

# Computational framework to predict and shape human–machine interactions in closed-loop, co-adaptive neural interfaces

Received: 24 May 2024

Accepted: 24 January 2026

Published online: 23 March 2026

 Check for updates

Maneeshika M. Madduri<sup>1</sup>, Momona Yamagami<sup>1,2</sup>, Si Jia Li<sup>3</sup>, Sasha Burckhardt<sup>4</sup>, Samuel A. Burden<sup>1</sup> & Amy L. Orsborn<sup>1,3,5</sup> 


Neural interfaces can restore or augment human sensorimotor capabilities by converting high-bandwidth biological signals into control signals for an external device via a decoder algorithm. Leveraging user and decoder adaptation to create co-adaptive interfaces presents opportunities to improve usability and personalize devices. However, we lack principled methods to model and optimize the complex two-learner dynamics that arise in co-adaptive interfaces. Here we present computational methods based on control theory and game theory to analyse and generate predictions for user–decoder co-adaptive outcomes in continuous interactions. We tested these computational methods using an experimental platform in which human participants ( $N = 14$ ) learn to control a cursor using an adaptive myoelectric interface to track a target on a computer display. Our framework allowed us to characterize user and decoder changes within co-adaptive myoelectric interfaces. Our framework further allowed us to predict how changes in the decoder algorithm impacted co-adaptive interface performance and revealed how interface properties can shape user behaviour. Our findings demonstrate an experimentally validated computational framework that can be used to design user–decoder interactions in closed-loop, co-adaptive neural interfaces. This framework opens future opportunities to optimize co-adaptive neural interfaces to expand the performance and application domains for neural interfaces.

Neural interfaces can restore or augment human capabilities<sup>1–6</sup>. In a neural interface, signals from the user are translated via a decoder algorithm to control a device. Decoder algorithms that adapt to users during interface operations can improve performance and enable personalization<sup>2,7–17</sup>. Users also adapt as they learn to control the interface because they receive real-time feedback<sup>3,18–21</sup>. Introducing adaptive algorithms into neural interfaces, therefore, creates a co-adaptive system. Prior work has demonstrated the benefits of leveraging

co-adaptation to guide or assist user adaptation<sup>10,22,23</sup> and maintain performance over time<sup>10,14,24</sup>.

However, co-adaptive systems are challenging to design because they involve dynamic, two-learner interactions: both user and decoder may adapt simultaneously and in response to one another. Experiments have exposed the importance of identifying appropriate decoder parameters, such as the rate of decoder adaptation, to ensure stable and predictable interface control<sup>7,25,26</sup>. Methods to design and

<sup>1</sup>Department of Electrical and Computer Engineering, University of Washington, Seattle, WA, USA. <sup>2</sup>Department of Electrical and Computer Engineering, Rice University, Houston, TX, USA. <sup>3</sup>Department of Bioengineering, University of Washington, Seattle, WA, USA. <sup>4</sup>Undergraduate Program in Neuroscience, University of Washington, Seattle, WA, USA. <sup>5</sup>Washington National Primate Research Center, University of Washington, Seattle, WA, USA.

 e-mail: [aorsborn@uw.edu](mailto:aorsborn@uw.edu)

analyse two-learner systems are an active area of research in neural engineering<sup>27,28</sup> and dynamic game theory<sup>29</sup>, but co-adaptive neural interfaces are largely designed and implemented empirically. Theoretically grounded and experimentally validated techniques to model and quantify co-adaptive dynamics will unlock new ways to design robust interfaces that harness the full potential of user and decoder adaptation.

Existing frameworks for designing co-adaptive interfaces highlight the promise of model-based approaches, but they either do not capture the full range of dynamic interactions that can arise between users and decoders or do not make predictions about interaction outcomes. Past work on neural interfaces focused on the user adapting to a leading, optimal (fixed) decoder<sup>30</sup> or the decoder adapting to follow the user<sup>31</sup>, and most experimental neural interfaces are constrained to this limited range of co-adaptive interactions<sup>2,10,14,22</sup>. We instead seek methods that will be able to accommodate a wide spectrum of user–decoder interactions, where the device may shift from ‘leading’ to ‘following’ the user over time<sup>32</sup>. Game theory has been used to analyse the joint responses of two learners in human–human motor interactions<sup>9,33</sup> and human–machine interactions<sup>25,34,35</sup>. We built on these ideas, adapting tools from control theory and game theory to create a flexible, experimentally validated framework for the analysis and synthesis of co-adaptive outcomes of continuous interactions in neural interfaces.

We developed a myoelectric interface platform that allowed us to build experimentally validated frameworks for co-adaptive neural interfaces. We validated that users changed their behaviour alongside an adaptive decoder to control a cursor via their muscle activity, creating a co-adaptive system. We modelled this closed-loop system using control theory methods to quantify interactions between users and the device. This experimental platform and analysis toolkit allowed us to test analytical predictions about user–machine behaviours derived from a model treating the user and decoder as two agents in a game. Our experimental results revealed that user learning can be influenced through real-time interface adaptation and that these outcomes can be predicted using game-theoretic models. Together, our findings establish methods to analyse and synthesize co-adaptive systems that can enable next-generation approaches for designing and personalizing neural interfaces.

