population activity changes that differ in relation to both the decoder used during the block and the direction of the trial, but these changes were small in magnitude (<4% variance explained) for both participants, and were largest after the go-cue. The magnitudes and time course of the decoder and decoder-direction interaction components suggest that the users did not change strategies between decoder conditions, because different closed-loop strategies would likely result in large differences in neural activity between the blocks. These results are consistent with a hypothesis that the participants used the same strategy in all blocks, regardless of decoder condition, but that direction-specific kinematic differences between decoder conditions, such as path curvature, required feedback corrections that resulted in small neural activity differences and dPCA decoder-direction interaction components.

3.4. Simulated BCI trajectories

We observed consistent curvature in the closed-loop joint controlled trajectories from both participants, but it is difficult to interpret the paths without a model of the types of curvature that would result from different combinations of neural tuning and decoders. The neural tuning results suggested that single feature activity from both participants may be better explained by a Cartesian error model than a joint error model. Therefore, we hypothesized that the curvature seen in closed-loop trajectories may be a result of applying a joint-velocity decoder to Cartesian velocity tuned features. To test this hypothesis, we used a feedback-model of BCI control to simulate trajectories that may result from applying a joint velocity decoder to a population of Cartesian velocity tuned features. Figure 7 shows 2D projections of the average paths of 200 simulated reaches directly below the average trajectories of actual T5 joint control paths. The paths produced through simulating Cartesian tuning with a joint decoder shows many similar properties to the paths generated during actual T5 joint-control blocks, including target-dependent curvature of similar direction and magnitude. These results agree with our hypothesis and suggest that decoder mismatch, e.g. a joint velocity decoder applied to Cartesian tuned neurons, is capable of producing trajectory curvature similar to what was observed in both participants.

3.5. Swivel angle task performance

In the first of two swivel angle tasks, users controlled only the swivel angle of the arm to hit three joint targets while the endpoint of the hand was held constant. Across all trials, both users achieved a success rate of 100%. Figure 8(A) shows all swivel angle trials towards one target, with consistent trajectories reaching the target quickly. The second swivel angle task was a 4D posture matching task requiring users to simultaneously control the endpoint and swivel angle to reach sequential targets, using either a Cartesian + swivel or joint angle velocity decoder (supplementary video 1). Figure 8(B) shows the average and 95% confidence interval of the Cartesian + swivel decoder magnitude, separated between the Cartesian and swivel angle decoder components and targets. Both participants had large magnitude changes between the two target types (translation-only or swivel-only movements) when using the Cartesian + swivel angle decoder, showing they could selectively suppress endpoint movements while making swivel angle changes and suppress swivel angle changes during endpoint movements. Finally, performance in the 4D posture matching task was compared between blocks with the Cartesian + swivel and joint velocity decoders. Both participants achieved significantly more targets when using the Cartesian + swivel velocity decoder ($p < 0.001$), regardless of the required movement type (figure 8(C)).

