Intracortical brain-computer interface for navigation in virtual reality in macaque monkeys

Abstract

We present an innovative intracortical Brain-Computer Interface (BCI) to bridge the gap between laboratory settings and real-world applications. This BCI approach introduces three key advancements. First, we utilized neural signals from three macaque brain regions – primary motor, dorsal and ventral premotor cortex – enabling precise and flexible decoding of real-time 3D sphere/avatar velocities. Second, we developed a realistic, immersive three-dimensional virtual reality setup with dynamic camera tracking, allowing continuous navigation and obstacle avoidance that closely mimic real-world scenarios. Finally, our BCI approach is optimized for use by paralyzed patients, featuring a brief training phase without overt movements and closed-loop operation without retraining of the decoder, relying on the user's neural plasticity and the decoder's robust generalization across tasks. Our BCI adapted to different environments, targets, and obstacles, illustrating its potential to significantly enhance the quality of life for paralyzed patients by enabling natural, reliable and flexible control in complex settings.

Introduction

Brain-Computer Interfaces (BCI) are a promising technology that enable direct communication between the brain and external devices by translating neural activity into specific commands. BCIs can be used to enhance or restore functions lost due to disease or trauma, bypassing damaged biological pathways[1]. Intracortical BCIs (iBCIs) in nonhuman primates and humans have enabled control of computer cursors[2–9], manipulation of robotic or prosthetic arms[3–8,10–15], restoration of communication through typing[16–19], restoring motion of paralyzed limbs[15,20,21], and even the provision of sensory feedback[22].

Previous motor BCI studies have often focused on cursor or robotic arm control due to the limitations of a lab environment, which hamper the translation to real-world situations such as wheelchair navigation. Furthermore, realistic brain-controlled real-world navigation is frequently confronted with unpredictable events requiring highly flexible online corrections that are likely to depend on both premotor and primary motor cortex activity. To bridge the gap between lab and home environments, we developed a BCI approach in a three-dimensional virtual reality (VR) environment with stereoscopic vision for realistic simulations of increasing complexity and unpredictability. To maximize the potential of flexible control of the cortical motor system, we employed for the first time simultaneous recordings in dorsal and ventral premotor and primary motor cortex. While previous iBCI research was limited to simple tasks in basic settings (e.g., a sphere moving freely in a featureless space)[23,24], our approach offers a more complex and realistic VR environment, with 3D movements that closely mimic real-life scenarios. Additionally, we implemented a realistic navigation task that effectively mimicked real-life wheelchair control due to dynamic camera tracking and a realistic 3D VR environment.

Our iBCI successfully translated neural activity from three motor cortical areas into real time velocities in three monkeys, demonstrating the potential of our BCI system for real-world brain-controlled navigation.

Results

We implanted three rhesus monkeys with Utah arrays in three motor regions: primary motor cortex (M1), dorsal premotor cortex (PMd) and ventral premotor cortex (PMv). We designed five tasks to validate the effectiveness of our BCI (Table 1). In all tasks, the decoder was trained with data obtained during passive observation of the VR environment without any overt movements.

Task	Monkey	Chance level [%]	Success rate [%] (mean ± std)	Number of channels (mean ± std)	Number of sessions
Center-out task	Monkey 1	23	73 ± 4	140 ± 6	5
	Monkey 2	20	83 ± 8	49 ± 3	10
	Monkey 3	23	64 ± 13	217 ± 11	19
Navigation task	Monkey 1	32	69 ± 6	139 ± 6	6
	Monkey 2	28	70 ± 15	47 ± 5	7
	Monkey 3	32	59 ± 8	216 ± 10	15
3D Center-out task	Monkey 2	17	70 ± 7	43 ± 5	12
	Monkey 3	19	58 ± 8	202 ± 14	10
Respawn task	Monkey 2	23	76 ± 9	39 ± 4	11
	Monkey 3	26	82 ± 5	199 ± 15	9
Obstacle task	Monkey 2	18	63 ± 6	40 ± 4	10
	Monkey 3	23	70 ± 10	203 ± 17	11

Table 1|Estimated chance level, success rate (mean + standard deviation), number of channels used for decoding (mean ± standard deviation) and the number of sessions for each task and for each monkey. For Monkey 1, the average percentages of electrodes used for decoding in M1, PMv and PMd were 46%, 34% and 20% respectively. For Monkey 2, only electrodes in M1 were used. For Monkey 3, the percentages of electrodes used for decoding in M1, PMv and PMd were 36%, 41% and 23% respectively. The smaller percentage in PMd in monkeys 1 and 3 was because only 64 electrodes were available for recording, compared to 96 for M1 and PMv.

Success rates

To assess how well macaque monkeys could perform BCI-control tasks in a moderately complex 3D environment in VR, we focused on the Center-out and Continuous Navigation tasks (Supplementary Videos 1 and 2). All three monkeys performed significantly above chance level from the start of the recording period (Table 1, permutation tests: all $p < 0.0001$), with individual sessions achieving success rates as high as 96% (Fig. 1A, Fig. 1B and Extended Data Fig. 1). Performance in the Continuous Navigation task was slightly lower (4 – 13% fewer successful trials) compared to the Center-out task, most likely due to the dynamic camera tracking in the former task. Monkeys 1 and 2 had participated in a previous BCI project and reached success rates around 70% from the start of the recording period in both tasks. In contrast, Monkey 3 required more sessions (N=11 for the Center-out task and N = 14 for the Continuous Navigation task) to achieve a similar success rate, demonstrating a clear learning trend across sessions in the Center-out task (the first task learned; linear regression: $\beta=0.0156$, $F(1,17)=12.98$, $p=0.0022$, $R^2=0.433$). To verify that navigation was also possible from a first-person perspective, we removed the monkey avatar in the Continuous Navigation task in 3 sessions. Monkey 3 achieved similar success rates (average success rate of 55%) as in the standard Continuous Navigation task, confirming the adaptability of the BCI across perspectives. Thus, macaque monkeys are able to achieve accurate 2D cursor control and navigation in a 3D VR environment, even with dynamic changes in camera viewpoint during navigation, as well as from a first-person perspective.