## Results

We created an experimental platform to study co-adaptation using non-invasive myoelectric interfaces, which we subsequently used as a testbed to develop and validate computational frameworks for co-adaptive interactions. This platform created unfamiliar human–machine interactions for our human participants ( $N = 14$ , demographics in Table 1) that let us systematically manipulate decoder properties to investigate the effect of decoder adaptation on user adaptation. Here we use the term ‘adaptation’ to describe any change over time in the decoder or user, and do not aim to draw analogies to particular forms of motor learning<sup>36</sup>. Our myoelectric interface had similar properties to many common motor neural interfaces, including brain–computer interfaces: it used high-dimensional biosignals to control a lower-dimensional movement<sup>37,38</sup> and was a closed-loop system that can facilitate user adaptation<sup>12</sup>. We first analysed the experimental platform to establish that both users and decoders adapt, creating a co-adaptive system (see the ‘An adaptive myoelectric interface created a co-adaptive experimental platform’ section). We then leveraged control theory and game theory to develop computational methods that can measure and predict co-adaptive outcomes (see the ‘Co-adaptation outcomes matched control theory predictions’ and ‘Game theory model generated predictions for co-adaptation outcomes’ sections). Finally, we used our experimental platform to validate model predictions by manipulating the decoder adaptation parameters and demonstrate that our framework could predict the outcomes of co-adaptive interactions

**Table 1 | Participant demographics**

	Responses
Gender	Women, 9; men, 4; prefer not to answer, 1
Weight (mean $\pm$ s.d., pounds)	140 $\pm$ 21.6
Height (mean $\pm$ s.d., inches)	65 $\pm$ 3.4
Age (mean $\pm$ s.d., years)	23 $\pm$ 3.9
Forearm circumference (mean $\pm$ s.d., inches)	10 $\pm$ 0.8
Handedness	Right-handed, 12; left-handed, 2
Level of ambidexterity	Dominant, 8; mostly dominant, 5; ambidextrous, 1

(see the ‘Decoder effort penalty influenced user effort without changing performance’ section).

### An adaptive myoelectric interface created a co-adaptive experimental platform

Participants completed a two-dimensional trajectory tracking task using a 64-channel adaptive myoelectric interface (Fig. 1a and Extended Data Fig. 1a). The task was chosen for its advantages in analysing user sensorimotor transforms<sup>39,40</sup>. Participants controlled a cursor using surface electromyography (EMG) signals measured from their dominant forearm via a velocity-controlled Wiener filter decoder (Methods). The decoder was randomly initialized at the beginning of each trial and then updated online every 20 s using previously developed adaptive decoding algorithms (SmoothBatch<sup>41</sup>). Decoder adaptation conditions varied from trial to trial, and participants completed a total of 16 five-minute trials (arranged in two blocks of eight trials; Fig. 1b and Methods).

We first analyse the general properties of system behaviour independent of decoder adaptation conditions before examining the influence of decoder conditions (learning rate and cost functions; see the ‘Game theory model generated predictions for co-adaptation outcomes’ and ‘Decoder effort penalty influenced user effort without changing performance’ sections, respectively). The performance improved as the decoder adapted during a trial (Wilcoxon signed-rank test,  $P < 0.001$ ; Fig. 1c,d), and these performance improvements were not impacted by decoder initialization (Wilcoxon signed-rank test,  $P > 0.05$ ; Fig. 1e), similar to invasive brain–computer interface adaptive decoder studies<sup>27,41,42</sup>. These results confirm the expected behaviour of adaptive algorithms.

Although past work suggests that users probably adapt alongside the algorithm, we next aimed to confirm this behaviour in our testbed. System-level performance metrics alone cannot separate the effect of user adaptation from that of decoder adaptation because performance changes could be attributed to either learner. The decoder adapted during trial by design, leading to shifts in the decoder matrix ( $D$ ) according to the prescribed cost functions (for example, Fig. 5c, also reported in preliminary studies<sup>26</sup>). To validate that our myoelectric interfaces created a co-adaptive system, we examined whether users adapted during each trial. We asked whether the user’s behaviour changed as they controlled the interface by estimating the relationship between the user surface EMG activity and the direction of intended movement—that is, an EMG tuning curve<sup>43–45</sup> (Methods). Variation in electrode placement and user behaviour resulted in a wide range of EMG tuning curves that varied across participants and across electrodes (Extended Data Fig. 2). EMG tuning curves from a sample participant revealed within-trial changes in user behaviour (Fig. 1f,g). We saw a range of changes in the EMG tuning curves in other participants (Extended Data Fig. 3). We quantified EMG changes for each participant by calculating the magnitude of the difference between EMG tuning curves (Fig. 1h) and angular differences in the preferred direction (Fig. 1i; Methods).