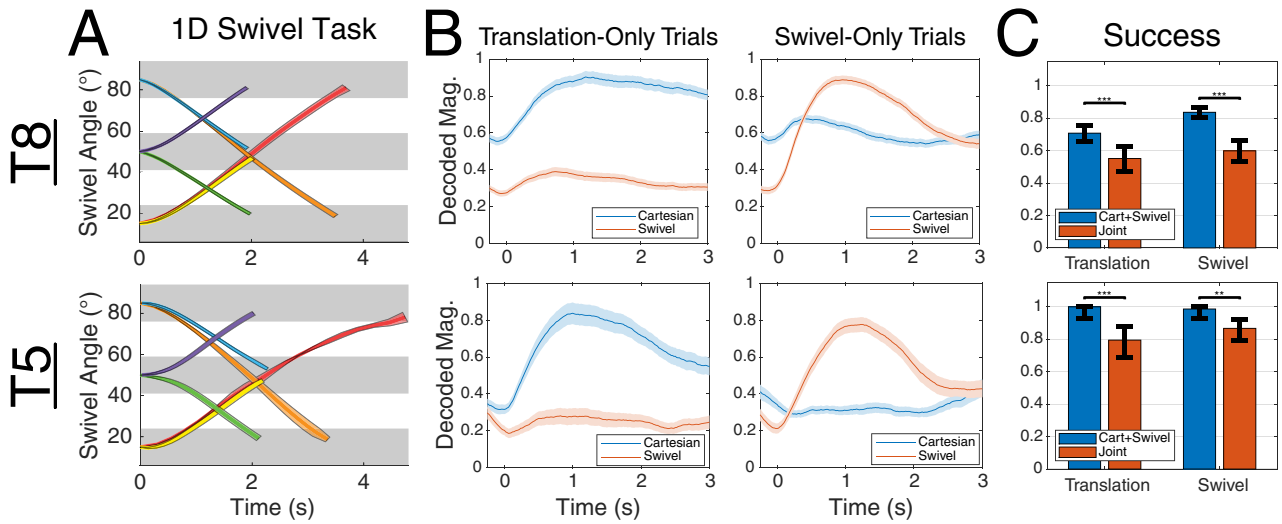

Figure 8. Swivel decoding performance. (A) 1D swivel task trajectories, average ± standard error (colored by target and starting swivel angles, allowed target regions shown in grey). (B) Average decoder magnitude during the 4D posture matching task, separated between the Cartesian (blue) and swivel (orange) components of the decoder and between translation (1st column) and swivel (2nd column) targets. (C) Success rates in the posture matching task, separated by decoder (color) and target type (column). Significance levels: ** $p < 0.01$, *** $p < 0.001$.

4. Discussion

4.1. Overview

In this study, we explored whether BCI control of arm movements would be superior using Cartesian velocity or joint velocity based control. Typical BCI's decode Cartesian commands for controlling a variety of effectors, but primate BCI literature suggests that joint information may be equally or better represented in intracortical motor cortex recordings [12–15]. Two human BCI users performed 3D and 4D virtual reaching tasks (figure 1) using Cartesian velocity and joint velocity decoders and their performance and neural tuning was compared across the decoder conditions.

4.2. Task performance

There were two primary differences found in performance metrics between the Cartesian and joint decoder conditions in the 3D Endpoint task, each of which showed superior performance using Cartesian velocity control of arm movements. First, both users acquired significantly more targets using a Cartesian velocity decoder than when using a joint velocity decoder (figure 3(A)). Additionally, joint-controlled trajectories appeared noisier and less consistent (figure 2), and this was supported by path efficiency and reach curvature metrics (figures 3(B) and (C)).

In the literature of human able-bodied movement studies, some tasks (such as reaching to the face) and workspaces have been shown to evoke curvature in natural reach trajectories [40, 41]. However, most studies have reported that targeted movements in a constrained workspace, similar to the virtual reality environment used in this work, are largely straight [42–44]. Additionally, it has been hypothesized that curvature in reaches could be a result of visual distortions [45], miscalculation of hand position [46], gravity [41], (factors that were likely reduced in our tasks), and that reaching movements are likely planned to be straight [42, 47]. Furthermore, when instructed to make straight movements, participants produce much straighter paths than may be expected naturally [48]. Therefore, it was expected that the participants of the current study, who were instructed to attempt to make straight movements, would be able to produce such movements towards the targets.

Furthermore, path efficiency and curvature metrics were also calculated in 4D 'joint space', comparing actual joint paths to hypothetical linear joint changes. Both 'joint space' metrics were worse during joint-control blocks, confirming that the consistent curvature in Cartesian space was not a result of users performing reaches that were straight in joint space (supplementary figure 2).

Finally, controlling a fourth kinematic dimension (e.g. decoding 4D joint velocity) does not necessarily decrease performance in the 3D Endpoint task. Success in the 3D Endpoint task is only measured in Cartesian space, so there exists a large manifold of joint angles which satisfy the target requirements. Thus, the dimensionality of the task remains 3D for both the joint and Cartesian decoder conditions.

4.3. Neural tuning

dPCA analyses revealed only small changes (<8% variance explained) in population activity between blocks performed under Cartesian and joint velocity control for both participants (figure 6). This result is congruent with participant responses during interviews, where both users described using the same strategy to control the arm regardless of decoder. The presence of direction-decoder interaction components could represent target-direction dependent changes between the two decoder conditions in either user strategy, neural encoding, or simply typical feedback corrections to varying kinematics

between the two decoder conditions (curvature, path length, etc). Both participants had small direction-decoder interaction components (<7% variance explained), which rose in magnitude during the reach period (supplementary figure 5). These small population activity differences are consistent with neural activity representing closed-loop feedback corrections to the consistently differing kinematics between the decoder blocks (figure 2).