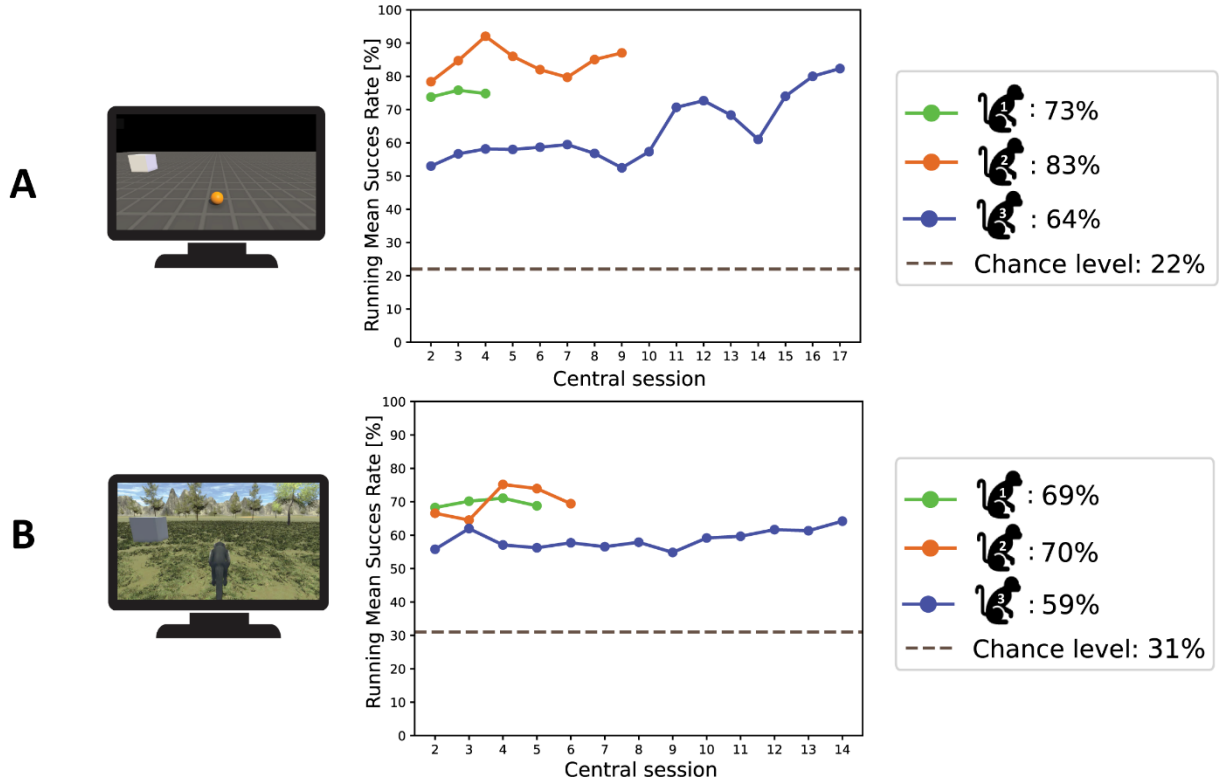

Fig. 1|Success rate over time for the main tasks. A Running mean of the success rate for the different sessions (window size = 3 and sliding window = 1), together with the estimated chance level and the average success rate of all sessions. Running mean success rate of Monkey 1, 2 and 3 for the Center-out task. **B** Same data for the Continuous Navigation task.

To investigate to what extent our animals could achieve flexible and accurate BCI control in more challenging VR environments, we designed three additional tasks. In the 3D Center-out task (Fig. 2A, Supplementary Video 3), in which the animal was required to move the sphere in three dimensions, the overall success rate averaged 70% across all sessions for Monkey 2 and 58% for Monkey 3. Although performance was lower than in the 2D Center-out task, it remained significantly above chance level from the start of the recording period. Surprisingly, both monkeys generalized effortlessly to the third dimension (z-direction) without the need of explicit training of the monkey.

We also observed markedly flexible and accurate BCI control in the Respawn task (Fig. 2B, Supplementary Video 4). Without the need for explicit decoder training or additional animal

training, both monkeys tested achieved around 90% success when the target object unpredictably shifted position during the trial with no significant difference in performance between respawn trials and no-respawn trials (Mann-Whitney U = 185629.5, p = 0.1556 for Monkey 2 and U = 101694.5, p = 0.0765 for Monkey 3).

Both monkeys also learned to achieve reliable obstacle avoidance in the Obstacle task (Supplementary Video 5). Initially, the success rate of Monkey 2 was lower in trials with an obstacle on the path (36%) compared to off-path trials (66%, Fig. 2C), but at the end of the recording period, the success rate of this animal in on-path trials reached 55% (which was still significantly lower than in off-path trials, U = 142032.5, p = $7.071e^{-16}$). In contrast, Monkey 3 performed markedly better in on-path trials, achieving an average success rate of 63%, closer to its off-path performance (73%, U = 132242.5, p = $6.534e^{-4}$, Supplementary Material). Overall, the monkeys successfully completed the various tasks with a high success rate and without the need for extensive training, even when navigating 3D trajectories, encountering obstacles, or adjusting to changes in target position during the trials.

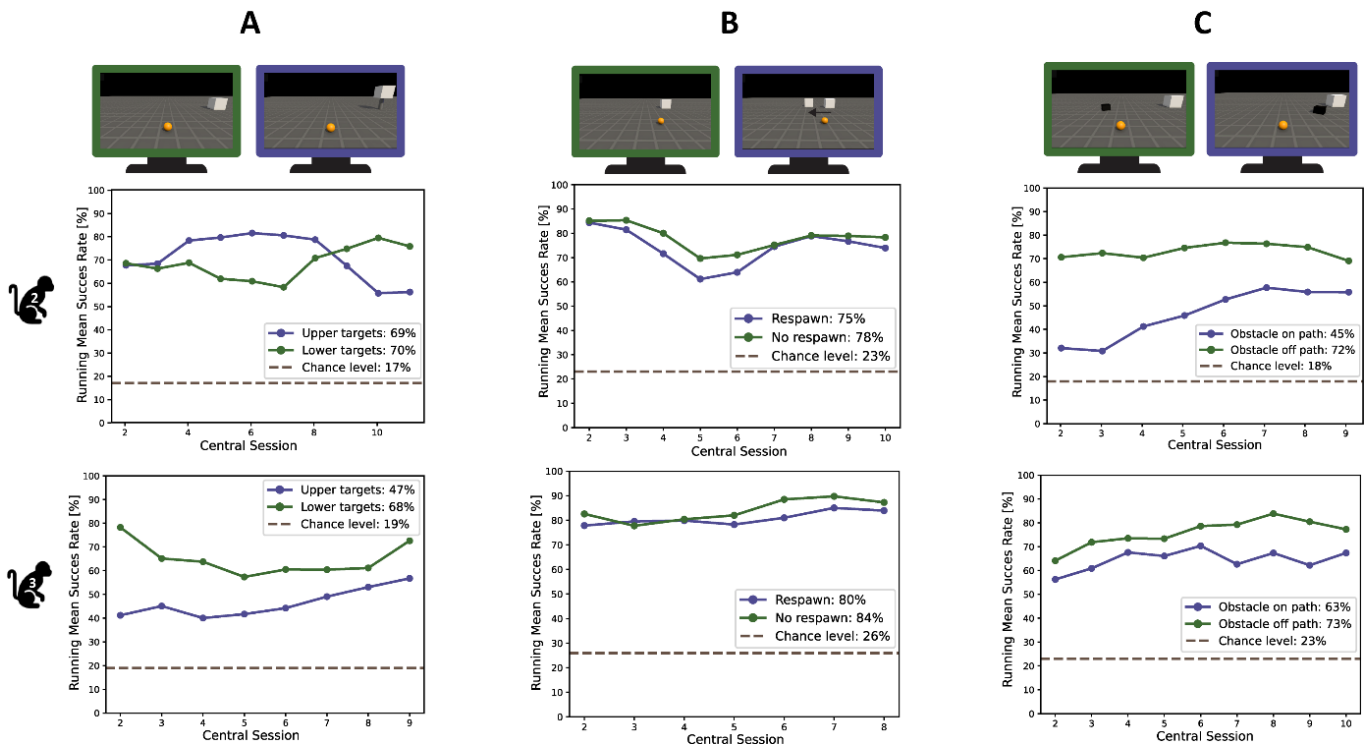

Fig. 2|Flexibility of BCI control. A Running mean success rate (window size = 3 and sliding window = 1) of Monkey 2 and 3: upper targets (blue) and lower targets (green) for the 3D Center-out task. The success rates for reaching the upper and lower were nearly identical (69% and 70%, respectively) and well above chance level for Monkey 2. Performance in this task was lower for Monkey 3, but the monkey still performed well above chance level for both targets. **B** Same data for the respawn task: respawn trials (blue) and trials without respawning (green). **C** Same data for trials with obstacle on path (blue) and obstacle off path (green) for the Obstacle task.

To determine the contributions of the different cortical areas to the decoding, we identified the most influential electrodes by calculating the summed weights on the latent variables for each electrode and applying a 50% cumulative weight threshold. In Monkey 1, PMv (38% of the electrodes used for decoding) and PMd (41%) contributed similarly, and remarkably much more than M1 (22%), while in Monkey 3, PMv (61%) contributed significantly more to the decoding than both M1 (19%) and PMd (20%). Similar results were obtained when using a 90% cumulative weight threshold. Note that in monkey 2 – who performed worse on the Obstacle Task – all electrodes used in the decoder were from M1 since the other arrays had no electrodes with spiking activity.

