Accurate neuroprosthetic control through latent state transition training

Andres Agudelo-Toro[1], Jonathan A. Michaels[1,2], Wei-An Sheng[1,3], and Hansjörg Scherberger[1,4,*]

[1]Neurobiology Laboratory, Deutsches Primatenzentrum GmbH, 37077 Göttingen, Germany; [2]Brain and Mind Institute, Western University, London, ON, Canada; [3]Institut des Sciences Cognitives Marc Jeannerod, Lyon, France; [4]Faculty of Biology and Psychology, University of Göttingen, 37073 Göttingen, Germany; *Corresponding author: hscherberger@dpz.eu

Abstract

Hand movements are an essential way that primates interact with the environment and they comprise some of our most complex behaviors. Despite remarkable recent progress in intracortical brain-computer interfaces, current neuroprostheses still lack the fine control required to interact with objects of daily living. We present a BCI training approach to develop neural activity latent variables for accurate hand shape control. The method shapes neural activity by daily training to target patterns and provides incremental control of degrees of freedom. When tested in two rhesus monkeys implanted in key areas of the cortical grasping circuit, our approach provides hand configuration accuracy comparable to native grasping. This training method was advantageous in a collision task, can potentially extend to control in higher-dimensions, and facilitates learning of novel actions in the kinematic latent space. Using our paradigm, neural activity patterns evolved to match the trained trajectories over time, making our approach a potential tool to study cortical learning.

Keywords

Grasping, neuroprostheses, brain computer interface, BCI, intention-estimation, decoder, primate

Introduction

Intra-cortical brain-computer interfaces (BCI) enable the direct control of an external device through brain activity. By targeting brain regions such as motor cortex, BCIs can connect individuals to assistive devices and provide a window into evolving brain function[1–3]. These capabilities have been successfully demonstrated in monkey experiments[4–9] and in human trials[10–15].

Hand function recovery is among the top priorities for individuals with spinal cord injury[16]. By establishing a direct, high-bandwidth connection to cortex, BCIs have the potential of driving a large number of degrees of freedom in a fully articulated hand prosthesis. With tens of joints and muscles, our hands are highly flexible, and can achieve a large number of configurations required for prehensile and non-prehensile interaction with objects[17,18]. Although significant progress has been made toward full hand BCI prostheses[15,19,20], and precise robotics and recording techniques are increasingly available[21–23], human BCI devices are still limited to basic grasping and cannot yet reproduce the rich set of configurations of the native-hand.

Given the complexity of the hand, BCI control of a fully articulated hand will likely require complementing training protocols. Humans can rapidly switch among hand configurations intuitively, i.e., without thinking

about individual finger or wrist joint rotations. This capacity evolves through lifelong learning that starts with grasping[24], but is refined through activities such as writing, typing or learning an instrument. A training protocol for hand shape selection would endow BCI users the capacity to traverse the continuous hand configuration space, and complement existing training approaches such as those for object manipulation[19,15] and individual finger control[20].

Towards this goal, we present a training protocol for the BCI control of hand shape. Our approach develops a neural latent space for the control of hand states through training of target latent transitions. Neural control of a complex multidimensional effector such as the hand can be generalized by a decoder transformation from a space of neural state variables to a space of kinematic state variables (Fig. 1). Hand configurations, such as grip types, can be seen as specific states in this space (Fig. 1A,B). Correspondingly, neural activity has to evolve over training days to cover this latent space (Fig. 1A,B). Our approach takes into account trajectories in the kinematic latent space that are important to achieve final postural configurations. To train latent space transitions, we replaced the executed kinematics in the decoder re-training stage with the attempted trajectories, building upon intention-estimation training strategies[25–28]. To develop the individually required degrees of freedom, we trained incrementally based on knowledge of the evolution of neural latent spaces[29–33]. We demonstrate our approach in a closed loop BCI (Fig. 1C,D), where two monkeys implanted with recording electrodes in key areas of the grasping circuit were trained to control a hand prosthesis subspace obtained from grasping kinematic data.

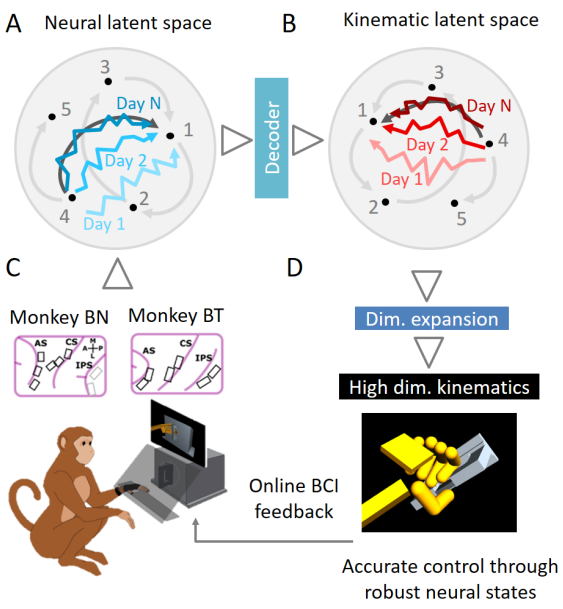

Figure 1. Overview of BCI technique and terminology. (A) The neural latent space is a lower dimensional subspace of the neural state space spanned by the activity of the observed neurons. (B) Neural activity can be decoded to drive a kinematic latent space, which is in turn, a subspace of the kinematic state space (e.g. joint angles). (A,B) Fixed effector postures such as grip types can be seen as fixed points in both spaces (points 1 to 5). Posture change intentions can be seen as transitions between these (gray arrows). At the start of training (Day 1), transition attempts are inaccurate and require shaping (light blue and red arrows). Over training, transition attempts are refined, better reflecting the subject's intention (Day 2 to N, dark blue and red arrows). (C) Basic setup. A subject sits in front of a screen displaying a virtual hand and rotating handle. The virtual hand is controlled by the BCI. Neural data is extracted from floating microelectrodes on the anterior intraparietal sulcus (AIP), hand area of M1, and area F5 of the ventral premotor cortex. (D) End effector values are reconstructed by mapping latent variables to the original kinematic state space through a dimensionality expansion step.

Our approach produces a set of robust state transitions in the kinematic space for hand state control. Accuracy of the BCI was comparable to native hand precision and its performance improved over traditional algorithms in a number of metrics, including success rate and variability of the grasps. When compared to a classic estimated-intention method in an environment with obstacles, our strategy achieved better task performance. Additional tests employing a model of the grasping circuit[34] showed that our approach extends well to high-dimensional state transitions with a large number of hand configurations. We observed that the training strategy eased learning of intermediate states in the hand configuration latent space. Our online BCI learning paradigm[1–3] created neural patterns related to the control of the kinematic latent variables. The neural population activity in motor cortex produced velocity related and rich position-like patterns over training sessions, confirming previous knowledge on the control of hand configuration[35–41], but in contrast to reaching

data[42,39,43]. Our combined training procedure enables the control of a hand effector that accurately reflects the grip intention of the subject and represents a potential tool to study the evolution of neural dynamics.

Results

Overview of the training protocol

We refer to our approach as Latent State Transition Training (LSTT). LSTT is composed of four core components: *construction of a kinematic latent space*, *incremental training of the kinematic latent variables*, daily *decoder fitting to attempted trajectories*, and *fine-tuning with a non-linear layer* after training of each latent variable (Fig. 2). We present LSTT in the context of our grasping task. In the task (Fig. 2A), two rhesus monkeys (BN and BT) were presented with a virtual rotating handle in five possible orientations, and asked to grasp the handle using one of the base grips of the GRASP hierarchy: power or precision[18]. Animals grasped the handle using a virtual arm avatar following a preparatory cue. Depending on the task setting, the arm avatar was driven by the native arm or the BCI (Fig. 2B, Methods). During the native arm setting, animals received visual feedback through the virtual environment while using their native arm to grasp a concealed real handle. During the BCI setting, animals controlled arm avatar only using their thoughts. The native arm was concealed in both settings. The native task setting was used to collect kinematic data and prepare the animals for the BCI task. To collect kinematic data, we used a high accuracy hand and arm tracker (1mm joint position error, Methods).

Tracked native hand kinematics are used to *construct a kinematic latent space* (Fig. 2C,D). We define the kinematic latent space as a subspace of the full-dimensional kinematic state space where a few latent variables describe most of the variance required for hand state switching. In our case, we used kinematic data collection while the monkeys grasped the handle at the different angles and grip types to describe the full-dimensional kinematic state space. Kinematic activity formed a 28 dimensional state space of joint angles (25 of the fingers and 3 of the wrist) used to build the kinematic latent subspace. Basic dimensionality reduction of the full space (principal components) showed that three components describe ca. 90% of the variance for the two grip types and five angles. Two example hand states in this space are the open hand (no grip) and the closed hand (power) states. Transitions within these switch between grip types and are non-linear (curved), as a result of phase differences between individual joint angle time profiles (Fig. S1). Although the three main principal components could be used as target latent variables, we chose not to use them as they were unintuitive and lacked the necessary precision (e.g. to correctly achieve the precision grip). Instead, a custom dimensionality reduction/expansion procedure (Methods) was used to *construct* the three main latent variables: grip aperture, wrist rotation, and grip type. This transform provided sufficient distance between conditions and preserved the variance required for the precision grip (Fig. 6 and Methods). Trajectories in the latent space were designed to increase separation between conditions and provided a set of target grip attempts used to train the BCI.

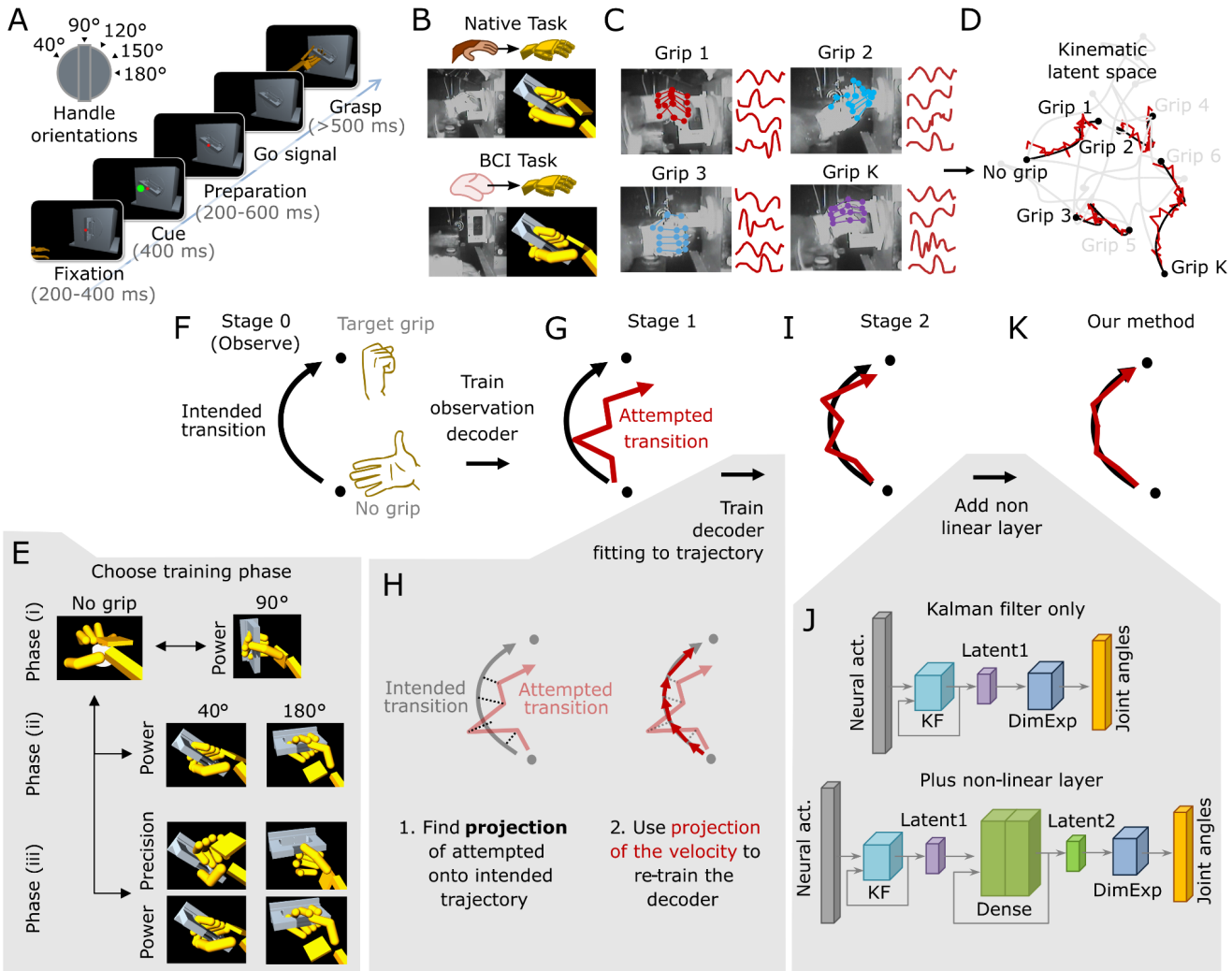

Figure 2. Description of the task and training strategy. (A) Handle orientations and task flow. The task started with a fixation dot, followed by grip type and handle orientation cues on the display. A "Go" signal was indicated by the absence of the fixation dot. The animal should then grasp the handle with its hand or BCI and hold it for at least 500ms. Monkeys were then rewarded and returned their hand to the initial resting position and initiated the next trial. Multidimensional tracking is used to determine a set of grip types by capturing kinematics of the hand joints. (B) The two basic task configurations used in the experiments. In the native task, the subject controls a virtual hand in a physics environment using its own hand. In the BCI task, the virtual hand is controlled only through brain activity while the subject remains still. (C) Kinematic recordings are used to build the grip latent space of grip configurations. Each grip type can be described by the trajectories of M joint angle kinematics over time (example red traces, in our case M=28). (D) Joint angle trial averages can be used to build an M-dimensional state space of K grip postures and transitions between the postures. This latent space tends to exist in a lower dimensional kinematic latent space of dimension P<<M (in our case P=3). (E) Training phases. Pictures illustrate the resting (no grip) posture and the target grips (power, precision) at different angles. Angles correspond to those of (A). Arrows indicate possible transitions. In Phase (iii) 4 transitions are possible. (F-I) Daily training. (F) At Stage 0, the subject observes fully automated grips while neural activity is captured. Neural activity and the idealized intended transitions are used to train an initial observation decoder. (G) The observation decoder is used to execute a first batch of grip attempts. Performance of the decoder during this stage is typically low. (H) Apply fit to trajectory. The attempted transitions of Stage 1 are projected onto the idealized (intended) trajectories to obtain the training velocity vectors. During grip hold periods, zero velocity and the idealized grip configuration is assumed. Neural and projected data are used to train a new decoder. (I) The Kalman filter (KF) decoder trained with Stage 1 data increases performance of the transitions but it is usually limited in accuracy. (J) A non-linear layer can be added to the decoder to increase accuracy. (K) The final LSTT decoder increases accuracy during transitions and hold periods.

Once a kinematic latent space is constructed, subjects are trained to *incrementally control the kinematic latent variables*. To reduce cognitive load, training is performed progressively through each axis of the latent space. We exemplify this step with our task. After subjects were implanted in three key areas of the grasping circuit: anterior intraparietal area (AIP), ventral premotor area F5, and hand M1 using floating microelectrode arrays (Fig. 1C), we proceeded to train the subjects to control the BCI. We trained the subjects in three phases: (i) grip aperture control, (ii) wrist rotation control (hand pronation/supination), and (iii) grip type selection (Fig. 2E, Fig. 6). The three phases were trained sequentially, each over several training days, not moving into the next phase until a performance criteria was met. To train each phase, we followed a BCI assisting procedure in which a certain percent of task control was done by the computer and the rest by the subject. The level of subject control incremented until achieving full brain control (BC, level 0% to 100%). The criteria to move to the next phase was achieving BC=100% and task performance ≥ 80% over several days.