Single feature encoding model performance in both participants showed that a 3D Cartesian position error vector better described neural activity than a 4D joint position error vector (figure 5), despite using one less explanatory variable. We also constructed a 4D coordinated joint error vector, which represented the joint velocities which produce movements that are straight to the target in Cartesian space. Results in supplementary figure 4 show that the Cartesian model similarly outperformed this alternative joint encoding model. Additionally, the single feature encoding results showed only slight differences in performance between blocks that were performed under Cartesian or joint velocity control. This result is consistent with the dPCA results, and further supports that neural encoding of intended arm movements did not change during closed-loop control with the different decoder conditions.

To test the hypothesis that the decrease in performance during joint control could be due to a decoder mismatch with the neural commands (users commanding a 3D Cartesian position error command, but the decoder is extracting a 4D joint error command), a BCI-user feedback model was used to simulate closed-loop trajectories with these assumptions (figure 7). The simulated trajectories were qualitatively similar in curvature and path for most target directions, supporting a hypothesis that the user generated a 3D Cartesian error vector, and trajectories during joint control blocks represent an imperfect kinematic transformation of the user command and expected feedback corrections.

Many previous studies give evidence that strong joint tuning is present in motor cortical areas [18, 49, 50], but our results show stronger tuning to a Cartesian position error representation of intended arm movements. However, there are a few key differences in this work that may account for lower joint tuning performance. First, both participants had extensive experience, through other studies, with a variety of virtual tasks which primarily utilized Cartesian velocity decoding for cursor control. This previous long-term (>1 year) learning may have biased their neural responses towards Cartesian movement representations. Second, there were no kinetics or musculoskeletal dynamics in the virtual arm movements, which may further influence neural representations towards Cartesian coordinate frames [48, 51]. The inherent noise in the BCI-commanded virtual effector also makes these tasks highly feedback controlled [37], while natural reaching (and thus neural planning of able-bodied movements) may rely more on feedforward planning and execution [48]. Finally, this study recorded from the cortex of participants with chronic spinal cord injuries, who may have different cortical representations of movement compared to primates performing actual reach and grasp movements [24]. In addition to cortical reorganization, it's possible that the lack of afferents from the limb (carrying proprioceptive and other sensory feedback), changes the neural representations of movement. However, the goal of this study was not to differentiate between the relative strength of neural tuning for joint velocity and Cartesian coordinates after chronic SCI. Instead, these experiments attempted to answer the question of whether a BCI user could learn to control functionally relevant arm movements using two sets of commands, and therefore limited the task to workspaces that paralyzed BCI users would likely have access to via an assistive device.

4.4. Focused joint control practice

Both participants had extensive previous experience controlling virtual movements using Cartesian velocity commands. Therefore, participant T8 performed three sessions of only joint velocity control to investigate if practice or short-term learning would contribute to changes in performance. Within and between the three sessions, there were no significant changes in success rate, path efficiency, or decoding SNR. Thus, the diminished performance during joint control blocks was not due to short-term learning deficits or a misunderstanding of the task. However, recent BCI literature suggests that a longer duration and more specialized training paradigm may enable participants to improve performance using a joint velocity decoder [52]. Primarily, a longer learning study might enforce straight trajectories, give better visual feedback, or slowly titrate the task from an easier, learned task to the final desired mapping from neural activity to effector kinematics [53–55].

4.5. Postural control through swivel angle commands

To investigate BCI control of whole arm endpoint and posture, we compared joint velocity decoders to novel Cartesian + swivel velocity decoders, which implemented the first direct BCI control of swivel angle. Single-dimension swivel angle movements were accurate and consistent (100% trial success in both participants, figure 8(A)). Multi-dimensional control, via a combined Cartesian + swivel angle velocity decoder, enabled independent modulation of the arm endpoint and swivel angle (figure 8(B)). In a 4D posture matching task, the Cartesian + swivel decoder performed significantly better than the joint decoder at achieving both the endpoint and the swivel targets (figure 8(C)). Thus, the decreased performance with the joint decoder cannot be attributed to the dimensionality constraint of the endpoint-only task (decoding four joint dimensions for a purely 3D task), as the joint velocity decoder similarly underperformed in the 4D task.