Trajectories and time to target

We investigated the sphere's trajectories during BCI control in the additional tasks. Fig. 3A shows that in both monkeys, the final y-coordinate of the sphere differed significantly between upper and lower target trials (Mann-Whitney U = 196746, $p = 8.422e^{-58}$ for Monkey 2; U = 76945, $p = 1.062e^{-34}$ for Monkey 3).

Next, we investigated how fast Monkeys 2 and 3 corrected the trajectories during respawning by calculating the reaction time, defined as the time required for the sphere to change its trajectory from the original target to the respawn target, starting from the moment the target was respawned (i.e. changed location). The average reaction time across all conditions was 661ms for Monkey 2 and 1058ms for Monkey 3 (Fig. 3B and C, Kruskal-Wallis test between conditions, Monkey 2: statistic = 91.84, $p = 5.18e^{-17}$; Monkey 3: statistic = 41.77, $p = 5.759e^{-7}$).

Finally, we explored the monkey's strategies for obstacle avoidance by plotting the trajectory of each individual trial per target for each monkey (Fig. 3D). Monkey 2 utilized both sides to navigate around the obstacles and made deviations from the straight trajectory closer to the target, whereas Monkey 3 tended to favor one side and often made a larger deviation early in the trajectory.

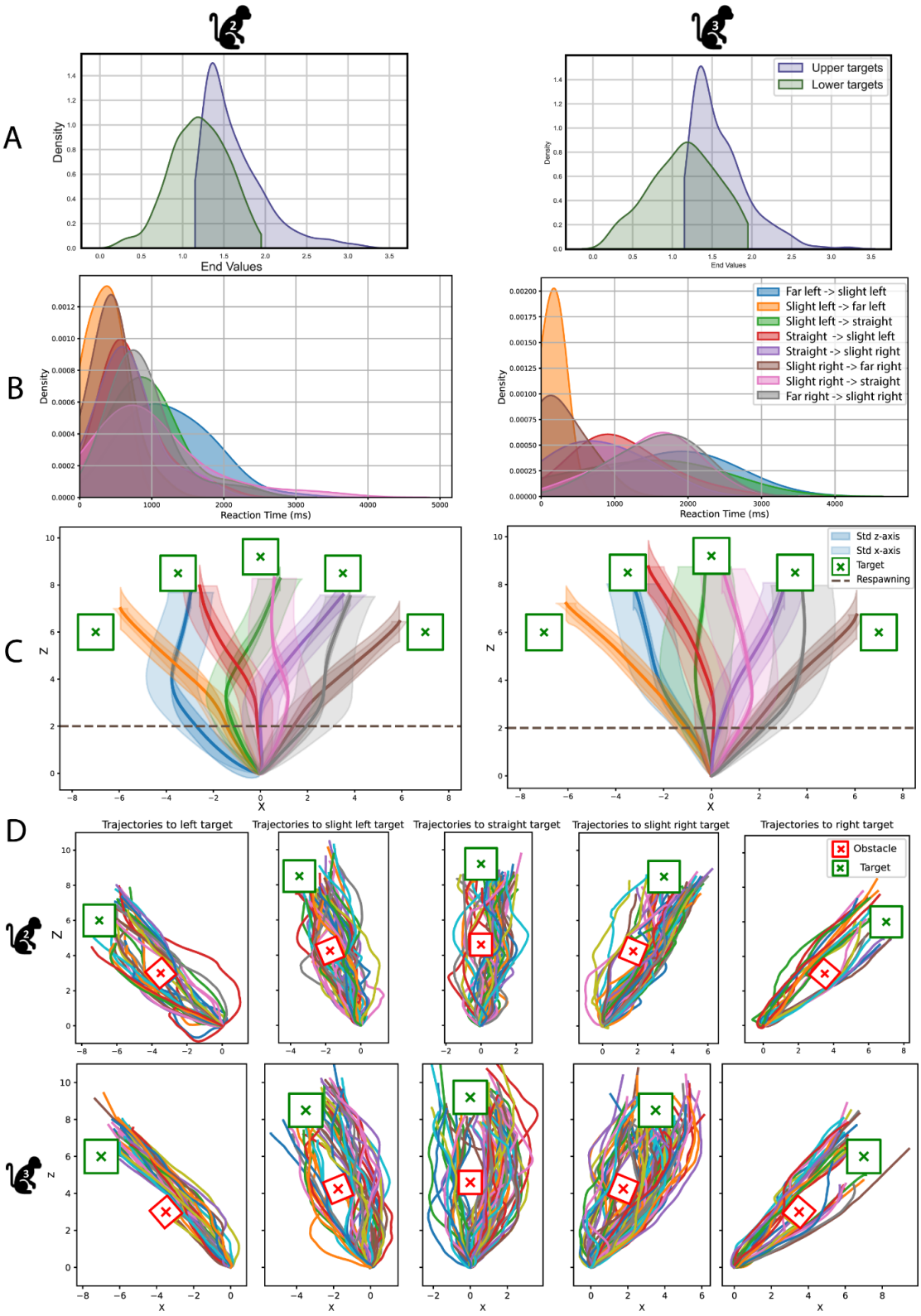

Fig. 3|Trajectories analysis of the three supplementary tasks for Monkeys 2 and 3. A 3D Center-out task: Density distribution of the y-coordinate of the endpoint values of trials with upper (blue) and lower (green) targets (across all sessions). **B-C** and Reaction time and average trajectory per respawn condition for the Respawn task. **B** Density distribution of average

reaction time per respawn condition over all trials and sessions. **C** Average trajectory for each respawn condition, together with the standard deviation in the x- and z-directions across all trials and all sessions. The respawn moment is also shown. **D** Individual trajectories of successful trial across all sessions for the Obstacle task, for each target (green square) when the obstacle (red square) was on the target's path.

In the Center-out task, the average time to target in the Training phase (controlled by Unity) was 2750ms, compared to 3645ms (Monkey 1), 3629ms (Monkey 2) and 3552ms (Monkey 3) in the Online Decoding phase (controlled by the monkeys). This result indicates that the time to target in the Online Decoding phase was on average 31% slower (across all monkeys) compared to the Training phase (Extended Data Fig. 2A). In the Continuous Navigation task, the average time to target was 63% slower during the Online Decoding phase (Extended Data Fig.2B), which highlights the increased difficulty of this task. For the time to target in the three additional tasks, see Extended Data Fig.2C-E.

Influence of surface electromyography (sEMG) and eye movements on decoded velocities

We wanted to assess the influence of sEMG on the decoded velocities (v, v_x, v_y, v_z). The mean correlation coefficients across sessions between sEMG and decoded velocities were consistently very low across tasks and velocity components, ranging from -0.08 to 0.06 for Monkey 1 and -0.20 to 0.13 for Monkey 2 (Extended Data Table 1). Cross-correlation analysis for lags ranging from -100 to 100ms (50ms increments) yielded similar results. Regression analyses further supported these findings, with coefficients of determination (R^2) values near zero for both the linear and non-linear models (Extended Data Table 2). Although the combined p-values from the regression analyses were statistically significant, the small effect sizes suggest that sEMG had minimal influence on decoded velocities. Furthermore, the high mean

184 squared error (MSE) values for both monkeys across all tasks and velocity components indicate
185 poor predictive performance of the models (Extended Data Table 2).
186 We also computed the Spearman correlation between the x and y components of the eye
187 position and the decoded velocity. Similar to the results for the sEMG analysis, we found very
188 small correlation coefficients for all velocity components and for each component of the eye
189 movement, except for a notable correlation between the x component of the eye position and v_x
190 (Extended Data Table 3), which is related to the monkeys tracking the sphere's position during
191 online decoding, as in natural behavior[25]. However, such a correlation was not observed during
192 the Continuous navigation task, where the camera dynamically followed the avatar. In this task,
193 the decoding is world-centered rather than eye-centered (decoding velocities to the left remain
194 left even if the camera has moved and the avatar has to go straight ahead), altering the
195 relationship between the eye position and the decoded velocity. These findings suggest that eye
196 movements did not fundamentally influence the decoded velocity.
197