To train the BCI daily, we used a three stage procedure (Fig. 2F-I, Methods)[25–27]. In the observation stage (Stage 0), the subject performs the delayed grasping task while observing automated, idealized grip attempts obtained from the kinematic recordings (performed fully by the computer i.e. BC=0%). We call these the *intended transitions*, as according to data, these are the ideal transitions to switch between grip types. Neural activity and the intended transitions during Stage 0 was used to train a decoder to be used in the initial control stage (Stage 1). In Stage 1, subjects attempted the grips using their own neural activity. We call these the *attempted transitions*. Due to misalignments between the intended and effective neural activity, control performance in Stage 1 is typically limited. Decoder re-fitting increases accuracy and increases performance of the task in a new stage (Stage 2)[25].

Neural activity during Stage 1 is used to train a new decoder using a *fit to attempted trajectory* approach. Fitting to trajectory is used to more optimally align neural activity to the kinematic latent space by preserving the non-linearities of hand state transitions. The trajectory fitting procedure is divided in two basic steps (Fig. 2H). In step 1, on the kinematic latent space, each grip transition attempt from Stage 1 is projected to its corresponding intended transition. In step 2, the projected vectors are used to build a new trajectory for each attempt. Neural data from Stage 1 and the projected trajectories are used to train the Stage 2 decoder.

The projection of the attempted onto the intended transitions is found as follows: for each state change attempt: (a) Divide each attempted and intended trajectory into decoder time-steps of equal time (N^{att} and N^{int} segments). (b) Determine the segment length of the attempted onto intended trajectory as:

$$L^{aoi} = \frac{N^{int}}{N^{att}}.$$

(c) Define the intended trajectory as an n-dimensional parametric curve $f^{int}(t)$ with timesteps $t = \{0, 1,..., N^{int}\}$. (d) The projection of attempted onto intended trajectory is found by finding the values on $f^{int}(t)$ corresponding to parameter $t = \{0, L^{aoi}, 2 \times L^{aoi},..., N^{att} \times L^{aoi}\}$. During hold periods, i.e. when the intended grip is achieved, it is assumed that the intended velocity is zero.

After training of the latent variables, we *fine-tuned the decoder by addition of a non-linear layer*. Training of the latent variables was performed with a Kalman filter (KF), but at the end of each phase, a hybrid KF/non-linear decoder was used for testing (Fig. 2J,K). This succession was intentional, as it is a way to develop robust latent variables (as for example, in variational autoencoders[44,45]). The KF is a useful BCI decoder[25,10,14,28,20] but it is prone to noise and inaccuracy[46,25,47–49]. Non-linear maps such as recurrent neural networks (RNN) are known to improve BCI decoding accuracy[46,49], as they might find idiosyncratic correlations between neural activity and kinematics or kinetics[50–52,49]. In our case, we used the noisiness of the Kalman filter to the subject's

advantage: training to performance >80% with the KF ensured the development of robust neural patterns for each grip type, thanks to the variability of the KF decoder. After this, a two-layer non-linear autoregressive with exogenous inputs (NARX) recurrent network[53] was added to the decoder. Empirically, NARX led to the best results in comparison to existing RNN training methods[46,54].

To measure the benefits of our training approach, we tested the following hypotheses: (a) the combined strategy produces a decoder close in quality to native grasp movements (Fig. 3); (b) when tested in an environment with collisions, LSTT fit-to-trajectory is advantageous over previous intention-estimation approaches (Fig. 4); (c) LSTT can potentially extend to a large number of grip types and generalize to non-trained configurations in high-dimensional latent spaces (Fig. 5); (d) our training procedure facilitates learning of intermediate configurations (Fig. 6); (e) our training approach develops neural activity patterns similar to the target latent kinematic variables (Fig. 7). We found that the developed single neuron and population patterns were rich in postural information, providing an example use of LSTT to explore cortical function (Fig. 7 and Fig. 8).

Accuracy is improved by a non-linear layer

Applying the above procedure led to the neural control of a hand prosthesis able to switch between grip types and select grip target angles. After implantation, we trained two monkeys to control the hand prosthesis using neural activity following the LSTT progression. Fig. 3 compares the control performance of the complete decoder to different control modalites after subjects successfully completed Phase (iii) of the training. At successful completion of Phase (iii), subjects were able to grasp the handle using the prosthesis with a precision or power grip and two different orientations (40° and 180°). Transitions from the no grip, resting posture, to the grip postures were less noisy using our approach than alternative decoders (Fig. 3A,B). Existing non-linear architectures were not able to correctly predict the grip types or achieve the desired grip configurations, even when the latent variables were developed. As an example, the output of a FORCE RNN decoder, successfully used in reaching experiments[46], is shown for both subjects (Fig. 3A). Performance with the FORCE RNN was too low for grasping and the experiments were aborted. To determine the effectiveness of the combined Kalman filter and a non-linear layer, we compared our decoder to a Kalman filter decoder in 5 and 6 online sessions in both animals. On each decoding session, after randomization of the decoder order, the two independent decoders were trained and tested independently (Fig. 3B-F, Methods).

We found, in general, that trajectories were less variable using the hybrid decoder (Fig. 3B). We assessed the accuracy of the prosthesis by comparing it to native hand accuracy (Fig. 3C-E). Noise levels were lower using the non-linear layer and the distribution of individual trial noise values was close to native hand variability (Fig. 3C). Holding accuracy, evaluated as the variance of the BCI while the subject is holding a grip was also close to that of the native hand (Fig. 3D). The amount of covered trajectory in kinematic space, determined by the ratio of executed to intended (BCI) or average (native) trajectory, was also lower for the hybrid decoder (Fig. 3E). Accuracy of the trajectory was also reflected in task performance, which was higher when the decoders are combined (Fig. 3F).

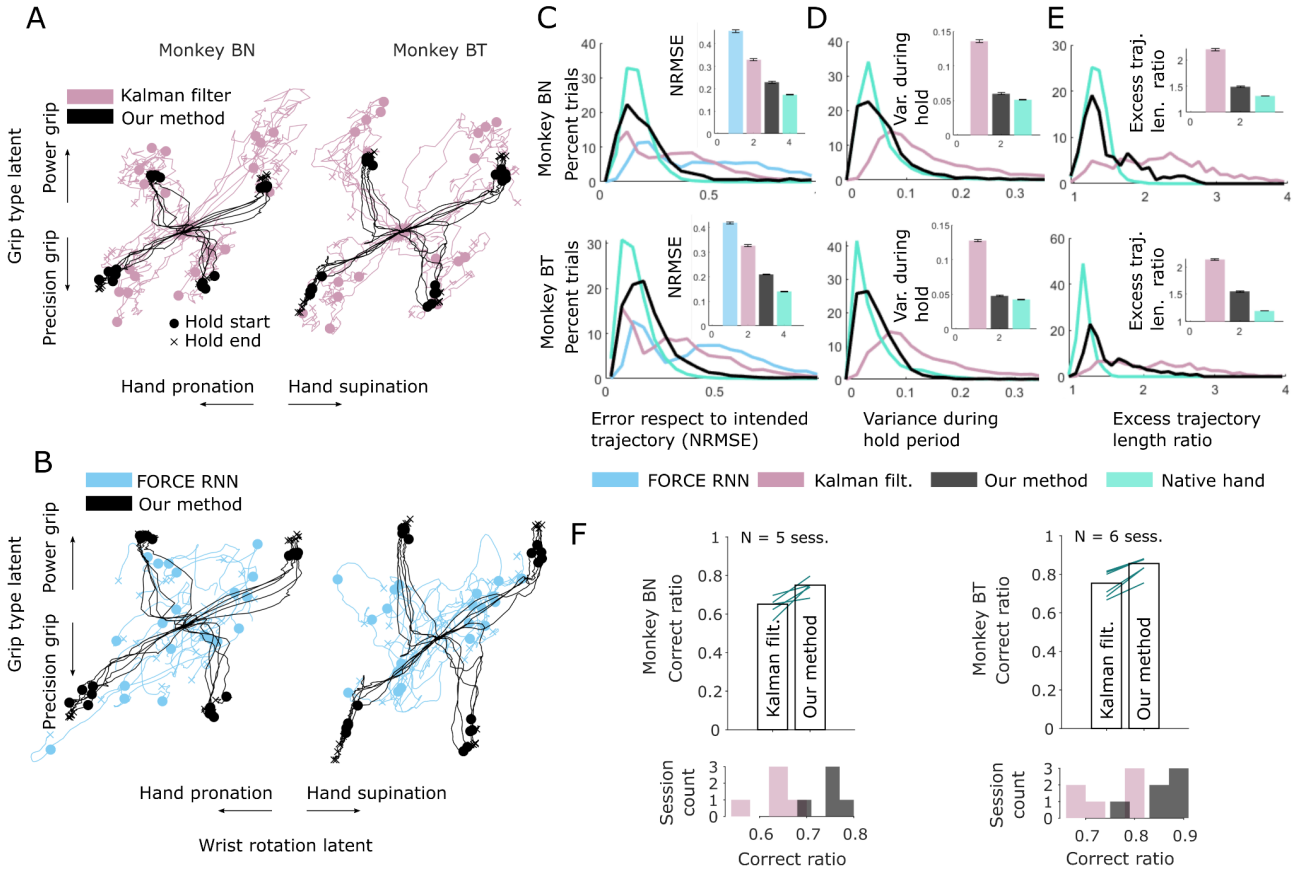

Figure 3. Performance comparison of full LSTT to only latent variable training with a Kalman filter (KF) at the end of Phase (iii). (A-B) Example trials from one comparison session seen as trajectories on the wrist angle and grip type latent variables. Trajectories start at the center (no grip) and end on the hold points corresponding to 40° and 180° handle orientations (circles and x's). (A) Example trials using the FORCE RNN compared to LSTT online trials. Due to the low performance of the FORCE RNN online, data was produced offline using the neural signal during the LSTT part of the session. (B) Example trials using the KF decoder and full LSTT online. (C-E) Comparison of grasp accuracy on native hand, KF and LSTT sessions for monkey BN (trials/sessions: native 2454/6, KF 1492/5, LSTT 756/5) and monkey BT (trials/sessions: native 1142/6, KF 1186/6, LSTT 1121/6). Native grasps were performed before BCI training. Here, output joint angles are compared. Inset bars represent the median and standard error. For the native hand, trial averaged trajectories and trial averaged hold poses were used as the idealized target. (C) Distribution of individual trials normalized root mean square error of the attempted to the idealized joint trajectory. As a comparison, the offline values of the FORCE RNN are included. (D) Distribution of how much joint angles vary during the 500ms hold period in relation to the idealized joint hold configuration. (E) Distribution of the ratio of executed to idealized trajectory length. (F) Lines: trial performance for the same sessions of monkey BN and BT for each decoder type. Bars: median performance. Trial wise performances are significantly different ($p<0.001$, Clopper-Pearson estimation for the Bernoulli process). Insets: distribution of session performances (same colors as C-E).

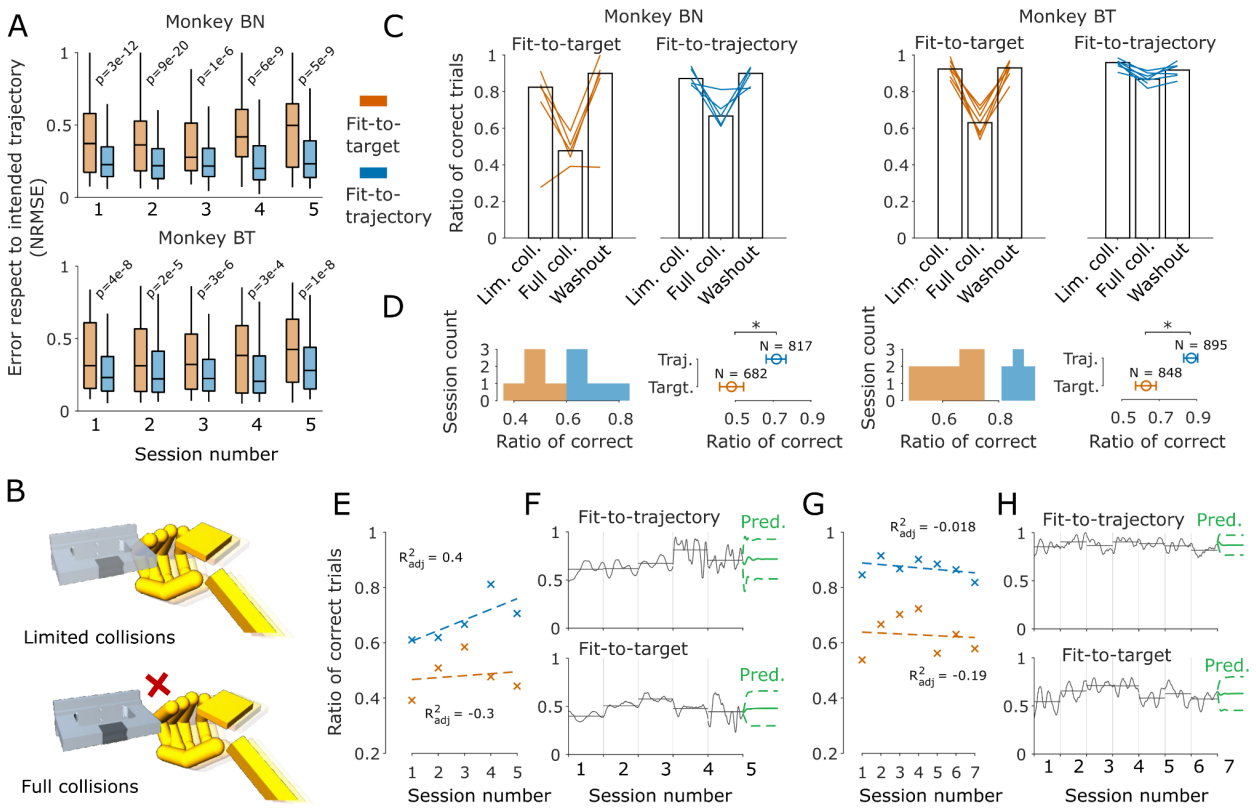

Figure 4. Comparison of fit-to-target and fit-to-trajectory strategies. On each decoding day, the initial fitting method was picked randomly and we then followed our three stages procedure for each decoder type. (A) Single-trial error distributions for the fit-to-target (red) and fit-to-trajectory (blue) strategies on 5 online decoding days for each monkey. The box plot displays the median, 25th and 75th percentiles, and range of the distribution. p-values: two-sided Wilcoxon rank-sum test. (B) Illustration of the limited-collisions/collisions task. During the limited collisions task, only the fingertips and precision button are able to collide. The rest of the virtual hand can go through the target handle and achieve correct grips just by matching the right pose. When collisions are enabled, the hand would collide with the handle if not properly prepared for grasping. (C) Performance comparison on the collision task for 5 sessions of monkey BN and and 7 sessions of monkey BT. Lines: percentage of correct trials for each part of the task per session. Bars: median performance across sessions. (D) Percentage of correct trials for the full collision task for each session (left) and across all sessions (right, median and 0.999 CI, Clopper-Pearson estimation for the Bernoulli process). (E-H) Performance over days for the collision part of the task. (E) Trend of the performance for the two fit methods for monkey BN (linear regression, adjusted R^2 value displayed next to each prediction). (F) Performance over sessions and forecast for fit-to-trajectory (top) and fit-to-target (bottom) single trials (20 steps moving average). Horizontal lines show the session means, green solid line shows the prediction and dashed lines the 95% confidence intervals of an ARMA(4,4) forecast. (G,H) Same as (E,F) but for monkey BT.