Participant T5's performance in the translation-only portion of the posture matching task (figure 8(C)) increased compared to the 3D endpoint only task (figure 3(A)), while participant T8's performance decreased. The blocks for these two tasks were completed on separate days, and therefore the cross-task performance discrepancy may reflect normal BCI day-to-day variability in signal quality, or perhaps a change in concentration or strategy due to the higher complexity of the

posture matching task. However, cross-task performance was not explicitly evaluated, and within-task comparisons between Cartesian and joint decoder conditions were completed within the same days.

Notably, our posture matching task design separated trials into targets that required translation-only or swivel angle-only changes (supplementary video 1). This task structure allowed us to assess independent performance of the two postural components (figure 8), even though the participants were in closed-loop control of all dimensions at all times. Therefore, while simultaneous control of translation and swivel angle has been demonstrated possible, further experimental tests are needed to assess performance during simultaneous translation and swivel angle postural targets. Successfully decoding a desired swivel angle opens new avenues for BCI-controlled assistive devices, potentially giving users more control over their prostheses and improving certain tasks, such as obstacle avoidance.

4.6. Implications for BCI control of arm reaching

These results show that BCI-controlled virtual arm movements are more accurate and straight when using Cartesian velocity commands compared to joint velocity commands. However, success rates were still moderate (>60%) with the lower performing joint decoder. Therefore, it is not surprising that BCI control over the joint velocities of an assistive device could restore functionally relevant movements over longer trial times [10]. Additionally, there may be systems in which BCI joint-controlled arm movements perform effectively, such as when the effector cannot perform straight movements (e.g. an FES system limited by range of motion or muscle fatigue) or when performing certain tasks that typically utilize curved movements (e.g. self-feeding and other head-directed movements [40]). However, the results of this study suggest that utilizing a controller to map decoded Cartesian commands to the required effector inputs will likely provide higher accuracy and path efficiency as compared to directly controlling joint velocities. Despite broad evidence of joint representations in the motor cortex during natural reaching, the participants in this study had difficulty controlling virtual reaching movements of a closed-loop BCI using joint velocity. Other proposed variables for BCI control of reaching, such as muscle activations and joint torques [14, 56–58], can require even more complex coordination patterns than joint velocities. Results from this study suggest that BCI arm control using these lower-level movement representations might be more difficult than decoding Cartesian velocities from people with paralysis.

5. Conclusion

This study explored whether BCI control of arm movements would be superior using Cartesian velocity or joint velocity based control. Two BrainGate2 participants with chronic SCI used a BCI to perform 3D and 4D virtual reality arm movements using both command interfaces. In all tasks, Cartesian control outperformed joint velocity control in success rate, path efficiency, and curvature. Neural tuning was explained better by a Cartesian encoding model, and population level analyses showed little difference in activity between the two decoder conditions. Finally, this work demonstrates the first human BCI control of swivel angle, achieving high performance and selective modulation in two participants.

Acknowledgments

The authors would like to extend an immense thank you to the participants and their families for their time and extensive contributions to our studies. Support for this study was provided primarily by the Eunice Kennedy Shriver National Institute of Child Health and Human Development (NICHD) of the National Institutes of Health (NIH) under award number R01HD077220, and by NICHD-NIH N01HD53403, the National Institute on Deafness and Other Communication Disorders of NIH under award number R01DC009899, and by T32 funding from the National Institute of Biomedical Imaging and Bioengineering of the NIH (5T32EB004314-15). Support was also provided by the United States Department of Veterans Affairs, Rehabilitation Research and Development Service, under award numbers A2295R, B4853C, B6453R, and N9228C, and by the MGH-Deane Institute, The Executive Committee on Research (ECOR) of Massachusetts General Hospital.

The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health, or the Department of Veterans Affairs or the United States Government. CAUTION: Investigational Device. Limited by Federal Law to Investigational Use.