198 ## Discussion

199

200 We have developed a novel method to achieve robust and highly flexible BCI
201 performance in a 3D virtual reality environment, based on the decoding of neural activity from
202 M1, PMd and PMv into real-time sphere/avatar velocities. Our BCI performance, defined as
203 the entire system including Utah arrays, decoder and VR environment, achieved real-time
204 continuous navigation in a 3D environment with dynamic camera tracking, closely mimicking
205 real-life navigation.
206 Our BCI approach represents a significant innovation in BCI research due to the fully
207 brain-controlled calibration process, minimal training time, remarkable generalization
208 capabilities and its capacity for rapid and accurate online corrections. Unlike traditional

methods that rely on physical limb movements[2,11,13,23,26] or incremental training approaches[10–12,17], our interface enabled direct neural control from the beginning of the Online Decoding phase. The Training Phase was notably brief (approximately 7 minutes), after which the monkeys were immediately capable of performing tasks successfully. Additionally, our decoder demonstrated robust generalization across environments (grid or forest) and moving entities (sphere or avatar). Notably, in the continuous Navigation task, we trained the decoder using a sphere on a grid, while during the Online Decoding phase, the monkey controlled an avatar navigating in a complex forest environment. Despite the substantial changes in both the moving entity and the environment, the animals performed the tasks with a high success rate. Furthermore, our decoder's capability extended to decoding movements in both 2D and 3D and also allowed to tackle new challenges not included in the Training phase. We demonstrated that our BCI could successfully generalize from 2D to 3D sphere control in the Center-out task without prior training of the monkey, as indicated by the robust BCI performance from the start of the experiment. Our monkeys were also able to adapt seamlessly to changes in the goal location during the Respawn task and were able to manage additional obstacles in the Obstacle task. The closed-loop nature of our system provided real-time visual feedback in 3D that promoted neural plasticity, allowing the monkeys to adapt their neural activity and improve control of the sphere/avatar[27]. These features demonstrate the robustness, versatility and innovation of our approach, making it a powerful tool for advanced iBCI applications.

Our BCI approach contained several innovations that may have contributed to its success. Unlike previous studies that used only M1[3,10,11,16,17,26], PMd [2] or a combination of both [12,13], we recorded simultaneously in M1, PMd and PMv. A parallel study (De Schrijver, Garcia-Ramirez, Decramer and Janssen, unpublished observations), showed that online cursor control based on PMv activity alone can reach a similar performance as when using M1 or PMd activity. The inclusion of PMv in the current study likely enhanced the precision and flexibility of the

BCI, as evidenced by the sizable contribution of PMv electrodes to the computation of the latent states. While it is possible to use solely M1 for velocity decoding, PMv (and possible PMd) may be an important factor in optimizing BCI performance and flexibility. Consistent with this idea, Monkey 2 (with only M1 electrodes) performed poorly in the obstacle avoidance task.

A second novel factor is the genuine 3D virtual reality environment with stereoscopic vision we used. We achieved depth perception by means of liquid crystal shutters operating at 60 Hz and synchronized to the refresh rate of the monitor, a setup in which monkeys achieve excellent performance in a 3D-structure categorization task[28–30] suggesting optimal stereo vision. In addition, the VR environment created in Unity contained many other depth cues (perspective, shading, texture, relative motion) similar to real-world situations. We could not determine the contribution of individual depth cues, but the combination of these cues during online sphere or avatar control undoubtedly contributed to an exceedingly realistic visual feedback to the animals.

Finally, the powerful decoding approach we adopted (based on Preferential Subspace Identification[31]) and the interaction between decoding and neural activity were also important factors. Unlike previous studies[32], we did not retrain the decoder in the course of a recording session. Instead, we allowed the animals to change their spiking activity based on the continuous visual feedback they received[33]. Even in sessions where the initial model fit to the training data was very low ($R^2 < 0.1$, 3-fold cross-validation), the online decoding performance improved markedly as the animal learned to control the sphere or the avatar using only 6 latent variables.

Our iBCI system demonstrates several critical capabilities that make it highly suitable for assisting paralyzed patients in real-life situations. It enables fast, precise control and continuous navigation, which are essential for tasks such as wheelchair control. The system can seamlessly adjust to changes in target locations and successfully perform obstacle avoidance,

all of which are necessary for navigating in complex, cluttered environments such as a house, while also responding quickly to changes in the patient's intentions. Our BCI was also capable of generalizing across different environments, transitioning smoothly from a basic environment to a complex virtual environment without additional training. This adaptability, combined with a short straightforward training phase, enhances its practicality for home use. Additionally, our BCI's performance was unaffected by muscle activity, ensuring reliable control even in patients with residual movement. These features suggest that our iBCI could significantly improve the quality of life for paralyzed patients by enabling them to perform daily tasks with greater independence.

Online methods

Surgery and recording procedures

Three male rhesus monkeys (*Macaca mulatta, 7-9kg*) were implanted with a titanium headpost that was fixed to the skull with titanium screws and dental acrylic. After training the monkey in a fixation task, we implanted three 96-channel Utah arrays with an electrode length of 1 or 1.5mm and an electrode spacing of 400µm (4x4mm, Blackrock Neurotech, UT, USA), guided by stereotactic coordinates and anatomical landmarks. The arrays were inserted with a pneumatic inserter (Blackrock Neurotech, UT, USA) with a pressure of 1.034bar and an implantation depth of 1mm. During all surgical procedures, the monkey was kept under propofol anesthesia (10mg/kg/h) and strict aseptic conditions. Postoperative anatomical scans (Siemens 3T scanner, 0.6mm resolution) verified the positions of the Utah arrays in dorsal premotor area F2, ventral premotor area F5c, and the hand/arm region of the primary motor cortex. All surgical and experimental procedures were approved by the ethical committee on

animal experiments of the KU Leuven and performed according to the *National Institute of Health's Guide for the Care and Use of Laboratory Animals* and the EU Directive 2010/63/EU.

During a recording session, neural data were collected with digital Cereplex M headstages (Blackrock Neurotech, UT, USA) that were connected to two digital neural processors. The data were subsequently sent to the Cerebus data acquisition system that recorded the signal of 256 channels, thus allowing simultaneous recording of the three brain areas (one bank of 32 electrodes in PMd was not recorded). The signal was high pass filtered (750Hz) and sampled at 30 kHz. Each recording day, the thresholds to detect spikes were set manually below the noise to capture the neural activity of individual neurons. Note that one channel could contain the signal of one or multiple neurons.

Experimental setup

The monkey sat in a chair with its head fixed and arms restrained in front of a Viewpixx 3D screen (VPixx Technologies Inc., Saint-Bruno, Canada, 1920x1080 pixels with a refresh rate of 120 Hz) in a dark room (Fig.4). On the screen, pairs of images for left and right eye with slight angular disparities were presented alternately at 120 Hz. The monkey wore shutter glasses with liquid crystal lenses that opened and closed when voltage was applied, which was perfectly synchronized with the screen at a frequency of 60Hz for each eye to achieve stereoscopic vision. A similar setup was used in detailed behavioral and electrophysiological testing of 3D vision based on disparity in macaque monkeys[28–30]. An infrared camera monitored the monkey's eye movements with a sampling rate of 500 Hz (EyeLink 500, SR Research, Ontario, Canada). Prior to each recording session, the eye tracking system was calibrated by requiring the animals to fixate targets presented on the 3D screen.