Daily training to produce the intended trajectory improves performance in a collision task

The daily fitting of neural activity to kinematic features is at the core of our approach. The main purpose of this strategy is to develop the neural patterns for the quick traversal of states in kinematic latent space and thereby for the quick and robust configuration of the hand effector. While intention-estimation approaches typically assume the purpose of the subject is to reach a target in space[25–27,20], we assume the intention of the subject is to follow a trajectory to reach a given hand configuration. This should be particularly useful in hand configuration control, where intermediate states are not obvious positions in space, and accuracy of the transitions is crucial to achieve end-effector configurations successfully.

To test the advantage of our approach, we compared our intention-estimation method to a fit-to-target approach, which has been shown to boost performance over methods with no intention estimation[25]. We first looked at a basic difference: the capacity of our method to preserve trajectories. In 5 decoding sessions, we tested both decoders following the three stages training procedure (Fig. 2F,G,I) alternating Stage 2 with a

decoder trained with fit-to-target or fit-to-trajectory (random selection of decoder type, Methods). We did not use the non-linear layer for a fairer comparison between the two methods. Error to the intended trajectory was lower using fit-to-trajectory for all sessions and in both animals (Fig. 4A, $p < 0.001$, two-sided Wilcoxon rank-sum test). This result is expected, as subjects were not forced to alter the trajectory in kinematic space and the simplest path was to achieve the target. We proceeded to test the effect of obstacles in the transitions.

The major advantage of fitting to trajectory is that the user should be able to quickly perform transitions on the desired high-dimensional state of the effector. We trained both animals to use the BCI under limited collisions in the virtual environment. In this configuration, only the center of the handle and fingertips were able to collide. However, the virtual environment could also be configured to a full collision mode where the full fingers, box and grasp handle were able to collide (Fig. 4B). We hypothesized that incrementing the difficulty to achieve the grips through collisions was a way to detect if the subject could react quickly enough and achieve the desired transitions. We hypothesized that it would be harder to achieve such intermediate transition steps if the decoder was trained to only reach specific targets. To set a baseline and detect possible frustration effects, we compared both training methods in a three parts experiment: 1) control under limited collisions, 2) control under full collisions, and 3) a washout under limited collisions. Enabling full collisions caused a performance drop for both strategies but fitting to trajectory preserved better performance (Fig. 4C,D; 5 and 7 comparison sessions for animals BN and BT, 0.999 CI, Clopper-Pearson estimate of the Bernoulli process). After finishing the full collisions part, performance returned to baseline during the washout.

One possibility under fit-to-target control is that the subject might be able to learn to overcome the obstacle by thinking about the intermediate states as targets. Although we cannot fully rule out this possibility (our subjects were not extensively trained in fit-to-target approaches), we looked for performance improvement during the fit-to-target/fit-to-trajectory experiments. Could the subject learn these intermediate steps over sessions? Linear regression on the session performance (Fig. 4E,G) and ARMA(4,4) forecasting on the single-trial moving average performance (Fig. 4F,H) failed to detect clear improvement trends in both training methods (except in fit-to-trajectory for monkey BN). We also did not find trends of improvement in follow-up comparison sessions of animal BT (12 total), indicating that it was still hard for the subjects to execute the transitions using the fit-to-target approach, and fit-to-trajectory provided the most convenient solution.

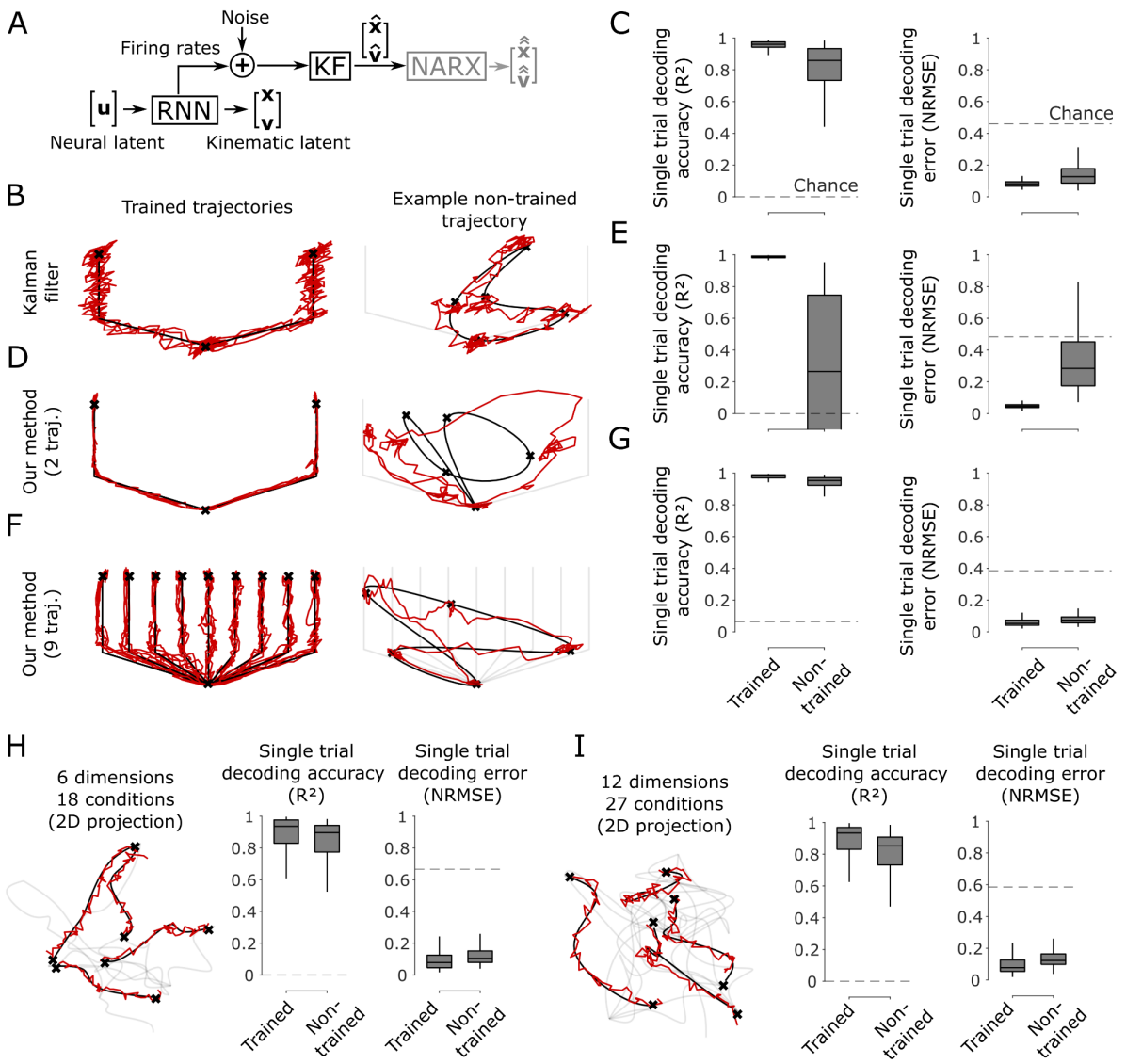

Figure 5. The LSTT strategy extends to untrained trajectories and high-dimensional latent spaces in simulations. (A) Illustration of the model. Recurrent neural networks (RNN) were trained to produce latent position and velocity vectors (**x**, **v**) from a latent neural vector representation **u**. The simulated RNN unit activations were used to train a Kalman filter decoder using fit-to-trajectory only or using the full LSTT strategy. (B-G) Simulations for an example 2D latent space with different decoder combinations and with varying number of target trajectories. Hold states are marked with x's, representing points where the BCI remains steady for 500ms. (B) Decoder fit to only two intended trajectories and without a non-linear layer. (Left) The two trained trajectories (black) and example executed trajectories (red). (Right) Example trial of a trajectory not observed by the training algorithm. Trajectories used for training are in light gray. (C) Ensemble accuracy and normalized root mean squared error with respect to the intended trajectory for all trials and RNNs trained with the two trajectories in (B). The box plot compares the performance for the test trials of the trained trajectory and non-trained random trajectories. Boxes: median, 25th and 75th percentiles, and range of the distribution. (D,E) Same as (B,C) but the combined LSTT decoder was used. (F,G) Same as (D,E) but 9 trajectories were used to train the decoder. (H,I) 2D projections of simulations for a 6D and 12D latent space using LSTT trained with 18 and 27 trajectories respectively. As above, example target (black), example executed (red), and trained (light gray) trajectories, and hold points (x) are shown. (C,E,G,H,I) Box plots show the ensemble result of 10 independent RNN simulations for each configuration. Chance levels for all plots are depicted as dashed lines (Methods).

Our strategy generalizes to untrained conditions in higher-dimensional latent spaces

Full hand configuration control involves 28 degrees of freedom with a lower, but still potentially high subspace of latent control variables[55], and a high number of target hand configurations[18]. Achieving intermediate states (non-canonical grip types) in this space is important for successful manipulation and obstacle avoidance. Would our fitting approach extend to untrained conditions in more complex latent spaces? We tested this in

simulations by training RNNs specialized to produce grasping dynamics[34]. We created arbitrary kinematic latent spaces and used the RNNs to create artificial neural latent spaces corresponding to these kinematics. We used the RNN artificial units to train our decoders (Fig. 5A).

To get a better understanding of the conditions required for high-dimensional decoder training, we tested first the simulation tool in low dimensions (Fig. 5A-G). We started by training a 2D latent space with only two trajectories using the RNN, fit-to-trajectory and the Kalman filter. The decoder was able to successfully produce the trained trajectories (Fig. 5B, left). We then used the RNN to produce arbitrary intermediate trajectories with random 500 ms stops within the latent space while keeping the decoder fixed. The decoder was able to produce the intermediate trajectories and stops (Fig. 5B, right). Accuracy was lower and error higher for the non-trained trajectories with respect to trained, using only the Kalman filter (Fig. 5C; 10 independent RNN simulations). Confirming our experimental results (Fig. 3), addition of a non-linear layer (full LSTT) increased accuracy and reduced error for the trained trajectories. However, this was not the case for the untrained trajectories (Fig. 5D,E). We found this can be resolved by adding internal trajectories to the training set, increasing the interpolation capacity of the non-linear decoder (Fig. 5F,G).

Low dimensional tests revealed two important requirements for the accurate traversal of the latent space through non-trained conditions: a) use of a non-linear layer, and b) training of internal trajectories to build the inner space of the latent variables. Can this inform the training in high-dimensions? We tested this in a 6 and 12 dimensional latent space with arbitrary trajectories (Fig. 5H,I). Increasing dimension required increasing the number of inner trajectories to preserve similar levels of accuracy and error in the non-trained case (Fig. 5H,I). Around 18 and 27 inner trajectories were required to produce acceptable accuracy and noise levels in the 6 and 12 dimensional cases respectively. This is likely because there is less chance of trajectories traversing as many dimensions as the number of dimensions increases. The number of required intermediate trajectories did not grow exponentially. Intermediate trajectories were generated randomly, indicating that a good coverage of the inner space would be required.

Incremental training of latent variables eased learning of the intermediate angles

Our RNN simulations highlighted the importance of learning intermediate trajectories. In this section we describe in detail our incremental training procedure and its extension to the training of intermediate angles. We chose three non-linear latent variables roughly corresponding to grip aperture, wrist rotation, and grip selection (Methods). In this 3D space we designed hand state transitions useful to achieve the target handle angles. Subjects were trained to progressively learn these three variables. Figure 6 displays the corresponding training trajectories in latent space. In Phase (i) subjects learned to open and close the hand to the 90° handle orientation using a power grip (Fig. 6A). In Phase (ii) subjects learned to pronate the wrist by grasping the handle at the 40° and 180° orientations (Fig. 6B). In Phase (iii) subjects were introduced then to grip selection by grasping the same handle orientations using two grip types (Fig. 6C; power and precision). Bends in the trajectories emulate the curvature in the latent kinematics (Fig. S1, Methods).

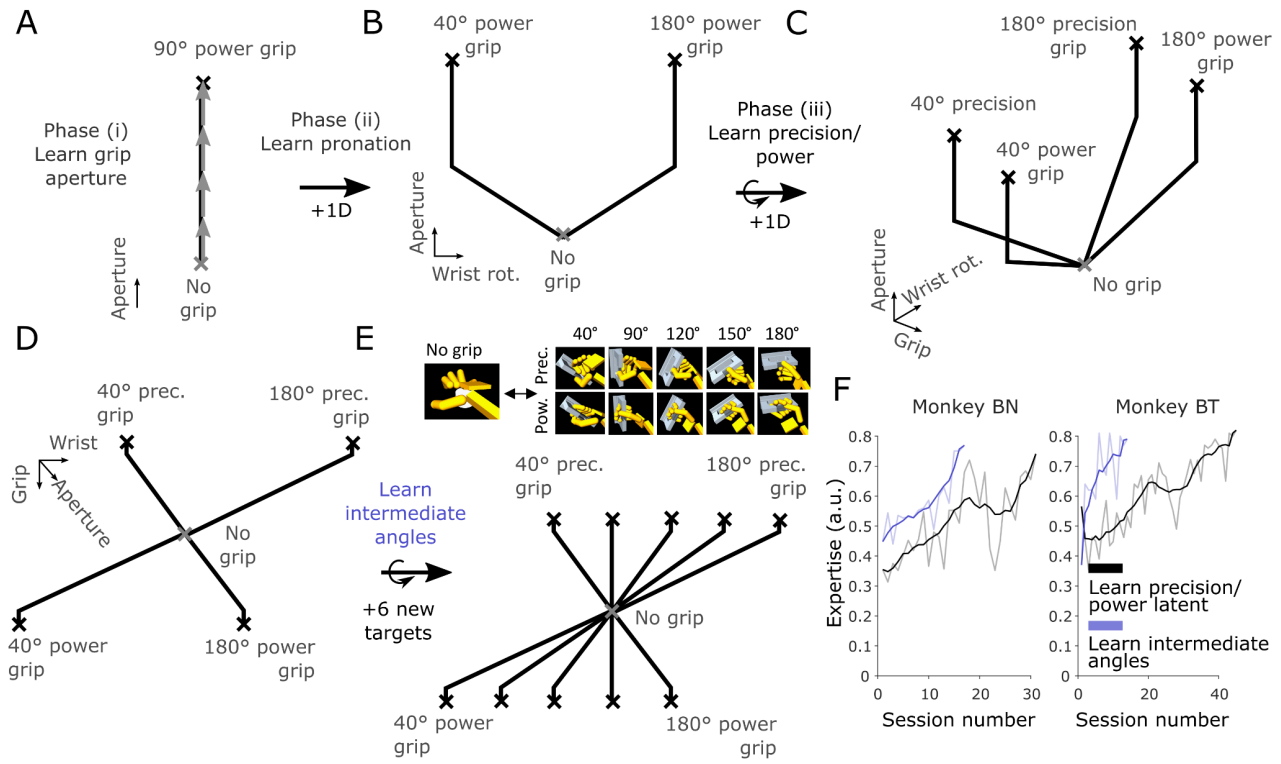

Figure 6. Detailed view of the incremental latent variables training and training of intermediate angles. (A-E) Schematic of the trajectories in kinematic latent space used to train the BCI. Note that the subject does not see these trajectories but only receives feedback from the grips on the screen. Bents in the trajectories aim to emulate the curvatures in the latent kinematics (Methods). (A) In Phase (i), subjects were trained to control the grip aperture dimension by changing the hand posture from open to hand closed. This is a single dimension and full brain control was achieved in 2-3 sessions. (B) In Phase (ii), subjects learned moving in the wrist rotation latent by selecting one the two wrist angles required for a power 40° and 180° grip. Perfecting this phase took 44 and 27 sessions for monkey BN and BT (Fig. 7C). (C) In Phase (iii), subjects learned the grip type latent which selected the grip type (power or precision) in the virtual hand prosthesis. (D-E) In addition to the training phases, subjects learned the intermediate target angles. For this we asked the subjects to also grasp the handle at the 90°, 120° and 150° orientations with both grips. In (D,E), we display the latent space from the top view (aperture points outside of the page) for clarity. Pictures in (E) display the virtual reality equivalent of the grips. (F) Learning curves to achieve "expertise" in phases (ii) and (iii). Expertise was defined as the ratio of correct trials times the average BC level on a given decoding session.