Because changes in EMG tuning were diverse across electrodes and participants, we assayed whether changes across a trial reflected a directed process. We compared the magnitude of change in the first 30 s (early) versus the last 30 s (late) to the magnitude of change on consecutive 30-s intervals. All participants showed significantly larger changes in the magnitude and tuning angles of EMG activity over the course of the trial than moment-to-moment fluctuations, suggesting that users adapted their behaviour in a directed fashion during the five-minute trial (Wilcoxon signed-rank test with a Bonferroni correction of  $N = 14$ ,  $P < 0.001$ ; Fig. 1h,i).

Interestingly, we observed that within-trial differences in the EMG tuning curves decreased as the trials progressed (Fig. 1j). We hypothesized that these longer timescale changes reflected user learning over the course of the experiment. We tested this prediction by comparing performance in the first and second blocks, which contained trials with matched decoder adaptation conditions. Decoder adaptation parameters were matched across blocks of trials, which means that differences in task performance (determined by the joint user–decoder system) must stem from differences in the user behaviour between blocks. We indeed saw that participants' average mid-trial and final task error was lower in block 2 compared with block 1 (Wilcoxon signed-rank test,  $P < 0.05$ ), suggesting that the task performance trajectories improved increasingly quickly in block 2 than in block 1 (Fig. 1k). Taken together, the user EMG and system performance suggest that users adapt alongside the decoder in our myoelectric interface, both within and across trials, establishing a co-adaptive system.

### Co-adaptation outcomes matched control theory predictions

Distinguishing user and decoder contributions to an interface is key to the understanding and designing of closed-loop, co-adaptive systems. Although decoders are known exactly, the user model must be estimated. The EMG direction tuning analysis we used above can characterize the user activity but does not provide a model for the user and, therefore, limits our ability to examine user–decoder interactions. We used methods from control theory, a discipline focused on the analysis and synthesis of feedback systems, to model the behaviour of users as well as the closed-loop system. Prior work shows that control theory can estimate changes in user trajectory tracking behaviour by separating their feedback and feed-forward controllers<sup>39,46–48</sup>. We extended these techniques to our multi-input, multi-output system to analyse co-adaptive outcomes. These approaches involve modelling users as performing position-based control with perfect and immediate sensory feedback. These models, therefore, notably simplify computations known to contribute to motor control (such as sensory delays). Yet, these and related simplified models have proven

useful for capturing aspects of motor behaviour in humans and other model organisms<sup>40,49,50</sup>.

We define our model of the user's encoder as the closed-loop mapping between task information and EMG activity in our myoelectric interface (Fig. 2a; Extended Data Table 1 lists a summary of variables). Our encoder model,  $E \in \mathbb{R}^{64 \times 8}$ , considers feed-forward and feedback inputs to the user, which are derived from closed-loop task information presented to the user, namely, target position ( $\tau \in \mathbb{R}^2$ ), target velocity ( $\dot{\tau} \in \mathbb{R}^2$ ), position error ( $\tau - y \in \mathbb{R}^2$ ) and velocity error ( $\dot{\tau} - \dot{y} \in \mathbb{R}^2$ ). We formulate an encoder model with these feed-forward and feedback elements that depend linearly on the target position, cursor error and offset  $\beta$  (to capture the resting activity):

$$u = E \cdot \begin{bmatrix} \tau \\ \dot{\tau} \\ \tau - y \\ \dot{\tau} - \dot{y} \end{bmatrix} + \beta \cdot \mathbf{1}_t^T, \quad (1)$$

where  $\mathbf{1}_t^T = [1 \dots 1] \in \mathbb{R}^{1 \times t}$ , and  $t$  indicates time.

The encoder matrix has both feed-forward ( $F$ ) and feedback ( $B$ ) components:

$$E = [F_0 F_1 B_0 B_1] \quad (2)$$

such that each component of task information,  $\tau$ ,  $\dot{\tau}$ ,  $\tau - y$  and  $\dot{\tau} - \dot{y}$  corresponds to an element of the encoder, namely,  $F_0 \in \mathbb{R}^{64 \times 2}$ ,  $F_1 \in \mathbb{R}^{64 \times 2}$ ,  $B_0 \in \mathbb{R}^{64 \times 2}$  and  $B_1 \in \mathbb{R}^{64 \times 2}$ , respectively. The subscripts 0 and 1 represent the zeroth- and first-order dynamics of position and velocity, respectively. The output of the encoder is an EMG activity time series  $u \in \mathbb{R}^{64 \times t}$  that is input to the decoder,  $D \in \mathbb{R}^{2 \times 64}$ . The decoder outputs a cursor velocity, which is integrated by the system dynamics  $M$  to a cursor position  $y$ . We estimated the user's encoders  $E \in \mathbb{R}^{64 \times 8}$  within a trial with linear regression using 20-s batches of data, corresponding to the time intervals in which the decoder was held constant (Methods). This model appeared to capture useful aspects of user behaviour, yielding reconstructed cursor position and velocity trajectories that were correlated with the actual cursor movements (Fig. 2b).