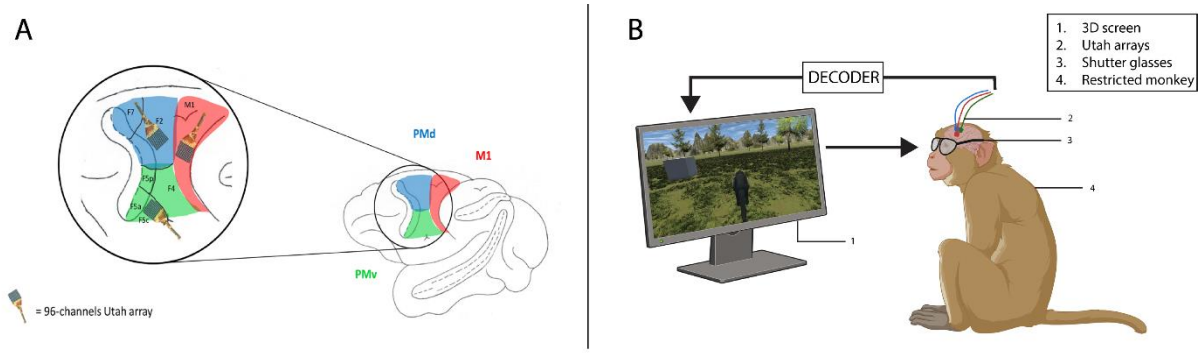

Fig.4|Implantation site and experimental setup. A Implantation site of three 96-channels Utah arrays: one in PMd, PMv and M1. **B** Experimental setup of closed-loop BCI: head-fixed monkey sitting in a chair with restrained arms. A 3D screen in front of the monkey displayed the task. The Monkey wore shutter glasses synchronized with the 3D screen, to allow the perception of depth.

Task description

During each recording session, multiple experiments were performed that each consisted of three distinct phases: a Training phase, a Decoder Training phase, and an Online Decoding phase. The main goal of each experiment was to move an entity (a sphere or a monkey avatar) from a starting point to a target that pseudorandomly appeared in the 3D environment.

Experimental phases.

In the **Training phase**, the monkey passively observed Unity's task execution using its AI-driven navigation system. This phase employed three strategically placed targets—far left, straight ahead, and far right—to cover the main directions in the 3D environment. For each target, we typically collected 30 trials while recording neural activity in three motor areas (M1, PMv and PMd) and the velocities of the moving entity. In the **Decoder Training phase** (typical

duration of a few sec to 1 min)**,** we trained a model using the neural activity and the entity's velocities that were recorded during the Training phase. In the **Online Decoding phase**, the monkey controlled the entity using its decoded neural activity. The tasks developed in Unity used computed velocities from the decoding algorithm as inputs, enabling real-time control of the entity's movement. The online decoding formed a closed-loop Brain-Computer Interface (BCI) as the monkey observed the decoded velocities in real-time on the 3D screen.

Tasks.

The first task was a Center-out task, featuring a sphere in a 3D environment that moved on a two-dimensional (2D) plane towards the target (a white cube, Fig.5A). During the Training phase, the sphere moved from a fixed starting point in a straight line to one of three targets (far left, straight, and far right) that appeared pseudorandomly. The eyes were monitored to ensure fixation on the part of the screen where the sphere moved towards the target. For each target, 30 correct trials were recorded. During the Online Decoding phase, the monkey needed to reach five possible target positions (Extended Data Fig. 3) that appeared pseudorandomly by controlling the sphere movements through the decoding algorithm.

In the Continuous Navigation task, the starting position at the beginning of each trial was the end position of the previous trial. In the Training phase, the monkey observed a sphere performing continuous movements on a 2D grid. The sphere followed a curved trajectory towards the target. The Center-out task and the Continuous Navigation task differed with respect to the camera movement. In the Center-out task, the camera remained fixed, the sphere moved in the z- (depth) and x-direction (width) and the targets remained at a constant distance from the camera, causing the sphere to appear smaller on the screen as it moved toward the targets. In contrast, in the Continuous Navigation task, the camera dynamically tracked the sphere on the grid. As the sphere approached the targets, their size increased on the screen.

Similar to the Center-out task, three targets were used during the Training phase, and 30 correct trials of each target were recorded. While the Training phase was performed with a sphere moving on a plane, the Online Decoding phase used a monkey avatar navigating in a 3D forest environment (Fig.5B). When the avatar rotated, the camera rotated correspondingly, revealing different sections of the 3D forest. During the Online Decoding phase, the monkey was required to navigate the avatar monkey to five different target positions in the forest environment. During this phase, we recorded 100 trials per session, for both main tasks.

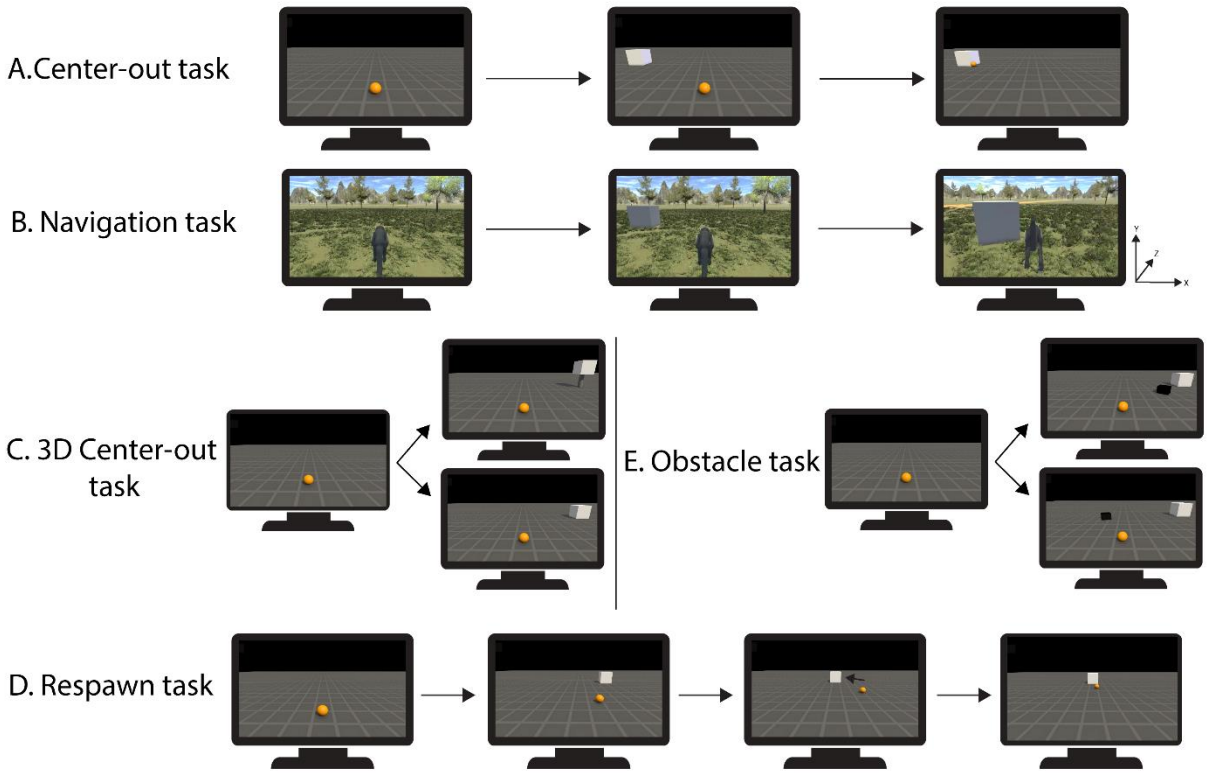

Fig.5|Experimental tasks. **A-B** Two main tasks: temporal sequence of the Center-out task (A) and Continuous Navigation task (B) in xyz-space. **C-D** Three supplementary tasks: 3D Center-out task with two possibilities, upper target and lower target (C). **D** Respawn task in chronological order, with the respawning of the target to a different location during the decoding trial. **E** Obstacle task with two possibilities during the decoding trial, above with obstacle on the path of the target and below with an obstacle off the path of the target

Monkey 2 and 3 performed three additional tasks that were based on the main Center-out task. In the first additional task, the sphere was moved in the x (width), y (height), and z

371 (depth) directions (3D Center-out task). This task used ten different targets, five of which were
372 situated on the plane at identical positions to those in the first Center-out task and the remaining
373 five targets shared the same xz-coordinates but had positive y-coordinates (i.e. located higher
374 than the five targets on the plane, balancing on thin cylinders, Fig.5C). In the Training phase of
375 the 3D Center-out task, six targets were shown; three on the ground and three elevated, and for
376 each target we recorded fifteen correct trials. During the Online Decoding phase, the animal
377 moved the sphere in three dimensions to reach each of the ten possible targets that appeared
378 pseudorandomly.