We argue that our training procedure builds a "scaffold" of neural patterns, and learning the intermediate angles in the rotating handle task would be easier than learning the different grip types. To test this, we extended the training beyond Phase (iii) to a novel phase, where subjects were asked to grasp the remaining intermediate angle configurations of the handle with both grips: 90°, 120°, and 150°. Training was identical as in the previous phases, with progressive control of the BCI through a BC level and intermediate trajectory objectives (Fig. 6D,E). We measured "ease of learning" as the time it took to learn each training step to an expert level. We defined expertise in a given session as the product of the ratio of correct trials to the average BC level achieved. To compare to other phases, we looked at the time it took our subjects to reach a given expertise level threshold (expertise >= 0.80). In comparison to learning the wrist rotation (44 sessions for monkey BN, and 27 sessions for monkey BT, next section) and the grip type selection (30 days for BN, and 43 days for BT, Fig. 6F) it took only 18 and 17 days to reach the expert level in the intermediate angles task (BN and BT respectively, Fig. 6F). Although we cannot rule out the opposite case was faster (learn intermediate angles first and then move to grip switch control), we observe that learning the intermediate angles was shorter than learning the new grips. In summary, our approach seems to facilitate the learning of novel intermediate conditions and can be extended to train intermediate state configurations.

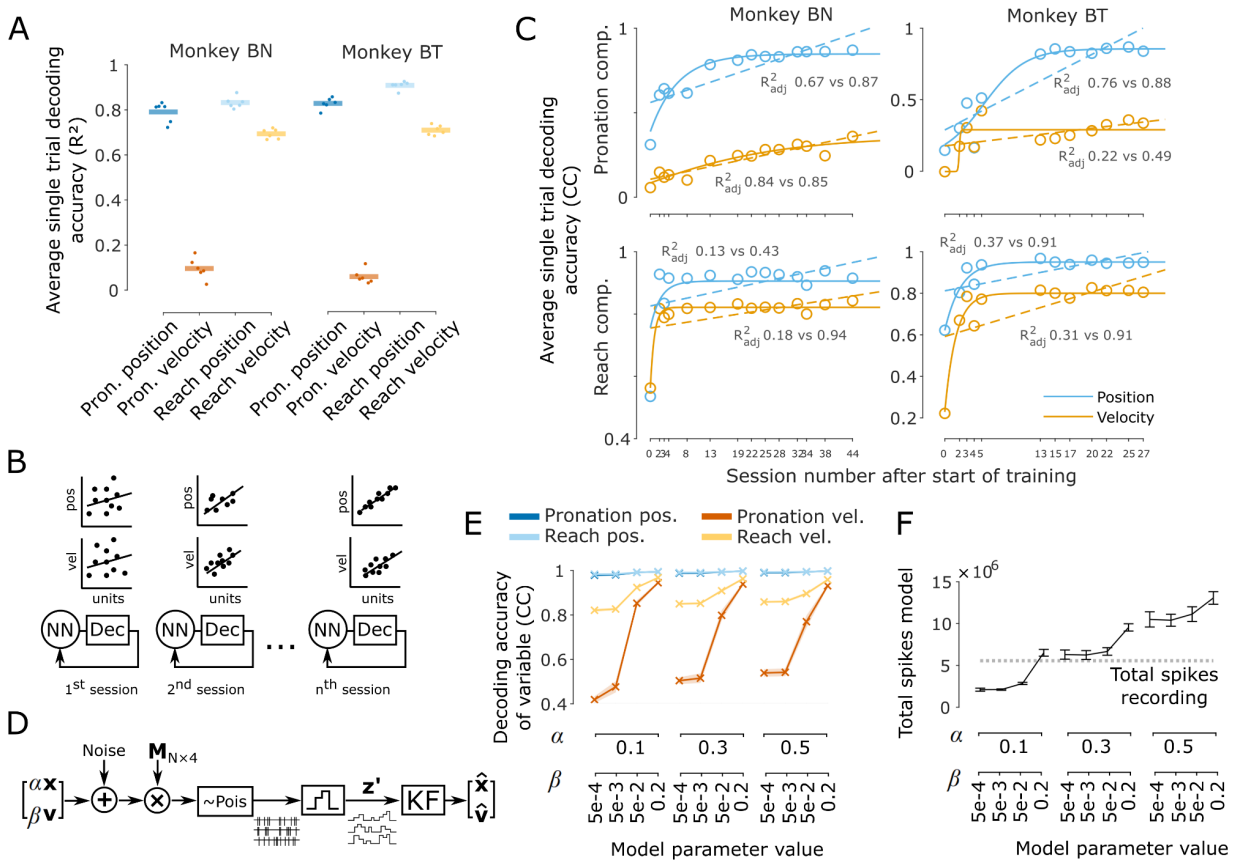

Figure 7. Neural latent space evolved as kinematic latent variables. (A) Correlation of decoded single-trials to trained target trajectories for six late-stage sessions of Phase (ii). Horizontal bars are the averages of all sessions (Pearson correlation coefficient). (B) Training of latent variables can be seen as the process of fitting a neural network NN in combination with a decoder (Dec) to produce a target variable. (C) Correlation of decoded single-trials to trained target trajectories during learning of Phase (ii) of the task. Lines show linear (dashed) and sigmoid (solid) fits over sessions and text the goodness-of-fit (linear vs sigmoid, adjusted R^2). (D) Encoding model. Kinematic signals are tuned with parameters α for position and β for velocity. The model mapped Z-scored kinematics injected with noise to a high-dimensional space on N simulated units (N=130). The artificial neural code was Poisson discretized. A decoder (Kalman filter) at the end tries to recover the input signal. (E) Correlation of decoded single-trials to trained target trajectories for different values of α and β using the encoding model on one BCI recording of monkey BN. For each point in the plot, we performed 10 simulations and plotted the average. Shades: 2×STD. (F) Average total amount of spikes for each point of (E). The dashed line is the total spike count of the original BCI recording. Error bar: 2×STD.

Neural activity patterns evolved as kinematic latents

We trained a hand prosthesis able to switch between grip types and accurately acquire grasp targets (Fig. 3). We also trained novel kinematic patterns and without concurrent hand movements (Methods). We were curious about the geometry of the neural population activity supporting control of the BCI. Our Kalman filter uses position and velocity of the latent variables as its main state (Methods), making these the most relevant patterns to look for in the population. On six late stage decoding sessions of training Phase (ii) (performance > 80% and BC 100%), we looked for correlates of these two major variables in the neural activity. For each session, we used the neural activity of the initial portion of the recording (first 10 trials of each condition) to train a decoder and predicted the single-trial latent kinematics of the rest of the session. In contrast to the online BCI, we used two independent Kalman filters (one for position and one for velocity) to prevent cross-talk between decoders. We compared the predicted offline kinematics to the intended online kinematics on each trial and averaged the results for each session. We chose the intended kinematics as this would give us a measure of how much the neural population "fitted" to the intended variables. Neural activity correlated

well to the position and velocity variable but showed a higher correlation (Fig. 7A, Pearson correlation) and variance explained (Fig. S2A, R^2) to the position variable.

Correlation to the target trajectories implies that at the end of the training, neural patterns acquired some of the shape of the latent space for successful control of the BCI (Fig. 7B). Were these correlations spurious? We looked back to the training of Phase (ii) which spanned 45 and 28 sessions preceding the six sessions above (BN and BT, respectively). On these sessions, we looked for days with performance above 60% and more than 30 trials per condition and selected 12 and 14 relatively even spaced sessions (BN and BT). Session averaged correlation coefficients showed that the target patterns were not present at the start of the training, and correlation evolved over time (Fig. 7C).

Of particular interest to motor neuroprostheses is our finding of abundant position information with respect to velocity. An important aspect to consider is that velocity-like signals might be thought of as "harder to catch" as they are usually fast sharp spikes that can be hidden by noise. We argue however that this is likely not the case if the neural population is considered, as in the population, linear combinations of position like signals can easily produce velocity like signals (see for example Fig. 8C,D) and the fitting procedures used to train the KF (Fig. 7A-C) should be able to find such combinations. Another potential problem is that the sampling resolution is not enough to catch such fast transitions. To evaluate this, we built an encoding model where two tunable parameters (α and β) would control the amount of kinematic information encoded in artificially generated spikes (Fig. 7D, Methods). Same as above, we used KF decoders to recover the kinematics out of the artificial spikes and compared them to the original kinematics using correlation coefficients (Fig. 7E,F). We kept the bin sampling parameters equal to the online setting. For a BCI session of monkey BN, a scan of possible values for α and β suggested that velocity information was recoverable from the signal (Fig. 7E). Confining the parameter space by the total amount of spikes captured in the recording (a proxy measure of total information available) indicated that two regimes: a = {α = 0.1, β = 0.2} and b = {α = 0.3, β = 5e-4 to 5e-2} matched well our total spike counts (Fig. 7F). This suggests that our sampling capacity could have captured velocity information if it was there (regime a) but our results (Fig. 7A,C) suggest that velocity information was scarcer (regime b). We found similar results using data of monkey BT (Fig. S2B,C).

Neural activity during native arm use reflects abundant posture information

Data from the BCI task provided evidence of abundant position information during brain control. The presence of position information is important for accurate control as it provides a reliable signal to maintain stable hand postures. Previous studies have also found ample posture information during native hand control[35–41]. Was the asymmetry between postural and velocity information during BCI control a product of the training or a property of the circuitry? In an attempt to disentangle this effect, we looked at recordings of the grasping task using the native hand before the BCI training started. Example PETHs of single unit activity showed a mixture of low and high frequency components resembling typical single joint angle position and velocity, although notably, neural signal peaks were less sharp (Fig. 8A). To see if this property was common in the population, we took averages of the normalized Fourier transforms of the three signals and observed that the neural activity did not contain as many sharp components as the velocity signal (Fig. 8B).

We further determined the ratio of position to velocity information through pairwise correlation and cross-validated regression. These provide two complementary views of the correlative relation of cortical signals to kinematics. Pairwise correlation of the trial averaged patterns (Fig. 8C) is robust to noise (through averaging) but misses signals that can be produced by the population (e.g. linear combinations). Input to the regression is not trial averaged, being subject to noise effects, but detects potential population-wise linear combinations (Fig. 8D). On average, individual units of one recording tended to be more closely related to position than to velocity patterns (Fig. 8F top, t-test and Wilcoxon signed-rank test $p < 0.001$). This result should be robust to the potential effects of noise on the more sensitive, velocity sharp transition. Reducing the

signal to its shared principal components[21,56] produced similar results (Fig. 8F bottom PCA, 90% of variance). We found similar results on six recordings of each animal (Fig. S3A). We found a similar asymmetry comparing single trial prediction of individual position and velocity kinematics (Fig. 8G, one session, R^2, paired t-test and Wilcoxon signed-rank test $p < 0.001$). Similar results were obtained in six sessions of each subject (Fig. S3B) and the result was independent of whether the joint was part of the arm or the hand (Fig. S3C).

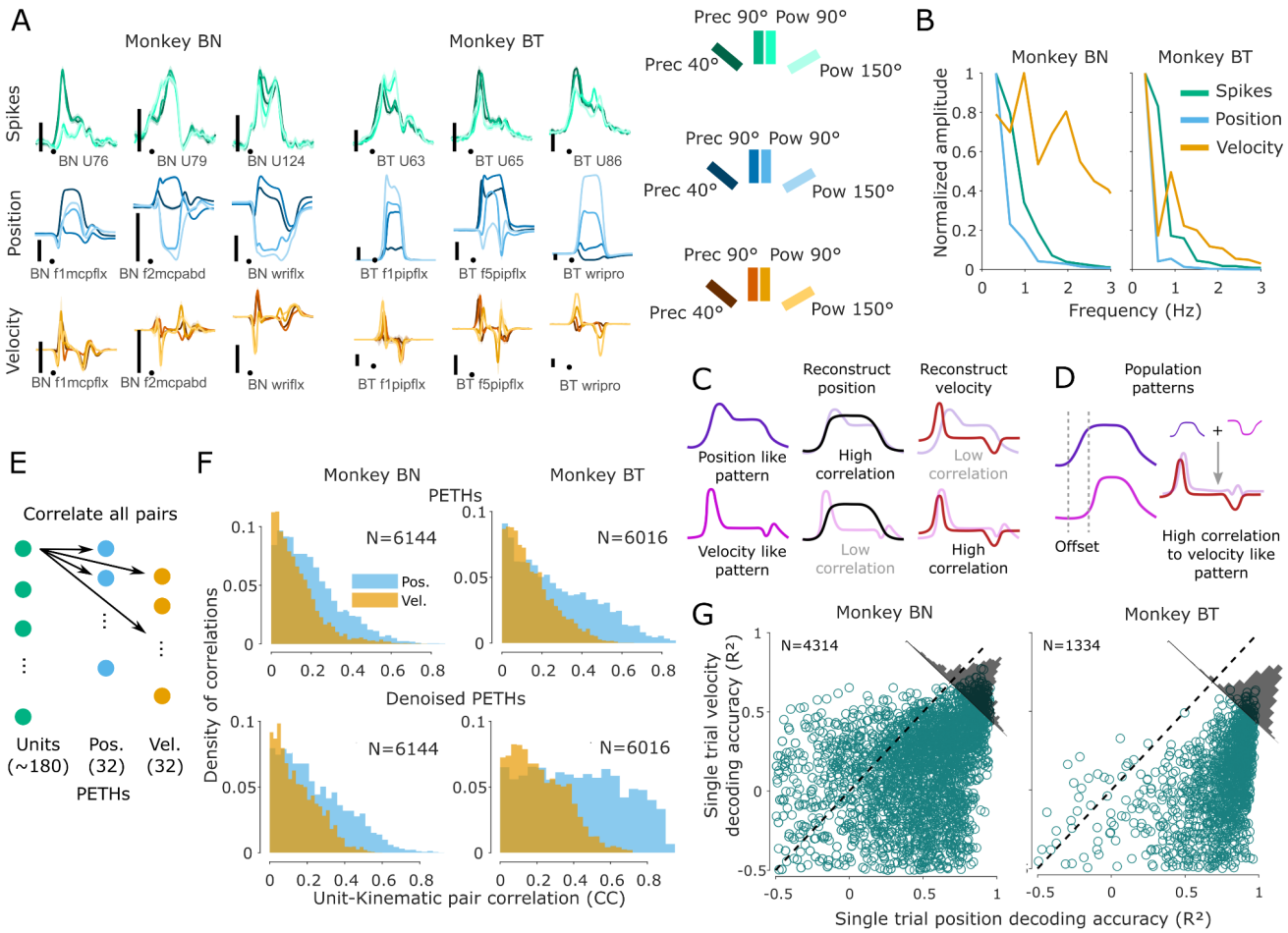

Figure 8. The amount of position information is proportionally higher during native grasping. (A) Peri-event time histograms (PETH) of example units and kinematics during the native task. Here positions are joint angles and velocities are joint angle velocities. Corresponding handle orientations are displayed on the right color bars. Dot indicates movement onset. Scale bars: 50 Hz, 10°, 100°/s, 1s. (U: unit code, f1: thumb, f2: index, f5: little, mcp: metacarpophalangeal joint, pip: proximal interphalangeal joint, wri: wrist, flx: flexion, abd: abduction, pro: pronation, Prec: precision, Pow: power). (B) Average of the Fourier transforms from all PETHs of one native arm recording of each monkey. Fourier transforms were applied to the PETHs, then normalized to the largest Fourier amplitude of each signal, and then averaged. (C) Schematic of possible unit/kinematic correlations. Neural patterns (purple, magenta) can resemble position patterns (black) or velocity patterns (red). (D) However, velocity-like signals can also be produced in the population through linear combinations of patterns with offsets. Here, the inverted (negative) magenta pattern plus the purple pattern produces a velocity-like pattern. (E) Illustration of the pairwise correlation in (F). (F) Top: Pairwise correlations (absolute value) distribution of all available kinematic unit/position and unit/velocity pairs for one native arm recording. For a single unit and a single kinematic, patterns for each condition were compared independently and the CC averaged. N: number of unit/kinematic pairs. Bottom: same but here only 90% of the variance was kept (PCA). Distributions are significantly different ($p<0.001$, t-test and Wilcoxon signed-rank test) (G) Comparison of decoding accuracy of position and velocity kinematics from neural activity in one native arm recording. Each position (32) and velocity (32) kinematic was predicted from neural activity on single trials using independent Kalman filters. The quality of the prediction was measured by comparing to average kinematics through R^2. Each point corresponds to one trial and one kinematic. Overall, the position decoder predicted the position patterns better than the velocity decoder the velocity patterns ($p < 0.001$, paired t-test). R^2 values below -0.5 were discarded.