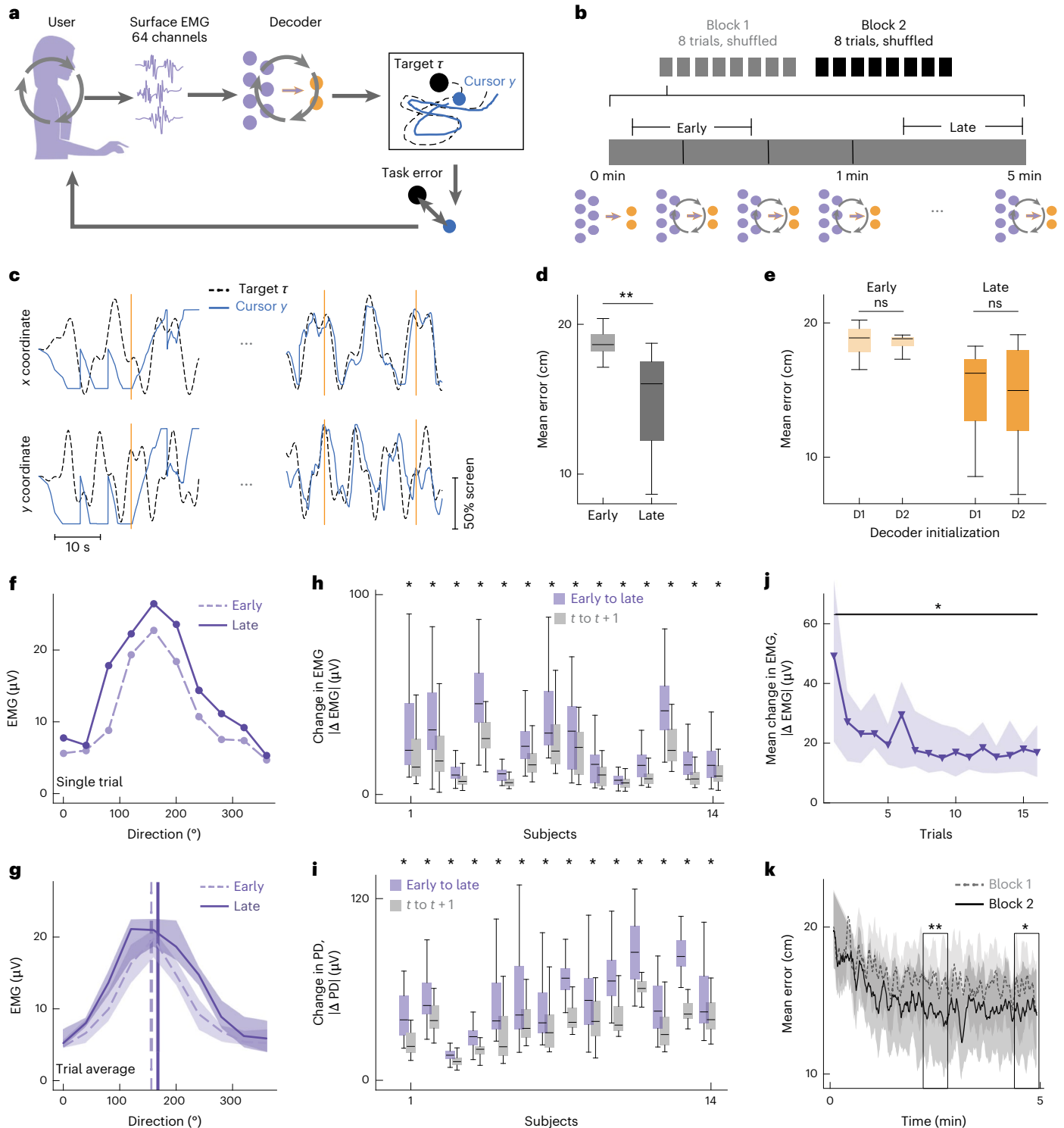
We then used our encoder model to quantify encoder–decoder interactions and analyse the performance of the closed-loop system. One common challenge in co-adaptive systems is convergence to a stable solution or performance<sup>25,27,28</sup>. Our performance-based analysis above suggested that our co-adaptive interface may have converged (Fig. 1c,d), but neither proves stability nor ideal performance. We

**Fig. 1 Co-adaptive myoelectric interface.** **a**, Experiment schematic. Users tracked a target by controlling the two-dimensional cursor velocity with muscle activity via an adaptive decoder. **b**, Session schematic. Participants completed two blocks of 8 five-minute trials with different, randomized decoder conditions; decoders were initialized randomly and adapted every 20 s. **c**, Example cursor (solid blue) and target (dashed black) trajectories. The orange lines denote decoder updates. **d**, Mean error ( $N = 14$ , centre shows median; box shows 25th–75th percentiles; whiskers extend to  $1.5 \times$  the interquartile range) early (first 30 s) and late (last 30 s) in trials. Error ( $\|\tau - y\|$ ) was averaged across trials for each participant. One-sided Wilcoxon signed-rank test,  $**P = 0.0001$ . **e**, Mean error ( $N = 14$ ) separated by decoder initialization (D1, D2) early and late in trials. Error ( $\|\tau - y\|$ ) was averaged across trials for each participant. Two-sided Wilcoxon signed-rank test, (early)  $ns = 0.67$ , (late)  $ns = 0.43$ . **f**, EMG tuning curve for one EMG channel early (dashed light purple) and late (solid purple) within a single trial. **g**, Average EMG tuning curve for one channel early (dashed light purple) and late (solid purple) across all trials ( $N = 16$ , median; shading shows the 25th–75th percentile range). The preferred directions for each curve are shown with vertical lines. **h**, Norm difference of the average early versus late EMG tuning curves ( $\|\Delta\text{EMG}\| = \|\text{EMG}(\text{late}) - \text{EMG}(\text{early})\|$ ) (purple) compared with