Author contributions

DY contributed in study conception, designed the experiments, collected data with participants T5 and T8, performed data analyses, and drafted the manuscript, which was further edited by all authors. FRW assisted in design of experiments, contributed to data analyses and results interpretation, and collected data with participant T5. WDM and BM performed data collection with participant T8. PR also performed data collection with participant T5. BW provided participant T8 with post-surgical care. JS and JM planned and executed neurosurgical placement of the electrode arrays for participant T8. LRH was the investigational device exemption sponsor-investigator of the pilot clinical trial. ABA, KS, and RFK assisted in study conception and design, and all authors assisted in interpretation of results and manuscript writing.

ORCID iDs

D Young https://orcid.org/0000-0003-1861-971X
F Willett https://orcid.org/0000-0002-2652-8511
W D Memberg https://orcid.org/0000-0002-2545-8574
B Murphy https://orcid.org/0000-0001-7784-3434

K V Shenoy https://orcid.org/0000-0003-1534-9240
L R Hochberg https://orcid.org/0000-0003-0261-2273

References

[1] Hochberg L R, Serruya M D, Friehs G M, Mukand J A, Saleh M, Caplan A H, Branner A, Chen D, Penn R D and Donoghue J P 2006 Neuronal ensemble control of prosthetic devices by a human with tetraplegia *Nature* **442** 164–71
[2] Jarosiewicz B *et al* 2015 Virtual typing by people with tetraplegia using a self-calibrating intracortical brain–computer interface *Sci. Transl. Med.* **7** 313ra179
[3] Pandarinath C, Nuyujukian P, Blabe C H, Sorice B L, Saab J, Willett F R, Hochberg L R, Shenoy K V and Henderson J M 2017 High performance communication by people with paralysis using an intracortical brain–computer interface *Elife* **6** 1–27
[4] Gilja V *et al* 2015 Clinical translation of a high-performance neural prosthesis *Nat. Med.* **21** 1142–5
[5] Hochberg L R *et al* 2012 Reach and grasp by people with tetraplegia using a neurally controlled robotic arm *Nature* **485** 372–5
[6] Collinger J L, Wodlinger B, Downey J E, Wang W, Tyler-Kabara E C, Weber D J, McMorland A J, Velliste M, Boninger M L and Schwartz A B 2013 High-performance neuroprosthetic control by an individual with tetraplegia *Lancet* **381** 557–64
[7] Wodlinger B, Downey J E, Tyler-Kabara E C, Schwartz A B, Boninger M L and Collinger J L 2015 Ten-dimensional anthropomorphic arm control in a human brain−machine interface: difficulties, solutions, and limitations *J. Neural Eng.* **12** 016011
[8] Aflalo T *et al* 2015 Decoding motor imagery from the posterior parietal cortex of a tetraplegic human *Science* **348** 906–10
[9] Bouton C E *et al* 2016 Restoring cortical control of functional movement in a human with quadriplegia *Nature* **533** 247–50
[10] Ajiboye A B *et al* 2017 Restoration of reaching and grasping in a person with tetraplegia through brain-controlled muscle stimulation: a proof-of-concept demonstration *Lancet* **389** 1821–30
[11] Childress D S 1992 Upper-limb prosthetics: control of limb prostheses *Atlas of Limb Prosthetics: Surgical, Prosthetic, and Rehabilitation Principles* (Maryland Heights, MO: Mosby, Inc.)
[12] Scott S H and Kalaska J F 1995 Changes in motor cortex activity during reaching movements with similar hand paths but different arm postures *J. Neurophysiol.* **73** 2563–7
[13] Ajemian R, Bullock D, Grossberg S and Kine S G 2000 Kinematic coordinates in which motor cortical cells encode movement direction *J. Neurophysiol.* **84** 2191–203
[14] Oby E R, Ethier C and Miller L E 2013 Movement representation in the primary motor cortex and its contribution to generalizable EMG predictions *J. Neurophysiol.* **109** 666–78
[15] Kakei S, Hoffman D S and Strick P L 1999 Muscle and movement representations in the primary motor cortex *Science* **285** 2136–9