Discussion

We presented a BCI training strategy that enables accurate prosthetic configuration control through training to target trajectories and progressive introduction of degrees of freedom. The fitting method generalized to more complex configurations and eased grasping in an environment with obstacles. Addition of a non-linear layer to the decoder boosted accuracy of the prosthesis. We found that daily and incremental training to the set of target trajectories developed neural activity patterns resembling the kinematic activity. Such patterns matched the position and velocity variables input to the decoder, but in particular position variables were present, confirming previous observations and facilitating accurate control. Taken together, these findings propose a novel way to complement training protocols for the challenge of dexterous prosthetic manipulation and offer a training approach to shape neural activity patterns for basic research of cortical activity.

The advent of modern neural recording and motion capturing techniques[21,57,23] offers an opportunity for the development of advanced prostheses but also demand novel methods to achieve full high-dimensional control. This work aims to start answering this question: what are the right abstractions to train a BCI when we want to traverse a control space we cannot easily visualize? We suggest that the hand configuration control space is one of such spaces. In our method, we start by collecting kinematic data and use that to build a kinematic state space. Given that the state space is that of the hand joint angles and high-dimensional, its geometry is not obvious and requires careful consideration and design of the latent variables used for training. It also requires a mapping from the latent state to the high dimensional state space for final output. The target latent space might be curved, requiring control methods to handle these curvatures. Even though our training data is based on the two basic grips of the grasp hierarchy[17], the same principle could be applied to larger datasets of grip configuration selection[18] once the right subset of variables is established[55]. There are two potential alternatives to our approach: a discrete grip classifier or a continuous control space where grip transitions are linearized. We believe both alternatives are sub-optimal as they will limit the full expression capacity of the limb or fully articulated hand. As able-bodied hand use and our RNN simulations demonstrate, the brain's expressive capacity could in principle take over and discover the full potential use of the effector after some basic training.

We propose incrementally training latent control variables to achieve precise multi-grip control. Although similar ideas have been presented in a research context[32], we expand on this idea with an alternative protocol in several forms. First, we demonstrate this in more than one degree of freedom rather than only expanding the neural latent space to deal with a perturbed control mapping in the same two dimensions. Second, we demonstrate this in an applied setting, not as a form of exploration of the capacities of the intrinsic neural latent space. Third, we show that this is possible for more than one control variable (e.g. not only velocity). Fourth, we demonstrate this in new brain areas. Finally, we also show that our protocol achieves not only expansion but shaping of the neural latent space, which is crucial in our case for the precise execution of grip attempts. Achieving precise grip configurations is important for hand control, yet this is only one of the functions of the hand. This “shaping” is only one of the potential neural patterns required for full hand control. Full hand control entails hand shaping but also individuated finger movement, non-prehensile actions, hand writing, etc. Our protocol can be one of a set of complementary protocols[19,32,15,20,58] that will be required to develop a fully neural latent space able to drive an articulated hand prosthesis.

Our daily training paradigm enables configuration of the artificial hand through quick transitions. This training approach is related to intention-estimation approaches such as the ReFIT algorithm[25–27,20]. A common feature in previous approaches is the assumption that movement attempts point to an end target[28]. In our case, in addition to targets, we also consider trajectories. Our Kalman filter estimates position and velocity and

integrates velocity information into the position estimate[59,60]. The inclusion of the position estimate was motivated by multiple reports of the availability of the position correlate[6,35–37,39,41], and interest in the effects of BCI training on neural latent space shaping. Our hypothesis was that training using both variables would develop two latent spaces, one related to position and one to velocity. An ideal velocity control neural latent space can be visualized as an M-dimensional sphere of variable radius r, with M the degrees of freedom of the effector. A point on the surface of the sphere represents the subject's intent to move in one particular direction of the configuration space, and r represents the intended speed. The corresponding ideal position control neural latent space can be viewed as an irregular M-dimensional shape whose boundaries represent limits on the workspace of the effector (e.g. joint limits). Transforming the neural state to the kinematic state is linear operation (plus an offset) in the measurement update equation of the Kalman filter[60], thus the filter's prediction is an affine transform. Inclusion of the position estimate and trajectory details ensure a better alignment between the neural and kinematic latent spaces as more information about the shape is available to the fitting procedure. Our latent space shape results show that there is abundant position information available. When comparing the trained decoder in our collision task, inclusion of the trajectory provides information on the required transition and does not require the subject to learn necessary intermediate steps to avoid collisions. This is at the core of our approach, as position information provides additional information during the quick action required to achieve a hand configuration when the subject retarget the grip to an intermediate step. Note, however, that this might be a particularity of hand configuration control. For example, reach-oriented applications might discard position information[14,25] (but see also Stavisky et al. (2018)[42]).

We propose the use of a hybrid scheme to improve accuracy of the kinematic output. The latent variable output of the Kalman filter is fed to a two layer recurrent neural network[53] trained to produce the target latent trajectories. This configuration was motivated by the principle of an "information bottleneck" (e.g. as in variational autoencoders[44,45]) in the context of hand state dimensionality reduction[55]. Following this principle, we first trained the subject to produce the output without the non-linear layer to ensure robust latent features were developed. In this respect, noisy input of the Kalman filter is advantageous as it should pressure the brain to produce the correct patterns to achieve the targets.

Other practical factors motivated this arrangement: non-linearities on the transitions and hold states, potential of extensibility to more complex control spaces, and low tolerance to noise of pure non-linear/RNN solutions. A double decoder setup comes at the cost of complexity. We believe however this is justified for our application. The Kalman filter has been applied in numerous BCIs[25,10,14,28,20] thanks to its simplicity, robustness to neural noise, and compute cost, which is essentially two matrix multiplications once the filter reaches its steady state. Yet, characteristics of the neural data limit its accuracy[25,47,48], requiring alternative or hybrid solutions[46–49]. Our approach combines the best of two worlds: robustness to neural noise and expressive capacity for non-linearities such as hold states. Although we present a non-linear decoder online for maximum three latent variables, the capacity of the decoder extends to multiple dimensions as our RNN simulations show.

We found that training of the latent variables shaped velocity and position-like neural population patterns in a causal setting[1,2]. These results have important implications for brain controlled hand prostheses. Evidence for position or velocity correlates in motor cortex has traditionally been mixed[61–64] with a tendency to find superior direction of movement or velocity information in arm areas during reaching[65,66,61], and superior position information in hand areas during grasping[6,35–37,39,41]. A safe claim is that both correlates are present as observed in naturalistic movement studies[35,36], with a possible exception for the magnitude of the velocity vector (speed) which tends to be poor[47,49,67]. In BCI control of reach, decoders typically focus on velocity[8,11,10,25,42,20], as velocity is a natural variable for hand or cursor navigation and position is considered a "nuisance" variable[25,42], although other studies showed this depended on concurrent movement of the subject[42]. We employ both position and

velocity in our decoder. Our results show: 1) there is naturally posture and velocity information during native hand control, 2) if you train for it, you can develop both position and velocity patterns, 3) these patterns evolved with the decoders, showing their usefulness for the output. A developed set of patterns with both velocity and posture information is useful for accurate hand control and both variables can be a valuable source of information for the challenge of prosthetic hand control.

Our work shows that it is possible to shape neural latent spaces to specific target functions in the context of motor control. By fitting to a set of target position and velocity target trajectories, daily training evolves these patterns and they can be volitionaly recalled by the BCI user. Recent interest on the evolution of neural latent spaces in motor cortex[29–33] has shown that neural patterns are constrained to a covariance space[30], training can expand this space[32], and activity prefers remapping to reconfiguration[31]. A protocol for latent space shaping offers an opportunity to explore the evolution and geometry of this latent structure. An example of potential explorations is the learning of the intermediate grip targets. Although subjects already had experience with the intermediate angles task through native arm control, learning of the intermediate targets was not immediate but also not as demanding as learning the wrist rotation or the grip type. Future research is required to explore what properties of the evolution of the covariance structure facilitated this learning step.

Methods

Experimental model and setup

Animal model

Two adult rhesus monkeys (*Macaca mulatta*) participated in this study (monkey BN, male, 10 kg, 12 years old; and monkey BT, male, 16 kg, 13 years old). All animal housing, care, and experiments were conducted in accordance with the German and European laws governing animal welfare, and in agreement with the US Guidelines for the Care and Use of Mammals in Neuroscience and Behavioral Research (National Research Council, 2003), and the UK Guidelines for Non-human Primate Accommodation, Care and Use (NC3Rs, 2017).

Experimental setup

We built an experimental setup in which monkeys could grasp a virtual rotating handle presented in a 3D graphics environment using a controllable virtual arm avatar. The virtual arm avatar was either controlled by the native arm or a BCI. During native arm control, the real arm of the monkey was tracked in real-time with a full arm tracking system and information from the tracker was used to fully control the arm avatar (see below). During BCI control, online neural or pre-recorded kinematic activity was used to drive the virtual arm. The native arm was occluded with a covering dark plate such that the animal could only receive visual feedback from the virtual display. The virtual arm and handle used a physics engine to detect virtual collisions (MuJoCo, DeepMind, UK). The virtual handle was carefully matched with an equivalent in size, and rotating, real handle such that the monkey received grasping feedback when using the native arm. Both the real and virtual handles rotated in synchrony and could be set in any orientation from 0° to 180°. We defined 0° to be the leftward and 90° to be the upright orientation of the handle (Fig. 2A). The real handle had sensors attached to detect either the power or precision grips. Equivalent physics engine sensors (pressure, crossing, and location detectors at different hand positions) were used to detect correct grips in the virtual environment. The task control programs were implemented using custom LabVIEW code (National Instruments, USA).

Task paradigm

Animals were trained to sit in a primate chair and perform a self-paced delayed grasping task while looking at the virtual reality display. The monkeys could initiate a trial by placing their grasping hand (right hand for both

animals) onto a rest sensor that enabled a red fixation dot on the middle of the virtual handle. After a random period (200 to 400 ms), the handle rotated from 90° to a random orientation (40°,90°,120°,150°,180°), and shortly after, a green color cue was shown next to the handle indicating either a power (left cue) or a precision (right cue) grip. After the grip type cue was turned off, monkeys had to withhold execution for a variable period (200 to 600ms) until the red cue disappeared. The monkeys were then allowed to execute the grasp. A grasp was considered correct when the grip was held for 500ms. Not completing the grasp within 2000ms aborted the trial. All correctly executed trials were rewarded with a liquid reward after the hold period ended. Animals could then return to the hand rest position and initiate a new trial. Error trials were aborted and immediately followed by an error cue and error period. Only correct trials were used in the analyses. The non-grasping hand was gently restrained in a long acrylic tube. In the native arm task, we limited the handle orientations to 40°, 90°, and 150°.

BCI task

The BCI task was identical to the native arm task except for the control of the virtual arm, which was exerted by the computer, the animal (through neuronal activity), or a mixture of both. The native arm was unrestrained during the BCI task. However, lifting the native hand out of the resting sensor at any point during the trial invalidated it. We employed two levels of difficulty in the BCI task. In the "partial collisions" mode, only the fingertips and sensors on the handle that detected the grip type were enabled as colliding elements in the physics engine. In the "full collisions" mode, the full hand and handle were enabled as colliding elements.

Data collection and decoding

Native arm tracking

Previous to the recordings, monkeys were extensively trained in the native arm task while wearing an instrumented glove and observing their own movements in the virtual environment. Finger, hand, and arm kinematics of the acting hand were tracked with eight magnetic sensor coils (Wave tracker, Northern Digital, Canada) placed onto the fingernails, the hand's dorsum, and the wrist to compute the positions of the arm joints in real-time. This enabled tracking the positions of the 18 main arm joints in 3D space including fingers, wrist, elbow, and shoulder[68,59]. We developed a real-time inverse kinematics model to convert those in a 32 joint angles signal. The 32 joint angles were distributed on the shoulder (3), elbow (1), wrist (3), and fingers (25, 5 for each finger: flexion, spread, and rotation at the knuckle, and flexion of the proximal and distal interdigital joint). We obtained the joint angular velocity as the time derivative of each joint angle. The virtual arm model was built from exact measurements of each subject's arm and hand dimensions, and was driven by an equal number of degrees of freedom of joints (32 virtual servos).

Neural recording

After successful training of the native arm task, the animals were implanted with a titanium head holder attached to their skulls, and, in a subsequent surgery, with permanent floating microelectrode arrays (FMA, Microprobe, USA). We targeted three key cortical areas of the grasping circuit: anterior intraparietal area (AIP), ventral premotor area F5, and the hand area of primary motor cortex (M1). In monkey BT we recorded from 6 individual 32 channel arrays implanted into AIP, F5, and M1. Monkey BN was implanted with 8 individual 32 channel arrays but only six arrays were used (F5 and M1, Fig. 1C). On each array, the lengths of the electrodes increased towards the sulcus and ranged from 1.5 (1st row) to 7.1 mm (4th row). Further surgical procedure details have been described in previous experiments[68].

The neural signal was sampled at full bandwidth with a frequency of 24 kHz, a resolution of 16 bits, and simultaneously sorted online and saved to disk (RZ2 BioAmp Processor, Tucker Davis Technologies, USA). For online spike sorting, the signal was hardware filtered (bandpass 0.3-7 kHz) and spikes were detected using the RZ2 default software. Sorting was performed daily and manually on each channel, taking the previous

recording session as a template. Inactive channels were identified by comparing their peri-event time distributions at the beginning of the session to a uniform distribution (Kolmogorov–Smirnov test) and then discarded. No further selection criteria were applied to select the channels. All active units and multi-units were used for online decoding and offline analysis.

The number of recording sessions analyzed varied for each experimental task. For the analysis of the native arm task, we used data from 6 sessions of each animal. Each native arm recording session consisted of 481 ± 103 trials (mean ± SD) and 194 ± 2 single/multi-units for monkey BN, and 221 ± 85 trials and 189 ± 3 single/multi-units for monkey BT. The number of BCI sessions and unit counts varied for each BCI experiment with sessions ranging from 100 to 250 trials. The number of units and multi-units varied with counts ranging from 150 to 180 for monkey BN, and 130 to 160 for monkey BT. Primary motor, premotor, and parietal areas are all potential targets for motor prostheses[69,7,70,37,59,38]. All units and multi-units were analyzed and decoded together, irrespective of brain area.