norm difference of the average consecutive 30-s intervals ( $\|\Delta\text{EMG}\| = \|\text{EMG}(t) - \text{EMG}(t+1)\|$ ) (grey) for all channels, computed for each subject ( $N = 63$ , one-sided Wilcoxon signed-rank test with a Bonferroni correction of  $n = 14$ ;  $*P < 0.001$  for all subjects). **i**, Magnitude of angular change in EMG-preferred direction early versus late ( $\|\Delta\text{PD}\| = \|\text{PD}(\text{late}) - \text{PD}(\text{early})\|$ ) (purple) compared with angular change in consecutive 30-s intervals ( $\|\Delta\text{PD}\| = \|\text{PD}(t) - \text{PD}(t-1)\|$ ) (grey) for all channels, computed for each subject ( $N = 63$ , one-sided Wilcoxon signed-rank test with a Bonferroni correction;  $*P < 0.001$  for all subjects). **j**, Average change in EMG tuning curves within a trial ( $\|\Delta\text{EMG}\| = \|\text{EMG}(\text{late}) - \text{EMG}(\text{early})\|$ ) for all subjects as a function of trial number ( $N = 14$ , median; shading shows the 25th–75th percentile range). Average change in EMG of the first trial is compared with the last trial (one-sided Wilcoxon signed-rank test,  $*P = 0.0083$ ). **k**, Average error as a function of trial time in block 1 (dashed grey) and block 2 (solid black) ( $N = 14$ , median; shading shows the 25th–75th percentile range). The black boxes represent comparisons of mid-trial and end-trial task error between block 1 and block 2 (one-sided Wilcoxon signed-rank test,  $**P = 0.00043$ ,  $*P = 0.015$ ). All box plots (**d**, **e**, **h** and **i**) show the median (centre line), 25th–75th percentiles (box; interquartile range) and  $1.5 \times$  the interquartile range (whiskers). Panel c adapted with permission from ref. 26, Elsevier.

exploited the separation of the feed-forward and feedback pathways in our encoder model to empirically measure the stability and tracking error properties of the closed-loop encoder–decoder system. Perfect trajectory tracking is obtained when the user’s feed-forward input  $F$  becomes the pseudo-inverse of the decoder  $D$  and system dynamics  $M$ , which results in the following conditions:  $D \cdot F_0 = 0$  and  $D \cdot F_1 = I$ , where  $0$  is the zero matrix and  $I$  is the identity matrix (Methods and equation (18)). We used known decoder values and estimated user encoders to quantify these values in our experimental data, which revealed that  $D \cdot F_1$  approaches the analytically ideal values on average across all trials (Fig. 2c). Interestingly, we found that  $D \cdot F_0$  slightly deviated from the

expected values for perfect tracking (Fig. 2c). These deviations from the analytical ideal were found to correlate with the user tracking error (Extended Data Fig. 4), suggesting that our encoder estimates capture meaningful properties of user behaviour. Similarly, closed-loop stability can be achieved if the feedback components of the encoder–decoder product ( $D \cdot B_0$  and  $D \cdot B_1$ ) are approximately negative-definite diagonal matrices. In this case, the eigenvalues of the closed-loop system’s dynamics are approximately located at the diagonal entries (that is, in the left-half complex plane), ensuring stability (chapter 5 of ref. 51). Our experimental data observations matched the predictions for a stable system (Fig. 2d).



We also used our encoder estimation model to revisit user-learning-related changes in experiments. We found that user encoder subspaces showed larger changes across the five-minute trial compared with subsequent time intervals (Wilcoxon signed-rank test,  $P < 0.001$ ; Fig. 2e (left) and Methods), consistent with directed changes observed in the EMG tuning curves (Fig. 1h,i). As expected, decoder subspaces also showed directed changes within each trial (Wilcoxon signed-rank test,  $P < 0.001$ ; Fig. 2e (right) and Methods). We further observed that the within-trial change in user encoder subspaces decreased across sequential trials (Fig. 2f), mirroring our EMG and performance-based analyses, showing that users also learn over the course of the experiment (Fig. 1j,k). Decoder changes within each trial, by contrast, did not notably vary across trials in the experiment since the decoders were reinitialized randomly at the beginning of each trial (Fig. 2f). Our models, therefore, revealed the evidence of co-adaptation between the encoder and decoder, which was consistent with our EMG-based analyses.