379 For the last two tasks, the Training phase was identical to the Center-out task. During
380 the Online Decoding phase of the Respawn task, the target could respawn (i.e. disappear and
381 reappear at another location) to one of its adjacent target positions after the sphere crossed a
382 specific point in the space (z-coordinate equal to two, Fig.5D). Thirteen target patterns were
383 possible and chosen pseudorandomly: no respawning of the five target positions, and
384 respawning of each target in two possible directions (to the left or to the right) or in one possible
385 direction for the most left and most right target. In the Obstacle task, an obstacle was added
386 during the Online Decoding phase (Fig.5E). The obstacle appeared pseudorandomly on or off
387 the path to the target. The obstacle was smaller than the target, situated in the middle of the
388 trajectory to the target and oriented in the direction of the specific target, so that the sphere had
389 to deviate from a straight trajectory in on-path trials. During the Online Decoding phase of the
390 three additional tasks, 120 trials were recorded per sessions. The dimensions of all tasks can be
391 found in Extended Data Fig. 3.

392

393 Experimental protocol.

394 In the Center-out tasks, the sphere appeared at the starting position on the xz-plane,
395 while for the Continuous Navigation task, the starting position of the sphere/avatar was the end

position of the last trial. To start a trial, the monkey had to fixate on the sphere or avatar for a duration of 500ms. Subsequently, a target appeared at one of the predetermined positions. The sphere or avatar had to reach the target within a predefined time period, denoted as maximum trial time (5.5s for the Center-out task and the 3D Center-out task, 8s for the Continuous Navigation task and 6.5s for the Respawn and Obstacle tasks). A trial was successful when the sphere or avatar stayed inside a predefined box around the target (target window) for 500ms. Successful trials were rewarded with a liquid reward. An unsuccessful trial occurred if the target was not reached within the maximum trial time or if the sphere, avatar, or target exited the camera's field of view. Trials in which the subject failed to maintain fixation at the beginning of the trial were aborted and not further analyzed. Upon completion of a trial in the Center-out tasks (whether successful, unsuccessful, or aborted), both the sphere and the target disappeared from the screen. If in the Continuous Navigation task the avatar did not reach the target in the predefined time window or if the target went out of the scene, the trial stopped and the avatar remained at its final position, while the target disappeared.

Unity

Software and hardware.

The tasks were developed using Unity 2021.3.16f1 and implemented on standard desktop computers. The Unity's 3D Universal Render Pipeline (URP) template was used as a starting point for developing each task. The subject interacted with the tasks developed in Unity using a custom-built BCI system. Unity was installed on a first computer controlling the 3D screen that the monkey was looking at. On a second computer, the decoding model (in python) was trained to decode velocities in real-time. On a third computer, the central task script (in python) was executed, facilitating the exchange of commands between the three programs.

These commands included basic instructions such as configuring trials, starting and stopping trials, and predictions. A Local Area Network (LAN) facilitated communication between the three programs using ZeroMQ over TCP/IP sockets to enable efficient and reliable data exchange in a distributed system.

Scene composition.

The Center-out task was set up in an environment with a flat plane featuring a grid pattern, providing multiple depth cues in 3D space. The sphere was able to translate in the x and z directions in the Center-out, Respawn and Obstacle tasks and in the x, y, and z directions for the 3D Center-out task. The targets were white static cubes. In the 3D Center-out task, the elevated targets were located on thin cylinders. In the Continuous Navigation task, the environment was a virtual forest in an open field bordered by mountains (Fig.5B). The navigational entity was now represented by a monkey avatar (a capuchin monkey) obtained from the Unity store[34]. The dimensions of each scene are shown in Extended Data Fig. 3.

Movement during training.

The movement of the sphere and avatar during the Training phase of the experiment was controlled by a built-in Unity component, the Nav Mesh Agent. Attached to the sphere/avatar game object, the Nav Mesh Agent operated on a baked Nav Mesh on the navigation plane. The 'SetDestination' method was utilized to update the Nav Mesh Agent's destination, initiating the autonomous movement of the sphere towards the target in a straight trajectory. The motion began with an acceleration, gradually reaching a steady speed of four units per second. As the sphere approached the target position, it decelerated, ensuring that the sphere came to a complete stop just before making contact with the target. For the Continuous Navigation task, the sphere moved along a smooth, curved trajectory towards the target. Our

method leveraged Bezier curves to compute a curved path towards the target. The trajectory path was divided into destination points, and the 'SetDestination' function guided the avatar through these points, resulting in a realistic and more natural movement pattern. The movement during the 3D Center-out task involved the sphere moving on the plane (to the lower targets) and flying from the ground to the elevated targets. The use of the nav mesh agent was not possible due to its restriction to the surface-only baked nav mesh. Instead, the Unity build-in function 'transform.Translate' was used to move the sphere in a straight line, along with the simulation of acceleration and deceleration.

BCI-controlled movement.

The decoding algorithm produced velocities (vx, vy, vz) that dictated the direction and the speed of the sphere/avatar's movement. These velocities were transmitted to Unity every 50ms, forming the basis for real-time updates to the sphere/avatar's position in the virtual environment. The sphere and the avatar were equipped with a rigid body component. The decoding algorithm dynamically adjusted Unity's rigid body velocity, translating navigational intent into realistic sphere and avatar movements. The sphere/avatar were only responsive to the decoding algorithm's output during the interval from target onset to target acquisition. The sphere was limited to translation in the x, y (for the 3D task), and z directions. In the Continuous Navigation task, the avatar could translate in the x and z directions while dynamically rotating based on the velocities. From the x and z velocities, we computed the rotation angle and forward speed. The forward speed (v) was computed as the magnitude of the velocity vector using the formula: $v = \sqrt{v_x^2 + v_z^2}$. The rotation angle (θ) was calculated using the arctangent function: $\theta = \tan^{-1}(\frac{v_x}{v_z})$. Rotation was achieved using the 'Quaternion.Lerp' function applied to the avatar's transform, ensuring a smooth rotation that corresponded to the avatar's actual movement.

Camera view.

A dual-camera setup replicated the visual perspectives of the left and right eyes to create binocular disparities. These perspectives were sequentially displayed on the 3D screen, simulating a stereoscopic or 3D view. The camera projection was configured for a perspective view, and the field of view for each camera was set to 60 degrees. For the Center-out task, the camera positions remained fixed. In contrast, during the Continuous Navigation task, the cameras dynamically tracked the avatar in the 3D environment. The tracking was achieved using the command ‘transform.LookAt()’, enabling the camera to smoothly translate and rotate in sync with the avatar's movements. Throughout the task, it was essential to verify that the sphere (for the Center-out tasks) or the target (for the Continuous Navigation task) remained within the camera's field of view at all times. To achieve this, the Unity function 'GeometryUtility.TestPlanesAABB(camera_frustum, collider.bounds)' was employed. This function assessed whether the collider bounds of the object (sphere or target) remained within the 3D space visible to the camera (camera fustrum). The camera frustum was computed with the Unity function ‘GeometryUtility.CalculateFrustumPlanes’. This evaluation was conducted for both the left and right eye camera views.