Kinematic space and kinematic latent space

In analogy to the "neural state space", we refer to the space of possible combinations of the 28 hand joint angles (including wrist and fingers) as the hand's "kinematic state space". All possible hand configurations exist within this space. However, hand anatomy and joint limits restrict this space to a kinematic subspace within the space (or "workspace" in robotics terms). We assumed that a set of synergies or "latent variables" within this kinematic subspace can describe a proportion of possible configurations of the hand. Correspondingly, we refer to space spanned by the latent variables as the "kinematic latent state space" or simply the "kinematic latent space". In the context of neural activity we refer analogously to the "neural latent space" as the neural subspace that aligns best to the latent variables. Although we limit this study to three latent variables, our approach is not limited to these, and other synergies can be used. Latent variables were non-linear on the kinematic state space and were normalized to ranges between -1.0 and 1.0.

Grip aperture was a positive-only axis where position 0.0 indicates the hand is open (flat) and position 1.0 indicates that a grip is completed (either a power or a precision grip). The *wrist rotation* axis was aligned to the handle orientations creating an equal distribution for each grip target from the 40° precision to the 180° power. To compensate for the difference in wrist orientation between the power and precision grip, there was an offset of ca. 50° between the grips (e.g. wrist angle for precision 90° corresponding to power 40°). In the *grip selection* axis, -1.0 shaped the hand to a precision grip and 1.0 to a power grip. Position (0,0,0) in these three axes was the neutral hand rest position. To encourage the subjects to prepare the hand configuration in time, we designed the latent space in such a way that curved trajectories were required to successfully prepare the hand before encountering the handle, which explains the bent trajectories in Figure 6. Not following the trajectories would result in incorrect hand preparations and potential collisions with the handle. To match the native task visually (reach-to-grasp movement) we automated the reach movement and synchronized its progression with the grip aperture axis according to hand/arm psychophysics[71].

Movement within the three variable latent space was mapped to movement in the 28 joint state space through a heuristic dimensionality expansion (DimExp) layer built from the full native arm dataset. The native arm dataset consisted of 10 different trajectories (5 power and 5 precision grips) in a 28 dimensional space. Each trajectory represented a state change from fully open hand to a target grip type at one of the 5 target angles. Trajectory times were matched so that all trajectories had the same number of time points (T). Using the 5 power trajectories, we built a power grip state space by smooth interpolation between time matched points in the trajectories. This resulted in a T×K×28 tensor with T the number of time points, and K the number of interpolation steps between the left-most to the right-most angle (ca. 300). We used extrapolation at the extremes of the state space and hence preventing the 40° and 180° angles were easier to the subject by saturating to the extremes. To convert from latent variables to the full power grip state space, we linearly

mapped the T index to the grip aperture axis and the K index to the wrist rotation angle. In this form, the 28 dimensional vector in the point (T, K) corresponded to the dimensionality expanded state. We followed the same procedure to construct the precision grip state space and latent space. To bind the two spaces, we aligned the two state spaces by angle (40° precision corresponding to roughly 120° power) and interpolated between the two spaces with L the interpolation steps between grip types (ca. 60). This resulted in a T×K×L×28 tensor and (T, K, L) were linearly mapped to our three latent variables.

BCI decoding

Spike counts and behavioral data were streamed from the RZ2 and a behavior control PC to a decoding PC using the UDP protocol. Custom LabVIEW code was used to store the online signal and run the decoders. MATLAB (Mathworks, USA) code was used to train the decoders. The decoder was fed online with spike counts binned at 20ms without further processing. The main online and offline decoder was a position-velocity Kalman filter previously described[59]. The filter follows exactly the implementation of Wu et al.[60]. The filter had 3 to 7 state variables (1-3 positions**,** 1-3 velocities, and 1 offset term) depending on the number of degrees of freedom of the training phase. Only the position state output of the decoder was used to drive the BCI. For the collision experiment, we implemented both innovations of the ReFIT protocol[25], but found a substantial decrease in performance with the second innovation, and therefore used only the first innovation.

To complement the Kalman filter, a non-linear autoregressive with exogenous inputs (NARX) recurrent network model was used (function "narxnet", MATLAB Deep Learning Toolbox). The NARX model has two layers, and receives feedback with the output of the network instead of its internal state in contrast to typical RNNs[53]. Layer 1 was a tanh layer with 20 units and layer 2 a linear layer with 1-3 units (depending on the training phase). The network received as input the position output of the Kalman filter (1-3 states) and its own output (1-3 states). The network was trained using Bayesian regularization, and the Real-time Recurrent Learning algorithm (RTRL), an alternative to back-propagation through time that has gained recent interest due to its advantages in online settings[72]. The network was trained offline only once for each phase, and using an example Kalman filter session as input and the target trajectories as output. We compared the combined KF and NARX decoder to a FORCE RNN decoder, which we implemented exactly as in[46] using the publicly available code of the publication. Output of the NARX network was given to the DimExp layer as described above.

BCI training

Monkeys were trained to use the BCI through an assisted procedure. A brain control (BC) level variable was adjusted by the trainer from 0% to 100%, with 0% being fully assisted control (pre-recorded trajectories), and 100% full brain control. Intermediate values would mix proportionally pre-recorded trajectories and subject control. To calibrate the BCI, we followed a three-stage procedure to first initialize and then refine the decoder. At the start of each session, the monkey first observed the virtual arm perform 10 automated trials of each grasping condition (Observation or Stage 0). The model arm followed pre-defined latent trajectories and grasped the handle error-free. We used the neural data during this stage and the corresponding intended trajectories to train an initial decoder. This decoder was used on a new stage (Initial or Stage 1) at variable BC levels (depending on the progress of the animal). The subject used the BCI until completing 10 correct trials of each grasping condition and a new decoder was trained using fit-to-trajectory (or fit-to-target on the collision experiments). The new decoder was used in Stage 2. The NARX decoder could be alternatively enabled on Stage 2. In our experience, more than 10 grips per condition or more than 3 stages did not significantly improve decoder performance. Importantly, all BCI data reported corresponds to BC 100% except on results showing learning curves.

To calibrate the decoder using fit-to-trajectory, the output position and velocity of the previous stage are replaced with the attempted positions and velocities projected onto the intended trajectory. At hold points, positions are replaced by the intended target position and velocity is set to zero. We calculated the projection

(see Fig. 2 and Fig. S4 for a rational) as follows: (a) Divide each attempted and intended trajectory into decoder time-steps of equal time (20ms in our case) resulting in N^{att} and N^{int} segments. (b) Determine the segment length of the attempted onto intended trajectory as:

$$L^{aoi} = \frac{N^{int}}{N^{att}}.$$

(c) Let the attempted and intended trajectory be defined as n-dimensional parametric curves $f^{att}(t)$ for parameter $t = \{0, 1,..., N^{att}\}$ and $f^{int}(t)$ for parameter $t = \{0, 1,..., N^{int}\}$. Here, each value of t matches a trajectory time step. (d) The projection of attempted onto intended trajectory is found by finding the values on $f^{int}(t)$ corresponding to parameters $t = \{0, L^{aoi}, 2 \times L^{aoi},..., N^{att} \times L^{aoi}\}$.

The values obtained at points $t = \{0, L^{aoi}, 2 \times L^{aoi},..., N^{att} \times L^{aoi}\}$ can be used to produce the position p_i^{aoi}, and velocity vectors v_i^{aoi} of the attempted onto intended trajectory. After this, the Stage 1 neural activity, p_i^{aoi}, and v_i^{aoi} are used to train the Stage 2 decoder. Note that this transform preserves the time taken in the attempted trajectory and hence matches the number of neural activity timesteps. We provide an example Matlab code with a simple implementation of this method in the supplementary section (Fig. S5).

Experiment procedures and data analysis

Decoder comparisons

To compare online decoders (Fig. 3 and Fig. 4), we paired decoding blocks of similar number of trials from each decoder type from the same recording day. The sequence of decoding blocks was chosen randomly each day. On the fit-to-trajectory and target comparisons, each decoder was trained independently following our 3 stages approach. Unless noted, a decoding block usually consisted of ca. 100 trials. On the fit type comparison, the same decoder was used in the three blocks: partial collisions (100 trials), full collisions (100 trials), and washout (20 trials).

Processing and comparison of neural and kinematic signals

To compute peri-event time histograms (PETH) in single unit and joint angle comparisons both spike rates and kinematics were split in 20ms bins, averaged within conditions, and smoothed with a Gaussian kernel (50ms SD). Before averaging, we employed time-warping[73] by aligning to trial events. For single trial unit and joint angle comparisons, we used regression through KF or least-squares. We used the coefficient of determination (R^2) or Pearson correlation coefficients (CC) in its standard form to determine the quality of predictions or correlations. For offline analyses, we preferred the KF as a predictor because of its robustness as a model-based estimator, and our own experience[59]. In general, the KF performed better than least-squares alone in our datasets. All offline regression analyses used cross-validation using the initial portion of the recording (first 10 trials of each condition) to train a decoder and predict the rest of the session.

RNN models

We used recurrent neural network (RNN) models of grasp areas to test generalization of our approach[34]. In our simulations we used single module RNNs with 20 units, a recttanh non-linearity, and α=1e-5, β=1e-2 as regularization parameters. RNNs were fed with a latent neural signal **u** and trained to produce the position **x** and velocity **v** kinematic latents. Signal **u** was produced by multiplying the normalized target latent position by a random NxM matrix with M the number of latent kinematics and N=1.5*M. We simulated 5 different settings in 2D, 6D, and 12D using artificially generated trials (Fig. 5, B: 90, D: 90, F: 90, H: 180, and I: 270 trials) with time dynamics similar to the real data (including 500ms hold periods). In the 2D simulations, we used artificial

training trajectories similar to the trajectories on our training paradigm. In the 6D and 12D simulations, training trajectories were generated as curves within a hypersphere using natural cubic splines. For each setting, we generated 50 additional non-trained conditions using splines and 500ms hold periods. We trained 10 randomly initialized RNNs for each setting and used all trials to compare performance (50 total). For each RNN we trained a new decoder using 50% of the data and predicted the other 50%. Decoders were trained with the internal activity of the RNN (activations) and the target latent positions and velocities using fit-to-trajectory and NARX as in the online case. RNN activations were injected with Gaussian noise of 20% of the signal amplitude. Chance levels were estimated by retraining decoders with neural data shifted by a random number of bins.

Encoding model

To test the capacity of the Kalman filter to recover position and velocity information from binned signals, we encoded the kinematic variables of the BCI task as spike rates. To do this, the intended position and velocity signals of an example session from each animal (2 positions, 2 velocities, 20ms samples) were multiplied by scaling factors (α, β), Z-scored, injected with Gaussian noise (20% of the signal amplitude), and offset as positive only values. The resulting signal was then multiplied by an Nx4 normally distributed random matrix (N=130) and given to a Poisson filter to produce spike-like counts. A portion of the resulting artificial spike train (10%) was used to train a Kalman predictor, and the resulting decoder was used to predict the kinematic variables of the remaining session. Decoding accuracy of the model data was measured as in the previous cases by comparing the output of the decoder to the intended kinematics. We ran each model parameter combination 10 times with different noise and random matrices and reported the average and the 95% confidence interval of each output (correlation coefficient or total spikes).

Author contributions

AAT, JAM, WS, and HS designed research; AAT, and JAM designed the algorithm; AAT, WS, and HS performed research; AAT, and JAM analyzed data; and AAT wrote the paper; AAT, JAM, WS, HS reviewed the paper.

Acknowledgments

We thank N. Bobb, S. Borchert, M. Dörge, A. Filippow, and K. Menz for technical and theoretical help. We thank F. R. Willett, and J. Goodman for helpful comments on an earlier version of this manuscript. This work was supported by Deutsche Forschungsgemeinschaft Grants FOR-1847 B3, SFB-889 C09, and EU Horizon 2020 B-CRATOS GA 965044.

References

1. Fetz, E.E. (1969). Operant Conditioning of Cortical Unit Activity. Science *163*, 955–958. 10.1126/science.163.3870.955.
2. Moxon, K.A., and Foffani, G. (2015). Brain-Machine Interfaces beyond Neuroprosthetics. Neuron *86*, 55–67. 10.1016/j.neuron.2015.03.036.
3. Golub, M.D., Chase, S.M., Batista, A.P., and Yu, B.M. (2016). Brain–computer interfaces for dissecting cognitive processes underlying sensorimotor control. Current Opinion in Neurobiology *37*, 53–58. 10.1016/j.conb.2015.12.005.

4. Serruya, M.D., Hatsopoulos, N.G., Paninski, L., Fellows, M.R., and Donoghue, J.P. (2002). Instant neural control of a movement signal. Nature *416*, 141–142. 10.1038/416141a.
5. Taylor, D.M., Tillery, S.I.H., and Schwartz, A.B. (2002). Direct Cortical Control of 3D Neuroprosthetic Devices. Science *296*, 1829–1832. 10.1126/science.1070291.
6. Carmena, J.M., Lebedev, M.A., Crist, R.E., O'Doherty, J.E., Santucci, D.M., Dimitrov, D.F., Patil, P.G., Henriquez, C.S., and Nicolelis, M.A.L. (2003). Learning to Control a Brain–Machine Interface for Reaching and Grasping by Primates. PLoS Biol *1*, e42. 10.1371/journal.pbio.0000042.
7. Musallam, S., Corneil, B.D., Greger, B., Scherberger, H., and Andersen, R.A. (2004). Cognitive control signals for neural prosthetics. Science *305*, 258–262. 10.1126/science.1097938.
8. 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*, 1098–1101. 10.1038/nature06996.
9. Nuyujukian, P., Kao, J.C., Ryu, S.I., and Shenoy, K.V. (2017). A Nonhuman Primate Brain–Computer Typing Interface. Proc. IEEE *105*, 66–72. 10.1109/JPROC.2016.2586967.
10. Hochberg, L.R., Bacher, D., Jarosiewicz, B., Masse, N.Y., Simeral, J.D., Vogel, J., Haddadin, S., Liu, J., Cash, S.S., van der Smagt, P., et al. (2012). Reach and grasp by people with tetraplegia using a neurally controlled robotic arm. Nature *485*, 372–375. 10.1038/nature11076.
11. 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. The Lancet *381*, 557–564.
12. Bouton, C.E., Shaikhouni, A., Annetta, N.V., Bockbrader, M.A., Friedenberg, D.A., Nielson, D.M., Sharma, G., Sederberg, P.B., Glenn, B.C., Mysiw, W.J., et al. (2016). Restoring cortical control of functional movement in a human with quadriplegia. Nature *533*, 247–250. 10.1038/nature17435.
13. Ajiboye, A.B., Willett, F.R., Young, D.R., Memberg, W.D., Murphy, B.A., Miller, J.P., Walter, B.L., Sweet, J.A., Hoyen, H.A., Keith, M.W., et al. (2017). Restoration of reaching and grasping movements through brain-controlled muscle stimulation in a person with tetraplegia: a proof-of-concept demonstration. The Lancet *389*, 1821–1830. 10.1016/S0140-6736(17)30601-3.
14. 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*, e18554. 10.7554/eLife.18554.
15. Flesher, S.N., Downey, J.E., Weiss, J.M., Hughes, C.L., Herrera, A.J., Tyler-Kabara, E.C., Boninger, M.L., Collinger, J.L., and Gaunt, R.A. (2021). A brain-computer interface that evokes tactile sensations improves robotic arm control. Science *372*, 831–836. 10.1126/science.abd0380.
16. Anderson, K. (2004). Targeting Recovery: Priorities of the Spinal Cord-Injured Population. Journal of neurotrauma *21*, 1371–1383. 10.1089/neu.2004.21.1371.
17. Cutkosky, M.R. (1989). On grasp choice, grasp models, and the design of hands for manufacturing tasks. IEEE Transactions on robotics and automation *5*, 269–279.
18. Feix, T., Romero, J., Schmiedmayer, H.-B., Dollar, A.M., and Kragic, D. (2016). The GRASP Taxonomy of Human Grasp Types. IEEE Transactions on Human-Machine Systems *46*, 66–77. 10.1109/THMS.2015.2470657.
19. 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. 10.1088/1741-2560/12/1/016011.
20. Nason, S.R., Mender, M.J., Vaskov, A.K., Willsey, M.S., Ganesh Kumar, N., Kung, T.A., Patil, P.G., and Chestek, C.A. (2021). Real-time linear prediction of simultaneous and independent movements of two finger groups using an intracortical brain-machine interface. Neuron. 10.1016/j.neuron.2021.08.009.
21. Gao, P., and Ganguli, S. (2015). On simplicity and complexity in the brave new world of large-scale neuroscience. Current Opinion in Neurobiology *32*, 148–155. 10.1016/j.conb.2015.04.003.
22. Piazza, C., Grioli, G., Catalano, M.G., and Bicchi, A. (2019). A Century of Robotic Hands. Annu. Rev. Control Robot. Auton. Syst. *2*, 1–32. 10.1146/annurev-control-060117-105003.
23. Mathis, A., Schneider, S., Lauer, J., and Mathis, M.W. (2020). A Primer on Motion Capture with Deep Learning: Principles, Pitfalls, and Perspectives. Neuron *108*, 44–65. 10.1016/j.neuron.2020.09.017.
24. Adolph, K.E., and Joh, A.S. (2007). Motor development: How infants get into the act. Introduction to infant development *2*, 63–80.