Finally, encoder and decoder changes within each trial, paired with system-level analyses, revealed nuances of how the co-adaptive system converged (Fig. 2g). The amount of change between consecutive decoders starkly decreased over the course of a trial, consistent with the algorithm converging towards a solution from its random initialization. User encoders, by contrast, changed a similar amount over the trial. This is consistent with user learning across trials, where they do not ‘reinitialize’ their encoder on each trial. Importantly, both encoders and decoders do not converge towards zero change within the five-minute trial, despite the fact that encoder–decoder pairs (Fig. 2c,d) and task performance (Fig. 1k) appear to converge. Considered together, our analyses suggest that the joint user–decoder system converges, yet neither agent becomes completely stationary. By modelling the user, our control-theoretic encoder model allowed us to more precisely resolve user–decoder interactions within the closed-loop, co-adaptive interface.

### Game theory model generated predictions for co-adaptation outcomes

Having established an experimental platform for co-adaptive interfaces and validated tools to analyse user–decoder adaptation, we next aimed to build tools to predict how users and decoders will interact. We leveraged game theory to model the co-adaptive interactions that arise between these two agents optimizing their individual objectives<sup>52,53</sup> (Fig. 3a). The two agents in our experiment—the user and decoder—co-adapt to achieve a shared goal: reduce task error. However, the user probably trades off error with effort, preferring strategies that achieve reasonable performance and reducing muscle activity<sup>54</sup>. The decoder’s adaptation algorithm was analogously designed to trade off error with the magnitude of transformation (or gain) it defines between the EMG

inputs and cursor velocity outputs. We, therefore, defined cost functions for the user’s encoder  $E$  and the decoder  $D$  as a linear combination of task error  $e$  and agent effort  $f$ :

$$c_E(E, D) = e(E, D) + \lambda_E f_E(E), \quad c_D(E, D) = e(E, D) + \lambda_D f_D(D), \quad (3)$$

where  $\lambda_E$  and  $\lambda_D$  are effort penalties. For tractability, we limit our analysis in this section to the one-dimensional case, where  $E$  and  $D$  are scalars, as in ref. 55. In our simplified case, the error in our task is the square difference between the feed-forward decoder output and target velocities:  $e(E, D) = (\dot{r} - DE\dot{r})^2$ . We similarly defined effort as  $f_E(E) = E^2$ ,  $f_D(D) = D^2$ .

The structure of these cost functions define a potential game<sup>56</sup> (Fig. 3b), which can be analysed in terms of a single potential function:

$$\phi(E, D) = e(E, D) + \lambda_E f_E(E) + \lambda_D f_D(D). \quad (4)$$

Using this potential function, we can characterize the existence of and convergence to stationary points ( $E^*, D^*$ ) defined by the conditions  $\partial_E \phi(E^*, D^*) = 0$  and  $\partial_D \phi(E^*, D^*) = 0$ . These game-theoretic stationary points represent joint strategies in the co-adaptive system that persist for a prolonged period of time. Such stationary points in a neural interface could correspond to stable neural representations (neural ‘encoding’) and stable decoder parameters, which have been observed in co-adaptive brain–computer interfaces<sup>2,10,14,27,41</sup>.

We model encoder and decoder adaptation in discrete time using gradient descent for  $E$  and smoothed best response for  $D$ :

$$E^+ = E - \alpha_E \partial_E \phi(E, D), \quad D^+ = \alpha_D D + (1 - \alpha_D) \arg \min_D \phi(E, D), \quad (5)$$

where  $\alpha_E > 0$ ,  $\alpha_D \in (0, 1)$  are learning rate parameters. The update for  $D$  is consistent with what is implemented in the experiments (Methods) and was based on the methods in refs. 10,41. We propose a gradient-based update for  $E$  as a model for user learning that incrementally improves performance, which is consistent with our findings for how encoders change (Fig. 2e–g).

Analysing game dynamics (Methods provides the calculation details) shows that the existence of stationary points is unaffected by learning rates  $\alpha_E$ ,  $\alpha_D$ , but these parameters determine the system’s overall rate of convergence—or, indeed, whether convergence occurs at all (Figs. 3d and 4a). When  $\lambda_E$  and  $\lambda_D$  are less than unity, there are three stationary points ( $E^*, D^*$ ): one at the origin and a mirror pair (Methods provides the analytical expressions). The stationary point at the origin is a saddle, and the other two are local minima of the agents’ shared potential function (Methods). The existence of multiple stationary points in the co-adaptive model suggests the same for the experimental co-adaptive system, and the initialization of either agent

### Fig. 2 | Control-theoretic analysis allows for user encoder estimation and encoder–decoder analysis. a, Block diagram model of the closed-loop system.