Reaching the target.

The target was reached by the sphere/avatar when it stayed within a predefined box around the target (target window) for 500 milliseconds. The dimension of the target window for the different tasks are shown in Extended Data Fig. 3.During the Respawn task, the target’s position and rotation were changed to the new position when the sphere crossed the grid coordinate equal to two. For the Obstacle task, an additional cube was added to the scene around which the sphere needed to navigate to reach the target.

Avatar animation.

In the Continuous Navigation task, the monkey avatar was equipped with walking, running, and idle animations. The monkey avatar adopted an idle animation, resting on both its arms and legs without any motion, when its velocity was at zero. If the avatar's velocity was higher than zero and below two units per second, the walking animation was activated. When the velocity exceeded 2 units/s, the running animation took over. The speed of the running animation was adjusted based on the actual velocity, providing visual feedback that reflected the avatar's current speed.

Additional functions.

The Continuous Navigation task contained virtual trees in the forest environment. The avatar could be blocked behind a tree if the subject did not navigate properly around it. To ensure the task focused on reaching the targets and not avoiding trees, a special function made sure that if the avatar stayed blocked behind a tree for the entire trial, its starting position was reset to the center of the forest for the next trial. This reset was initiated by checking the avatar's velocity (near zero for over four seconds). The environment in the Continuous Navigation task, whether the grid or the forest, was not infinite. If the avatar was close to the environment's edge, it was reset to the center at the start of the next trial. This was controlled by fixed boundaries to define the environment limits.

Decoding algorithm

The decoding algorithm translated neural activity of PMv, PMd and M1 into the movement of the sphere/avatar in real-time. Initial training of the decoder relied on labeled

521 training data, employing the Preferential Subspace Identification (PSID) algorithm developed
522 by Sani et al. (2021)[31]. Once the decoding algorithm was trained, it was able to predict real-
523 time velocities based on real-time brain signals using a non-linear extension of the PSID
524 method.
525
526 Electrode selection.
527 With our data acquisition system, we could record from 256 electrodes simultaneously.
528 Before each experiment, electrode signals were visually inspected, and if no spikes were
529 detected, the spike threshold was set below a predefined value. During the training of the
530 decoding algorithm, electrodes with thresholds below the predefined value were excluded from
531 the model.
532
533 Training.
534 The training of the decoding algorithm involved a supervised method using labeled data
535 from the Training phase. The data included spike rates of each selected electrode for each 50ms
536 time bin, along with the corresponding velocities (v_x, v_y, v_z) and positions of the sphere/avatar
537 at that timestamp. The PSID method was employed to train the decoding algorithm. The state
538 of the brain was considered as a high-dimensional latent variable at each time point. A dynamic
539 linear state space model was used to characterize both the neural activity and the behavior. Since
540 we only wanted to consider neural dynamics that were related to the behavior of interest, we
541 computed the behaviorally relevant latent variables (Extended Data Fig. 4A) together with the
542 corresponding model parameters in this Decoder Training phase using two parameters: the
543 number of extracted states $n_1 = 6$ and the projection horizon $i = 5$ [31].
544 After extracting latent variables from the neural signal, the next step involved deriving
545 velocities from these latent variables using regression (Extended Data Fig. 4B). Due to the

inherent non-linear relationship between the latent variables and velocities, a kernel approximation was employed as a preprocessing step. Using the Nystroem kernel approximation with the Radial Basis Function (RBF), the data underwent a projection into a higher-dimensional space, enhancing the capturing of non-linear patterns. The Nystroem method approximates the kernel matrix using a smaller subset of data points, reducing the computation time. The chosen parameters of the Nystroem kernel approximation were: $\gamma = 0.34$ that influenced the smoothness of the decision boundary of the Gaussian kernel, $n_{components} = 700$ was the number of data points used in the approximation and $random_state = 42$ was used for reproducibility, ensuring that the same random subset of data points was selected in each run. Next, a linear regression was applied to predict the velocities. The linear regression process identified the optimal-fitting flat hyperplane that minimized the difference between predicted and actual velocities. More specifically, a ridge regression was used, which introduced a regularization term to avoid overfitting the model and prevented potential multicollinearity (high correlation) in the data. The strength of the regularization was controlled by the hyperparameter $\alpha = 0.1$. To ensure that the model learned the relationship between latent variables and velocities based on the provided ground truth, both the kernel approximation and the linear regression were trained using the labeled training data. The different parameters used for the training of the decoder were computed using hyperparameter optimization on a subset of the training data in a preliminary pilot study.

Online decoding.

To predict velocities from neural activity in real-time, a recursive Kalman filter was used in combination with the regression step. The recursive Kalman filter estimated and updated the latent states of the system based on the observed brain signals in a sequential and iterative manner. The filter processed each new observation, in this case the neural activity in a 50ms

time bin, and refined its estimate of the latent states based on the current and past information. The principal advantages of using a recursive Kalman filter are the sequential updating of the latent states, the ability to handle noisy observations and to adapt to changing dynamics over time.

Statistics

Success rate and chance level.

The success rate was defined as the ratio of successful trials to the total number of trials for each session. To assess the significance of differences in success rates across various groups, we calculated Mann-Whitney U tests. The groups compared were the upper and lower targets for the 3D Center-out task, the respawn and no respawn trials for the Respawn task and the on-path and off-path obstacle trials for the Obstacle task (all taken across all sessions). To estimate the chance level of each task, we conducted a permutation test. For each 10,000 permutations, the target labels were shuffled and a test statistic—in this case, the success rate across trials—was computed. By comparing the observed statistic (success rate of the monkey across all trials and sessions) to the distribution of the permuted statistics, we could quantify the likelihood of observing such performance by chance. To determine if the observed success rate was significantly different from what could occur by random chance, we computed the p-value by determining the proportion of permuted test statistics that were as, or more, extreme than the observed success rate.

Reaction time.

We computed the reaction time in the Respawn task as the time interval between the change in target position and the detected point of deviation in the sphere's trajectory, which

corresponds to the monkey's response to the target change. We started by smoothing the trajectory using a low-pass Butterworth filter and a moving average. Next, we computed the tangents along the trajectory to determine the movement direction at each time point following the target jump time. The reaction point was defined as the first significant deviation in movement direction after the target jump, identified using a predefined angular threshold. Additionally, we took into account the delay (approximately 100ms) between the change in brain activity, which corresponds to the intent to change the sphere's direction, and the actual movement of the sphere on the screen.

Tim to target.

We calculated the time it took the monkey to move the sphere/avatar to the target. This time-to-target measurement started from the moment the sphere/avatar was allowed to move and ended when the trial was successful, defined as staying for 500ms within the target window. We compared the time-to-target during the Training phase, where unity was in control, and the Online Decoding phase, where the monkey was in control.

Influence of sEMG on decoded velocity.

To investigate whether arm and hand movement influenced the decoded velocity during the Training and Online Decoding phase, we recorded sEMG from the biceps (biceps brachii) and the thumb (abductor pollicis brevis) of Monkeys 1 and 2, using two dry-adhesive electrodes, along with a ground electrode placed next to the biceps electrode. The raw sEMG was preprocessed by applying a high-pass Butterworth filter (order = 4, cut-off frequency = 30Hz), a 50Hz notch filter, and a low-pass Butterworth filter (order = 4, cut-off frequency = 500Hz). The signal was rectified, and its envelope was obtained using a Butterworth low-pass filter

621 (order = 4, cut-off frequency = 20Hz)[35]. For every trial, the sEMG signal was segmented into
622 50ms bins to align with the decoded velocity, averaged over each bin, after which all trials of a
623 session were concatenated.