[16] Memberg W D, Polasek K H, Hart R L, Bryden A M, Otr L, Kilgore K L, Nemunaitis G A, Hoyen H A, Keith M W and Kirsch R F 2014 Implanted neuroprosthesis for restoring arm and hand function in people with high level tetraplegia *Arch. Phys. Med. Rehabil.* **95** 1201–1211.e1
[17] Aflalo T N and Graziano M S A 2007 Relationship between unconstrained arm movements and single-neuron firing in the macaque motor cortex *J. Neurosci.* **27** 2760–80
[18] Vargas-Irwin C E, Shakhnarovich G, Yadollahpour P, Mislow J M K, Black M J and Donoghue J P 2010 Decoding complete reach and grasp actions from local primary motor cortex populations *J. Neurosci.* **30** 9659–69
[19] Reina G A, Moran D W and Schwartz A B 2001 On the relationship between joint angular velocity and motor cortical discharge during reaching *J. Neurophysiol.* **85** 2576–89
[20] Soechting J F 1989 Elements of coordinated arm movements in three-dimensional space *Adv. Psychol.* **61** 47–83
[21] Wu W and Hatsopoulos N 2006 Evidence against a single coordinate system representation in the motor cortex *Exp. Brain Res.* **175** 197–210
[22] Wu W and Hatsopoulos N G 2007 Coordinate system representations of movement direction in the premotor cortex *Exp. Brain Res.* **176** 652–7
[23] Scott S H and Kalaska J F 1997 Reaching movements with similar hand paths but different arm orientations. I. Activity of individual cells in motor cortex *J. Neurophysiol.* **77** 826–52
[24] Hotz-Boendermaker S, Funk M, Summers P, Brugger P, Hepp-Reymond M-C, Curt A and Kollias S S 2008 Preservation of motor programs in paraplegics as demonstrated by attempted and imagined foot movements *NeuroImage* **39** 383–94
[25] Ajiboye A, Simeral J, Donoghue J, Hochberg L and Kirsch R 2012 Prediction of imagined single-joint movements in a person with high level tetraplegia *Trans. Biomed. Eng.* **59** 2755–65
[26] Velliste M, Perel S, Spalding M C, Whitford A S and Schwartz A B 2008 Cortical control of a prosthetic arm for self-feeding *Nature* **453** 1098–101
[27] Kang T, He J and Tillery S I H 2005 Determining natural arm configuration along a reaching trajectory *Exp. Brain Res.* **167** 352–61
[28] Tolani D and Badler N I 1996 Real-time inverse kinematics of the human arm *Presence Teleoperators Virtual Environ.* **5** 393–401
[29] Brandman D M *et al* 2017 Rapid calibration of an intracortical brain–computer interface for people with tetraplegia *J. Neural Eng.* **15** 026007
[30] Tolani D, Goswami A and Badler N I 2000 Real-time inverse kinematics techniques for anthropomorphic limbs *Graph. Models* **62** 353–88
[31] Jarosiewicz B, Masse N Y, Bacher D, Cash S S, Eskandar E, Friehs G, Donoghue J P and Hochberg L R 2013 Advantages of closed-loop calibration in intracortical brain–computer interfaces for people with tetraplegia *J. Neural Eng.* **10** 046012
[32] Eberle H, Nasuto S J and Hayashi Y 2017 Integration of visual and joint information to enable linear reaching motions *Sci. Rep.* **7** 40869
[33] Oyama T, Uno Y and Hosoe S 2007 Analysis of variability of human reaching movements based on the similarity preservation of arm trajectories *Lecture Notes in Computer Science* (Berlin: Springer) pp 923–32
[34] Nõmm S, Toomela A and Gaichenja I 2017 Towards the notion of average trajectory of the repeating motion of human limbs *Lecture Notes in Computer Science* (Cham: Springer) pp 582–91
[35] Kobak D, Brendel W, Constantinidis C, Feierstein C E, Kepecs A, Mainen Z F, Qi X-L, Romo R, Uchida N and Machens C K 2016 Demixed principal component analysis of neural population data *Elife* **5** e10989
[36] Kaufman M T, Seely J S, Sussillo D, Ryu S I, Shenoy K V and Churchland M M 2016 The largest response component in the motor cortex reflects movement timing but not movement type *eNeuro* **3** 0085–16