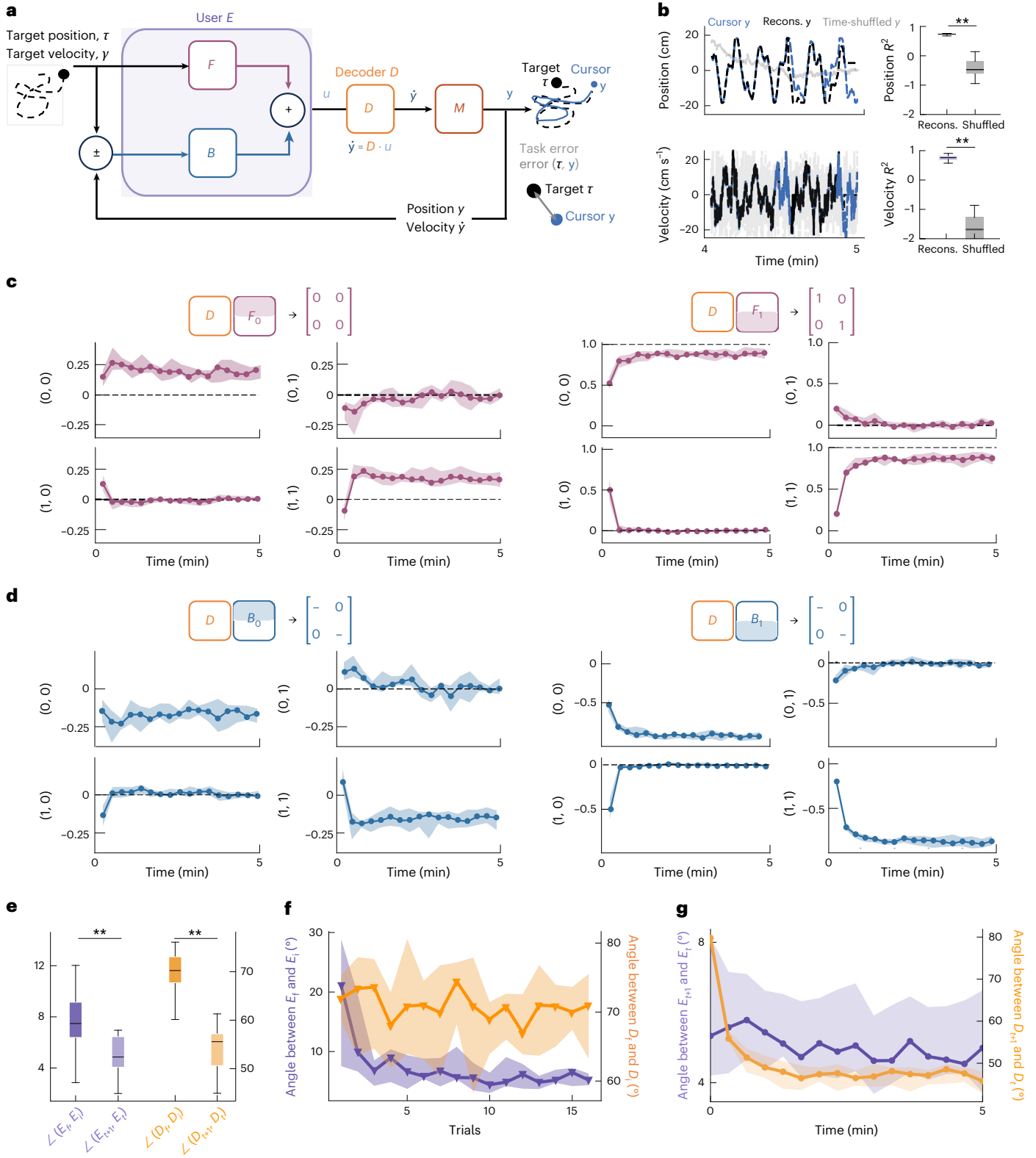
The user’s encoder  $E$  outputs myoelectric activity  $u$  to the decoder  $D$  to follow the target’s position and velocity. The decoder outputs velocity, which is then integrated by system dynamics  $M$  to generate a cursor position. The user sees their cursor moving with some velocity  $\dot{y}$  to position  $y$ . Error in the cursor position and velocity (relative to the target) acts as closed-loop feedback to the user. **b**, Example reconstructions of cursor position (top) and velocity (bottom) from encoder estimations (dashed black), compared with the actual position and velocity (dashed blue) with a time-shuffled baseline (solid grey). Distributions of average  $R^2$  values between the actual and reconstructed position (top right) and velocity (bottom right) for actual data versus time-shuffled controls ( $N = 14$ , median; shading shows the 25th–75th percentile; two-sided Wilcoxon signed-rank test, (position)  $**P = 0.0001$ , (velocity)  $**P = 0.0001$ ). **c**, Product of the decoder matrix with feed-forward contributions of the encoder matrix ( $N = 14$ , median; shading shows the 25th–75th percentile). The black dashed lines represent values that yield perfect trajectory tracking. **d**, Product of the decoder