25. Gilja, V., Nuyujukian, P., Chestek, C.A., Cunningham, J.P., Yu, B.M., Fan, J.M., Churchland, M.M., Kaufman, M.T., Kao, J.C., Ryu, S.I., et al. (2012). A high-performance neural prosthesis enabled by control algorithm design. Nature Neuroscience *15*, 1752–1757. 10.1038/nn.3265.
26. Orsborn, A.L., Dangi, S., Moorman, H.G., and Carmena, J.M. (2012). Closed-Loop Decoder Adaptation on Intermediate Time-Scales Facilitates Rapid BMI Performance Improvements Independent of Decoder Initialization Conditions. IEEE Trans. Neural Syst. Rehabil. Eng. *20*, 468–477. 10.1109/TNSRE.2012.2185066.
27. Shanechi, M.M., Orsborn, A.L., Moorman, H.G., Gowda, S., Dangi, S., and Carmena, J.M. (2017). Rapid control and feedback rates enhance neuroprosthetic control. Nature Communications *8*, 13825. 10.1038/ncomms13825.
28. Willett, F.R., Murphy, B.A., Young, D., Memberg, W.D., Blabe, C.H., Pandarinath, C., Franco, B., Saab, J., Walter, B.L., Sweet, J.A., et al. (2018). A Comparison of Intention Estimation Methods for Decoder Calibration in Intracortical Brain-Computer Interfaces. IEEE Trans Biomed Eng *65*, 2066–2078. 10.1109/TBME.2017.2783358.
29. Ganguly, K., and Carmena, J.M. (2009). Emergence of a Stable Cortical Map for Neuroprosthetic Control. PLOS Biology *7*, e1000153. 10.1371/journal.pbio.1000153.
30. 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*, 423–426. 10.1038/nature13665.
31. 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. Nature Neuroscience. 10.1038/s41593-018-0095-3.
32. Oby, E.R., Golub, M.D., Hennig, J.A., Degenhart, A.D., Tyler-Kabara, E.C., Yu, B.M., Chase, S.M., and Batista, A.P. (2019). New neural activity patterns emerge with long-term learning. PNAS, 201820296. 10.1073/pnas.1820296116.
33. Gallego, J.A., Perich, M.G., Chowdhury, R.H., Solla, S.A., and Miller, L.E. (2020). Long-term stability of cortical population dynamics underlying consistent behavior. Nature Neuroscience *23*, 260–270. 10.1038/s41593-019-0555-4.
34. Michaels, J.A., Schaffelhofer, S., Agudelo-Toro, A., and Scherberger, H. (2020). A goal-driven modular neural network predicts parietofrontal neural dynamics during grasping. Proc Natl Acad Sci USA, 202005087. 10.1073/pnas.2005087117.
35. 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*, 9659–9669. 10.1523/JNEUROSCI.5443-09.2010.
36. Bansal, A.K., Truccolo, W., Vargas-Irwin, C.E., and Donoghue, J.P. (2012). Decoding 3D reach and grasp from hybrid signals in motor and premotor cortices: spikes, multiunit activity, and local field potentials. Journal of Neurophysiology *107*, 1337–1355. 10.1152/jn.00781.2011.
37. Aggarwal, V., Mollazadeh, M., Davidson, A.G., Schieber, M.H., and Thakor, N.V. (2013). State-based decoding of hand and finger kinematics using neuronal ensemble and LFP activity during dexterous reach-to-grasp movements. Journal of Neurophysiology *109*, 3067–3081. 10.1152/jn.01038.2011.
38. Intveld, R.W., Dann, B., Michaels, J.A., and Scherberger, H. (2018). Neural coding of intended and executed grasp force in macaque areas AIP, F5, and M1. Scientific Reports *8*, 17985. 10.1038/s41598-018-35488-z.
39. Goodman, J.M., Tabot, G.A., Lee, A.S., Suresh, A.K., Rajan, A.T., Hatsopoulos, N.G., and Bensmaia, S. (2019). Postural Representations of the Hand in the Primate Sensorimotor Cortex. Neuron *104*, 1000-1009.e7. 10.1016/j.neuron.2019.09.004.
40. Rastogi, A., Willett, F.R., Abreu, J., Crowder, D.C., Murphy, B.A., Memberg, W.D., Vargas-Irwin, C.E., Miller, J.P., Sweet, J., Walter, B.L., et al. (2021). The Neural Representation of Force across Grasp Types in Motor Cortex of Humans with Tetraplegia. eNeuro. 10.1523/ENEURO.0231-20.2020.
41. Abbaspourazad, H., Choudhury, M., Wong, Y.T., Pesaran, B., and Shanechi, M.M. (2021). Multiscale low-dimensional motor cortical state dynamics predict naturalistic reach-and-grasp behavior. Nat Commun *12*, 607. 10.1038/s41467-020-20197-x.
42. Stavisky, S.D., Kao, J.C., Nuyujukian, P., Pandarinath, C., Blabe, C., Ryu, S.I., Hochberg, L.R., Henderson, J.M., and Shenoy, K.V. (2018). Brain-machine interface cursor position only weakly affects monkey and human motor cortical activity in the absence of arm movements. Sci Rep *8*, 16357. 10.1038/s41598-018-34711-1.
43. Okorokova, E.V., Goodman, J.M., Hatsopoulos, N.G., and Bensmaia, S.J. (2020). Decoding hand kinematics from population responses in sensorimotor cortex during grasping. Journal of neural engineering *17*,

046035.
44. Kingma, D.P., and Welling, M. (2014). Auto-Encoding Variational Bayes. arXiv:1312.6114 [cs, stat].
45. Pandarinath, C., O'Shea, D.J., Collins, J., Jozefowicz, R., Stavisky, S.D., Kao, J.C., Trautmann, E.M., Kaufman, M.T., Ryu, S.I., Hochberg, L.R., et al. (2018). Inferring single-trial neural population dynamics using sequential auto-encoders. Nat Methods *15*, 805–815. 10.1038/s41592-018-0109-9.
46. Sussillo, D., Nuyujukian, P., Fan, J.M., Kao, J.C., Stavisky, S.D., Ryu, S., and Shenoy, K. (2012). A recurrent neural network for closed-loop intracortical brain–machine interface decoders. Journal of Neural Engineering *9*, 026027. 10.1088/1741-2560/9/2/026027.
47. Golub, M.D., Yu, B.M., Schwartz, A.B., and Chase, S.M. (2014). Motor cortical control of movement speed with implications for brain-machine interface control. Journal of Neurophysiology *112*, 411–429. 10.1152/jn.00391.2013.
48. Sadtler, P.T., Ryu, S.I., Tyler-Kabara, E.C., Yu, B.M., and Batista, A.P. (2015). Brain–computer interface control along instructed paths. J. Neural Eng. *12*, 016015. 10.1088/1741-2560/12/1/016015.
49. Inoue, Y., Mao, H., Suway, S.B., Orellana, J., and Schwartz, A.B. (2018). Decoding arm speed during reaching. Nat Commun *9*, 1–14. 10.1038/s41467-018-07647-3.
50. Churchland, M.M., and Shenoy, K.V. (2007). Temporal Complexity and Heterogeneity of Single-Neuron Activity in Premotor and Motor Cortex. Journal of Neurophysiology *97*, 4235–4257. 10.1152/jn.00095.2007.
51. Rasmussen, R.G., Schwartz, A., and Chase, S.M. (2017). Dynamic range adaptation in primary motor cortical populations. eLife *6*, e21409. 10.7554/eLife.21409.
52. Naufel, S., Glaser, J.I., Koerding, K.P., Perreault, E.J., and Miller, L.E. (2018). A muscle-activity-dependent gain between motor cortex and emg. Journal of Neurophysiology. 10.1152/jn.00329.2018.
53. Diaconescu, E. (2008). The use of NARX Neural Networks to predict Chaotic Time Series. *3*, 10.
54. Martens, J., and Sutskever, I. (2011). Learning recurrent neural networks with hessian-free optimization. In Proceedings of the 28th International Conference on International Conference on Machine Learning ICML'11. (Omnipress), pp. 1033–1040.
55. Santello, M., Flanders, M., and Soechting, J.F. (1998). Postural Hand Synergies for Tool Use. J. Neurosci. *18*, 10105–10115. 10.1523/JNEUROSCI.18-23-10105.1998.
56. Gallego, J.A., Perich, M.G., Miller, L.E., and Solla, S.A. (2017). Neural Manifolds for the Control of Movement. Neuron *94*, 978–984. 10.1016/j.neuron.2017.05.025.
57. Schaffelhofer, S., and Scherberger, H. (2016). Object vision to hand action in macaque parietal, premotor, and motor cortices. eLife *5*, e15278. 10.7554/eLife.15278.
58. Willett, F.R., Avansino, D.T., Hochberg, L.R., Henderson, J.M., and Shenoy, K.V. (2021). High-performance brain-to-text communication via handwriting. Nature *593*, 249–254. 10.1038/s41586-021-03506-2.
59. Menz, V.K., Schaffelhofer, S., and Scherberger, H. (2015). Representation of continuous hand and arm movements in macaque areas M1, F5, and AIP: a comparative decoding study. J. Neural Eng. *12*, 056016. 10.1088/1741-2560/12/5/056016.
60. Wu, W., Black, M., Gao, Y., and Bienenstock, E. (2002). Inferring hand motion from multi-cell recordings in motor cortex using a Kalman filter. In.
61. Wang, W., Chan, S.S., Heldman, D.A., and Moran, D.W. (2007). Motor Cortical Representation of Position and Velocity During Reaching. Journal of Neurophysiology *97*, 4258–4270. 10.1152/jn.01180.2006.
62. Kalaska, J.F. (2009). From Intention to Action: Motor Cortex and the Control of Reaching Movements. In Progress in Motor Control Advances in Experimental Medicine and Biology., D. Sternad, ed. (Springer US), pp. 139–178. 10.1007/978-0-387-77064-2_8.
63. Omrani, M., Kaufman, M.T., Hatsopoulos, N.G., and Cheney, P.D. (2017). Perspectives on classical controversies about the motor cortex. Journal of Neurophysiology *118*, 1828–1848. 10.1152/jn.00795.2016.
64. Branco, M.P., de Boer, L.M., Ramsey, N.F., and Vansteensel, M.J. (2019). Encoding of kinetic and kinematic movement parameters in the sensorimotor cortex: A Brain-Computer Interface perspective. European Journal of Neuroscience *50*, 2755–2772. 10.1111/ejn.14342.
65. Georgopoulos, A.P., Caminiti, R., and Kalaska, J.F. (1984). Static spatial effects in motor cortex and area 5: Quantitative relations in a two-dimensional space. Exp Brain Res *54*. 10.1007/BF00235470.
66. Ashe, J., and Georgopoulos, A.P. (1994). Movement Parameters and Neural Activity in Motor Cortex and

Area 5. Cereb Cortex *4*, 590–600. 10.1093/cercor/4.6.590.
67. 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*, 2760–2780. 10.1523/JNEUROSCI.3147-06.2007.
68. Schaffelhofer, S., Agudelo-Toro, A., and Scherberger, H. (2015). Decoding a Wide Range of Hand Configurations from Macaque Motor, Premotor, and Parietal Cortices. J. Neurosci. *35*, 1068–1081. 10.1523/JNEUROSCI.3594-14.2015.
69. Fu, Q.G., Flament, D., Coltz, J.D., and Ebner, T.J. (1995). Temporal encoding of movement kinematics in the discharge of primate primary motor and premotor neurons. Journal of Neurophysiology *73*, 836–854. 10.1152/jn.1995.73.2.836.
70. Churchland, M.M., Cunningham, J.P., Kaufman, M.T., Foster, J.D., Nuyujukian, P., Ryu, S.I., and Shenoy, K.V. (2012). Neural population dynamics during reaching. Nature. 10.1038/nature11129.
71. Jeannerod, M. (1984). The Timing of Natural Prehension Movements. Journal of Motor Behavior *16*, 235–254. 10.1080/00222895.1984.10735319.
72. Marschall, O., Cho, K., and Savin, C. (2020). A Unified Framework of Online Learning Algorithms for Training Recurrent Neural Networks. Journal of Machine Learning Research *21*, 1–34.
73. Williams, A.H., Poole, B., Maheswaranathan, N., Dhawale, A.K., Fisher, T., Wilson, C.D., Brann, D.H., Trautmann, E.M., Ryu, S., Shusterman, R., et al. (2019). Discovering Precise Temporal Patterns in Large-Scale Neural Recordings through Robust and Interpretable Time Warping. Neuron *0*. 10.1016/j.neuron.2019.10.020.