[37] Willett F R *et al* 2017 Signal-independent noise in intracortical brain–computer interfaces causes movement time properties inconsistent with Fitts' law *J. Neural Eng.* **14** 026010
[38] Young D, Willett F, Memberg W D, Murphy B, Walter B, Sweet J, Miller J, Hochberg L R, Kirsch R F and Ajiboye A B 2018 Signal processing methods for reducing artifacts in microelectrode brain recordings caused by functional electrical stimulation *J. Neural Eng.* **15** 026014
[39] Willett F R *et al* 2017 Feedback control policies employed by people using intracortical brain–computer interfaces *J. Neural Eng.* **14** 016001
[40] Petreska B, Billard A, Petreska B and Billard A 2009 Movement curvature planning through force field internal models *Biol. Cybern.* **100** 331–50
[41] Atkeson C G and Hollerbach J M 1985 Kinematic features of unrestrained vertical arm movements *J. Neurosci.* **5** 2318–30
[42] Morasso P 1981 Spatial control of arm movements *Exp. Brain Res.* **42** 223–7
[43] Flash T and Hogan N 1985 The coordination of arm movements: an experimentally confirmed mathematical model *J. Neurosci.* **5** 1688–703
[44] Hollerbach J M and Flash T 1982 Dynamic interactions between limb segments during planar arm movement *Biol. Cybern.* **44** 67–77
[45] Wolpert D M, Ghahramani Z and Jordan M I 1994 Perceptual distortion contributes to the curvature of human reaching movements *Exp. Brain Res.* **98** 153–6
[46] Goodbody S J and Wolpert D M 1999 The effect of visuomotor displacements on arm movement paths *Exp. Brain Res.* **127** 213–23
[47] Flanagan J R and Rao A K 1995 Trajectory adaptation to a nonlinear visuomotor transformation: evidence of motion planning in visually perceived space *J. Neurophysiol.* **74** 2174–8
[48] Osu R, Uno Y, Koike Y and Kawato M 1997 Possible explanations for trajectory curvature in multijoint arm movements *J. Exp. Psychol. Hum. Percept. Perform.* **23** 890–913
[49] Saleh M, Takahashi K and Hatsopoulos N G 2012 Encoding of coordinated reach and grasp trajectories in primary motor cortex *J. Neurosci.* **32** 1220–32
[50] Chao Z C, Nagasaka Y and Fujii N 2010 Long-term asynchronous decoding of arm motion using electrocorticographic signals in monkeys *Frontiers Neuroeng.* **3** 3
[51] Nakano E, Imamizu H, Osu R, Uno Y, Gomi H, Yoshioka T and Kawato M 1999 Quantitative examinations of internal representations for arm trajectory planning: minimum commanded torque change model *J. Neurophysiol.* **81** 2140–55
[52] Golub M D, Sadtler P T, Oby E R, Quick K M, Ryu S I, Tyler-Kabara E C, Batista A P, Chase S M and Yu B M 2018 Learning by neural reassociation *Nat. Neurosci.* **21** 607
[53] Krakauer J W and Mazzoni P 2011 Human sensorimotor learning: adaptation, skill, and beyond *Curr. Opin. Neurobiol.* **21** 636–44
[54] Jarosiewicz B, Chase S M, Fraser G W, Velliste M, Kass R E and Schwartz A B 2008 Functional network reorganization during learning in a brain–computer interface paradigm *Proc. Natl Acad. Sci. USA* **105** 19486–91
[55] Sadtler P T, Quick K M, Golub M D, Chase S M, Ryu S I, Tyler-Kabara E C, Yu B M and Batista A P 2014 Neural constraints on learning *Nature* **512** 423–6
[56] Chhatbar P Y and Francis J T 2013 Towards a naturalistic brain–machine interface: hybrid torque and position control allows generalization to novel dynamics *PLoS One* **8** e52286
[57] Suminski A J, Willett F R, Fagg A H, Bodenhamer M and Hatsopoulos N G 2011 Continuous decoding of intended movements with a hybrid kinetic and kinematic brain machine interface *2011 Annual Int. Conf. IEEE Engineering in Medicine and Biology Society* (IEEE) pp 5802–6
[58] Ethier C, Oby E R, Bauman M J and Miller L E 2012 Restoration of grasp following paralysis through brain-controlled stimulation of muscles *Nature* **485** 368–71