624 We assessed the relationship between sEMG and decoded velocities using Spearman
625 correlation analysis. We computed the mean correlation across all sessions and the combined
626 p-value (Fisher's method) for the different tasks and for the different velocity components (v_x,
627 v_y, v_z, v = magnitude of velocity). To account for potential time delays, cross-correlation
628 analysis was conducted with lags ranging from -100 to +100ms (50ms increments). A positive
629 lag indicates that the sEMG follows the decoded velocity, while a negative lag suggests that the
630 sEMG precedes the decoded velocity, considering typical delays from motor cortex to sEMG
631 (50-150ms[36]) and to decoded velocity (50-150ms). Additionally, we performed linear and non-
632 linear regression (polynomial, degree = 2) analyses to evaluate how well sEMG predicted the
633 decoded velocities by computing the R^2 and the mean squared error (MSE) values, together
634 with the combined p-value.

635

636 Influence of eye-movements on decoded velocity.

637 During the Training and Online Decoding phase, the monkey watched a screen
638 displaying tasks while its eye movements were monitored continuously. Except for the
639 Continuous Navigation task, the monkey had to maintain its gaze within a large electronically
640 defined window that contained both the initial position of the sphere and the target. For the
641 Continuous Navigation task, the eye-window matched the screen size, because the camera view
642 rotated with the sphere/avatar. To estimate the influence of eye movements on the decoded
643 velocities, we computed the Spearman correlation between the eye position (x and y-
644 component) and the decoded velocity components (v_x, v_y, v_z, v = magnitude of velocity). The
645 eye movement data were segmented into 50ms bins, and the average value was taken as a

feature. The correlation coefficients were computed similar to those between the sEMG and the decoded velocity.

References

1. *Neural Interface: Frontiers and Applications*. http://www.springer.com/series/5584.
2. Santhanam, G., Ryu, S. I., Yu, B. M., Afshar, A. & Shenoy, K. V. A high-performance brain-computer interface. *Nature* **442**, 195–198 (2006).
3. Hochberg, L. R. *et al.* Neuronal ensemble control of prosthetic devices by a human with tetraplegia. *Nature* **442**, 164–171 (2006).
4. Taylor, D. M., Helms Tillery, S. I. & Schwartz, A. B. *Direct Cortical Control of 3D Neuroprosthetic Devices*. https://www.science.org.
5. Musallam, S., Corneil, B. D., Greger, B., Scherberger, H. & Andersen, R. A. *Cognitive Control Signals for Neural Prosthetics*. *Annu. Rev. Biochem* vol. 296 www.sciencemag.org/cgi/content/full/305/5681/254/ (2004).
6. Velliste, M., Perel, S., Spalding, M. C., Whitford, A. S. & Schwartz, A. B. Cortical control of a prosthetic arm for self-feeding. *Nature* **453**, 1098–1101 (2008).
7. Hochberg, L. R. *et al.* Neuronal ensemble control of prosthetic devices by a human with tetraplegia. *Nature* **442**, 164–171 (2006).
8. Aflalo, T. *et al.* Decoding motor imagery from the posterior parietal cortex of a tetraplegic human. *Science (1979)* **348**, 906–910 (2015).
9. Serruya, M. D. ;, Hatsopoulos, N. G. ;, Paninski, L. ;, Fellows, M. R. ; & Donoghue, J. P. Instant neural control of a movement signal. *Nature* **416**, 141–142 (2002).
10. Clanton, S. T., Schwartz, A., Pollard, N., Stetten, G. & Hogan, N. *Brain-Computer Interface Control of an Anthropomorphic Robotic Arm*. (2011).

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

12. Collinger, J. L. *et al.* High-performance neuroprosthetic control by an individual with tetraplegia. *The Lancet* **381**, 557–564 (2013).

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

14. Hochberg, L. R. *et al.* Reach and grasp by people with tetraplegia using a neurally controlled robotic arm. *Nature* **485**, 372–375 (2012).

15. Ajiboye, A. B. *et al.* Restoration of reaching and grasping movements through brain-controlled muscle stimulation in a person with tetraplegia: a proof-of-concept demonstration. *The Lancet* **389**, 1821–1830 (2017).

16. Jarosiewicz, B. *et al. Virtual Typing by People with Tetraplegia Using a Self-Calibrating Intracortical Brain-Computer Interface*. *Chethan Pandarinath* vol. 5 https://www.science.org.

17. Pandarinath, C. *et al.* High performance communication by people with paralysis using an intracortical brain-computer interface. *Elife* **6**, (2017).

18. Willett, F. R., Avansino, D. T., Hochberg, L. R., Henderson, J. M. & Shenoy, K. V. High-performance brain-to-text communication via handwriting. *Nature* **593**, 249–254 (2021).

19. Moses, D. A. *et al.* Neuroprosthesis for Decoding Speech in a Paralyzed Person with Anarthria. *New England Journal of Medicine* **385**, 217–227 (2021).

20. Moritz, C. T., Perlmutter, S. I. & Fetz, E. E. Direct control of paralysed muscles by cortical neurons. *Nature* **456**, 639–642 (2008).

21. Capogrosso, M. *et al.* A brain-spine interface alleviating gait deficits after spinal cord injury in primates. *Nature* **539**, 284–288 (2016).

22. Flesher, S. N. *et al. Intracortical Microstimulation of Human Somatosensory Cortex*. https://www.science.org.

23. Taylor, D. M., Helms Tillery, S. I. & Schwartz, A. B. *Direct Cortical Control of 3D Neuroprosthetic Devices*. https://www.science.org.

24. Wang, W. *et al.* An Electrocorticographic Brain Interface in an Individual with Tetraplegia. *PLoS One* **8**, (2013).

25. Flanagan, J. R. & Johansson, R. S. Action plans used in action observation. *Nature* **424**, 769–771 (2003).

26. Libedinsky, C. *et al.* Independent mobility achieved through a wireless brain-machine interface. *PLoS One* **11**, (2016).

27. Jay Edelman, B. *et al. On Closed-Loop Brain Stimulation Systems for Improving the Quality of Life of Patients with Neurological Disorders*.

28. Verhoef, B. E., Vogels, R. & Janssen, P. Contribution of Inferior Temporal and Posterior Parietal Activity to Three-Dimensional Shape Perception. *Current Biology* **20**, 909–913 (2010).

29. Verhoef, B. E., Vogels, R. & Janssen, P. Inferotemporal Cortex Subserves Three-Dimensional Structure Categorization. *Neuron* **73**, 171–182 (2012).

30. Van Dromme, I. C., Premereur, E., Verhoef, B. E., Vanduffel, W. & Janssen, P. Posterior Parietal Cortex Drives Inferotemporal Activations During Three-Dimensional Object Vision. *PLoS Biol* **14**, (2016).

31. Sani, O. G. , A. H. , W. Y. T. , P. B. , & S. M. M. Modeling behaviorally relevant neural dynamics enabled by preferential subspace identification. *Nat Neurosci* **24**, 140–149 (2021).

32. Gilja, V. *et al.* A high-performance neural prosthesis enabled by control algorithm design. *Nat Neurosci* **15**, 1752–1757 (2012).

33. Moritz, C. T., Perlmutter, S. I. & Fetz, E. E. Direct control of paralysed muscles by cortical neurons. *Nature* **456**, 639–642 (2008).

34. WDallgraphics. Capuchin Monkey. https://assetstore.unity.com/packages/3d/characters/animals/mammals/capuchin-monkey-216678.

35. Esposito, D., Centracchio, J., Bifulco, P. & Andreozzi, E. A smart approach to EMG envelope extraction and powerful denoising for human–machine interfaces. *Sci Rep* **13**, (2023).

36. Morrow, M. M. & Miller, L. E. Prediction of muscle activity by populations of sequentially recorded primary motor cortex neurons. *J Neurophysiol* **89**, 2279–2288 (2003).