Supplementary Figures

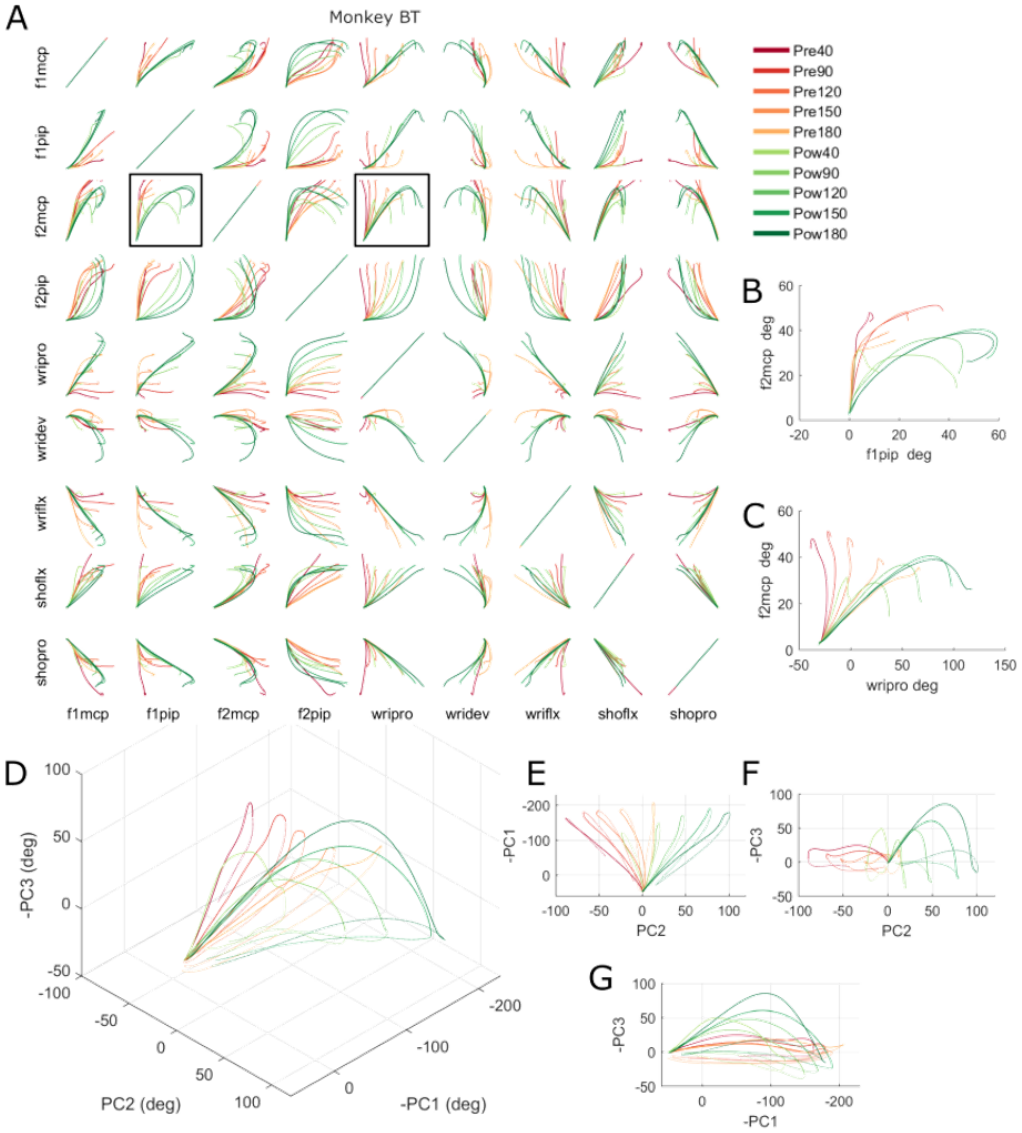

Figure S1

Grip execution describes curved trajectories in the kinematic state space. (A) Example projections of the trajectories in joint angle space during precision and power grips. Each sub panel compares the evolution of a joint angle (out of 32 of the arm) from the neutral hand rest position (where the 10 traces converge) to the handle hold period (divergent traces end). During execution of the grasp, joint angle pairs tend to describe curved trajectories (reflecting time differences in the evolution of each individual angle). Curvatures can be observed comparing arm and hand joint angle values or finger joint pairs. (B-C) Zoom-in into the example pairs highlighted in dark boxes in (A). Axes are in degrees. (A-C) Pow: power grips, Pre: precision grips, f1: thumb, f2: index, mcp: metacarpophalangeal joint, pip: proximal interphalangeal joint, wri: wrist, sho: shoulder, flx: flexion, pro: pronation. (D) Three first principal components from all joint angles (32 angles, >90% of the total variance). Solid line is the trajectory from hand rest to hold and dotted line is the return to rest phase of the movement. In general, movement in the kinematic space can be described as a curved trajectory. At hand rest, monkeys were usually in a neutral, relaxed position. During displacement to the handle (grip preparation) the whole hand was usually extended in preparation for the grip, and then the grip was executed. Upon return, monkeys usually "dragged" back the hand in a relaxed manner. The lesser curvature during the return phase indicates that most of the curvature is due to hand preparation which is not done when the animal returns the hand to rest. (E-G) Upper and side views of (D). Data are trial averages from one entire recording session from monkey BT executing two grip types and all handle orientations.

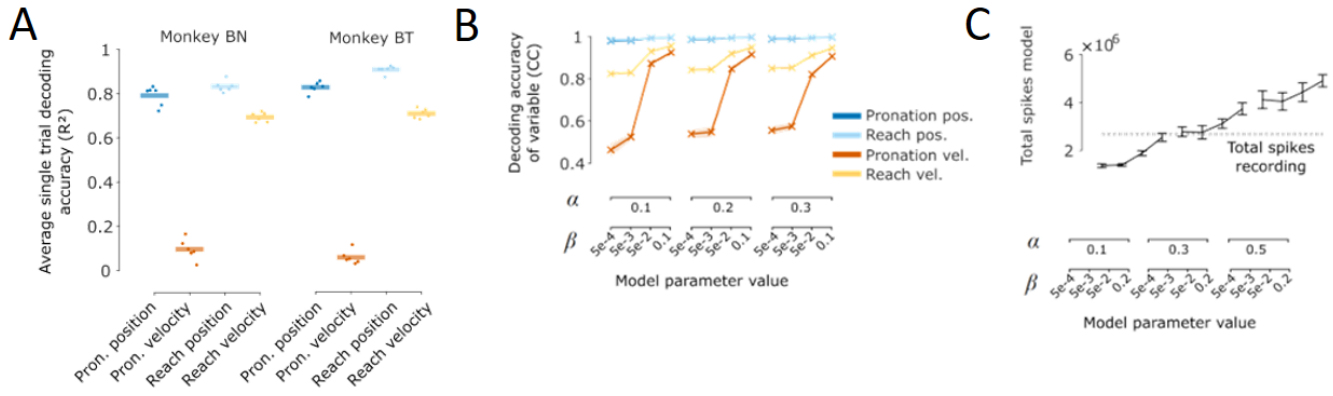

Figure S2

Neural latent space evolved as kinematic latent variables. (A) Coefficients of determination of decoded single-trials to trained target trajectories for six late-stage sessions of Phase (ii). Same data as Fig. 7A but now using R^2 as a metric. Horizontal bars are the averages of all sessions. (B,C) Same as Fig. 7E,F but for monkey BT. (B) Correlation of decoded single-trials to trained target trajectories for different values of α and β using the encoding model on one BCI recording of monkey BT (D). For each point in the plot, we performed 10 simulations and plotted the average. Shades: 2×STD. (C) Average total amount of spikes for each point of (B). The dashed line is the total spike count of the original BCI recording. Error bar: 2×STD.

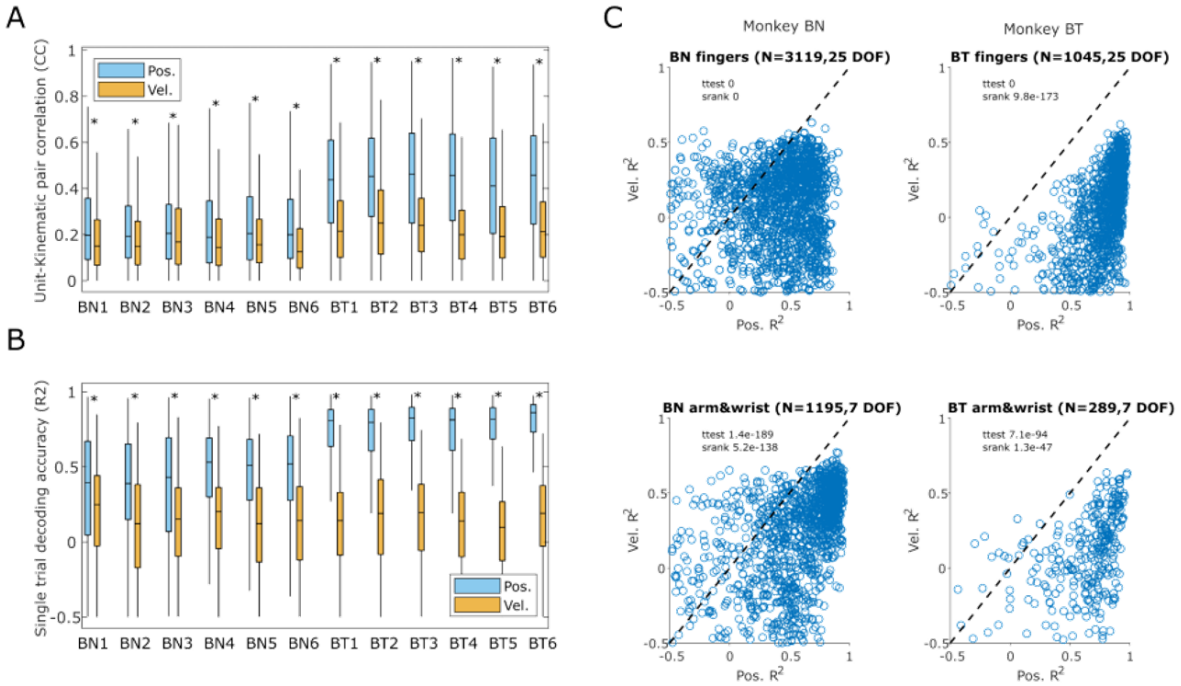

Figure S3

The amount of position information is also proportionally higher during native grasping. (A) Same analysis as in Fig. 8F with denoised PETHs for six native arm sessions. Bars show distributions of the pairwise correlation (absolute value) of all the available kinematics (position and velocities) to each unit. Distributions are significantly different ($p<0.001$, t-test and Wilcoxon signed-rank test). (B) Same analysis as in Fig. 8G for six native arm sessions. Bars compare decoding accuracy of position vs. velocity kinematics on single trials and single kinematics for the six native arm sessions. Position and velocity kinematics were independently predicted on each trial from neural activity using Kalman filters, and compared to average kinematics to produce R^2 values. R^2 values below -0.5 were discarded. Distributions are significantly different ($p<0.001$, t-test and Wilcoxon signed-rank test). (A, B) Horizontal axis identifies animal and session numbers. The box plot displays the median (horizontal black line), 25th and 75th percentiles (box), and range of the distribution (vertical black line). (C) Same as Fig. 8G but separating arm and hand joints. Comparison of decoding accuracy of position vs. velocity kinematics on single trials and single kinematics for the finger (top, 25 joint angles), and arm joints (bottom, 7 joint angles including wrist joint). Data corresponds to a single session. Position and velocity kinematics were independently predicted on each trial from neural activity using Kalman filters, and compared to average kinematics to produce R^2 values. R^2 values below -0.5 were discarded.

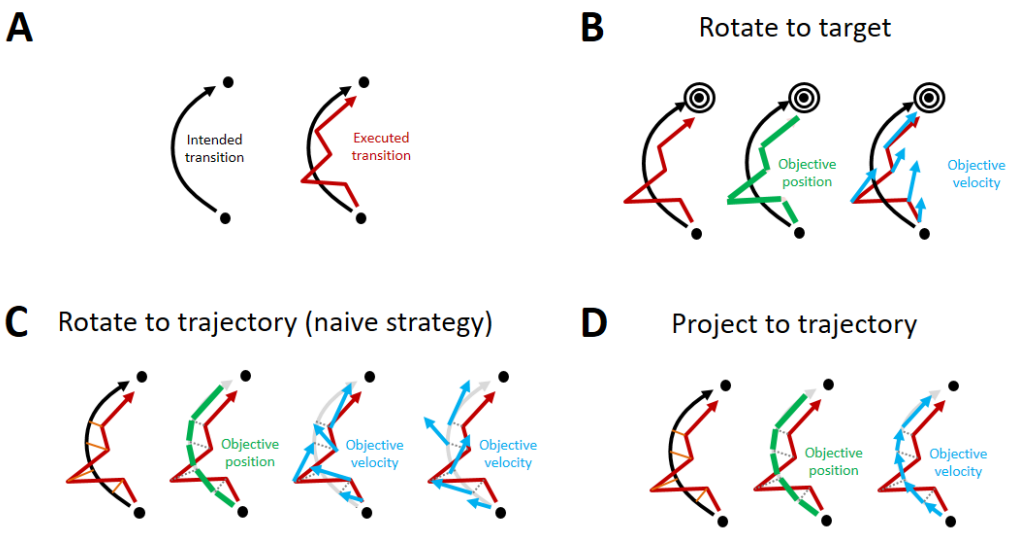

Figure S4

Potential intention-estimation strategies. (A) We assume there is an intended trajectory (ideal hand configuration transition) and an executed trajectory (obtained from data from an initial decoding attempt) in some kinematic space. (B) The ReFIT strategy[25] assumes that the intention is to reach a specific target in the kinematic space. To recalibrate the BCI, an objective position and an objective velocity are needed to train the new decoder. In ReFIT, it is assumed the subject's intent was to move towards the target and the velocity vectors are redirected to the target. The objective position is left unchanged in ReFIT as this has no influence in the decoder (second innovation). When trajectories are relevant, we opted for projecting the attempted trajectory onto the intended trajectory to preserve some of the velocity intent (note that this might imply some re-scaling). The objective position is clear in this case as this will be along the intended trajectory. There are two possible strategies to reorient the velocity vectors: (C) towards the trajectory, or (D) along the trajectory. Note however that rotating towards the trajectory implies that the intended velocities shoot out of the trajectory when the position is considered (C). We opted for strategy (D).

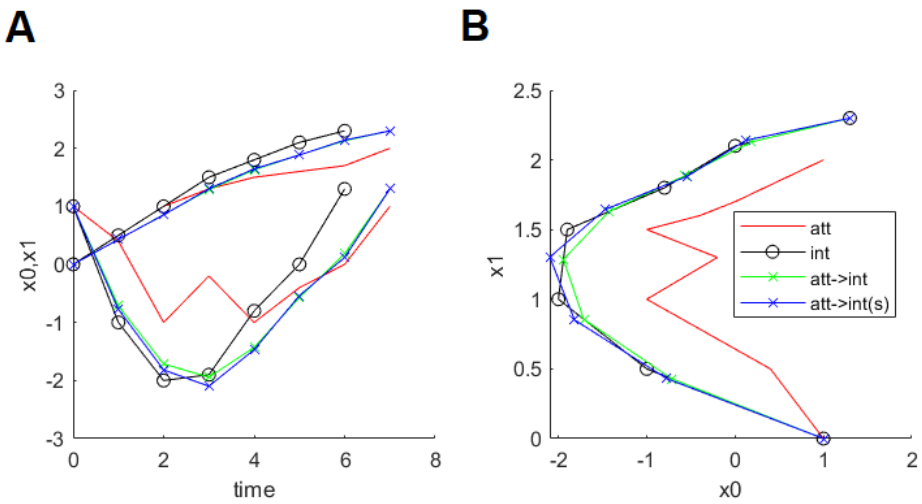

Figure S5

Example output of the Matlab example code for the fit-to-trajectory method. (A) Example latent variables x0 and x1 over time. (B) Latent space with variables plot against each other. Legend: att: attempted trajectory, int: intended trajectory, att->int: attempted onto intended trajectory with a linear interpolation method (Matlab function interp1), att->int(s): attempted onto intended trajectory with a spline interpolation method (Matlab function interp1).

Example code of the fit-to-trajectory method

The output of the Matlab code can be seen in Figure S5.

```
clearvars;
% Example trajectories
patt = [1 0; 0.4 0.5; -1 1; -0.2 1.3; -1 1.5; -0.4 1.6; 0 1.7; 1 2];
pint = [1 0; -1 0.5; -2 1; -1.9 1.5; -0.8 1.8; 0 2.1; 1.3 2.3];
Natt = size(patt,1)-1;
Nint = size(pint,1)-1;

% Prepare parameters
Laoi = Nint/Natt;
Tatt = 0:Natt;
Tint = 0:Nint;
Taoi = 0:Laoi:round(Natt*Laoi);

% Find parameters on the curve
paoi = interp1(Tint, pint, Taoi);
paois = interp1(Tint, pint, Taoi, 'spline');

% Plot
figure(1); clf;
subplot(1,2,1); hold on;
plot(Tatt,patt,'r'); plot(Tint,pint,'-ok');
plot(Tatt,paoi,'-xg'); plot(Tatt,paois,'-xb');
xlabel('time'); ylabel('x0,x1');
subplot(1,2,2); hold on;
plot(patt(:,1),patt(:,2),'r'); plot(pint(:,1),pint(:,2),'-ok');
plot(paoi(:,1),paoi(:,2),'-xg'); plot(paois(:,1),paois(:,2),'-xb');
xlabel('x0'); ylabel('x1');
legend({'att','int','att->int','att->int(s)'},'Location','e');
```