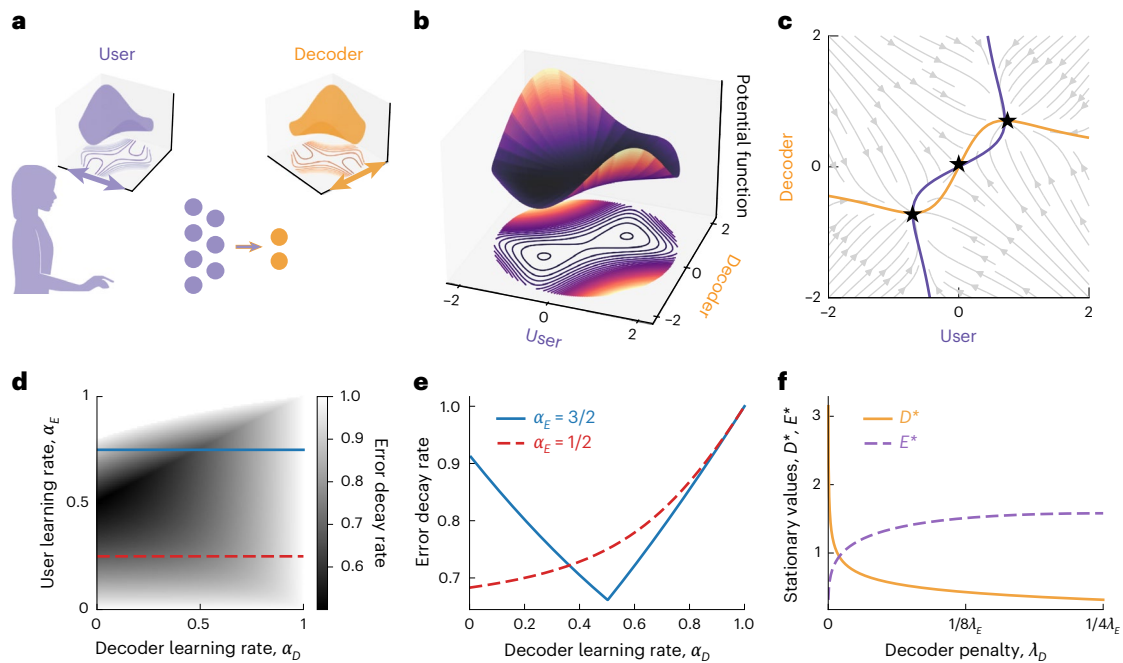
matrix with feedback contributions of the encoder matrix ( $N = 14$ , median; shading shows the 25th–75th percentile). The black dashed lines represent values that yield closed-loop stability. **e**, Average change in angle between the final and initial encoders ( $\angle(\mathcal{R}(E_f), \mathcal{R}(E_i))$ ) (dark purple) and consecutive encoders ( $\angle(\mathcal{R}(E_{t+1}), \mathcal{R}(E_t))$ ) (light purple). Average change in angle between the final and initial decoders ( $\angle(\mathcal{N}(D_f)^\perp, \mathcal{N}(D_i)^\perp)$ ) (dark orange) and consecutive decoders ( $\angle(\mathcal{N}(D_{t+1})^\perp, \mathcal{N}(D_t)^\perp)$ ) (light orange) ( $N = 14$ , centre shows the median; box shows the 25th–75th percentiles; whiskers extend to  $1.5 \times$  the interquartile range; two-sided Wilcoxon signed-rank test; (left)  $**P = 0.00012$ , (right)  $**P = 0.00012$ ). **f**, Average change in angle between final and initial encoders ( $\angle(\mathcal{R}(E_f), \mathcal{R}(E_i))$ ) (dark purple) and final and initial decoders ( $\angle(\mathcal{N}(D_f)^\perp, \mathcal{N}(D_i)^\perp)$ ) (dark orange) as a function of trial number ( $N = 14$ , median; shading shows the 25th–75th percentile). **g**, Average change in angle between consecutive encoders ( $\angle(\mathcal{R}(E_{t+1}), \mathcal{R}(E_t))$ ) (light purple) and consecutive decoders ( $\angle(\mathcal{N}(D_{t+1})^\perp, \mathcal{N}(D_t)^\perp)$ ) (light orange) over the course of a trial ( $N = 14$ , median; shading shows the 25th–75th percentile).

could influence the stationary point at which the system ultimately converges. This prediction is tested in Extended Data Fig. 5.

Setting learning rates of adaptive decoders has attracted interest in part because of its notable impact on the overall neural interface performance<sup>7,25,27,28,41</sup>. Our model allows us to make principled predictions that could be used to design and optimize adaptive algorithms. We assessed the convergence of equation (5) to the stationary points ( $E^*, D^*$ ) with  $E^*, D^* \neq 0$  by linearizing the dynamics about that point and then computing the eigenvalues of this linearization (chapter 5 of

ref. 57); the linearization was the same for both stationary points. We refer to the larger eigenvalue as the error decay rate, as it is the smallest number  $\rho > 0$  such that the error  $e(i)$  is bounded above by  $e(0)\rho^i$ , where  $i$  is the iteration number. Interestingly, this analysis predicts a Goldilocks effect of decoder learning rate on system convergence, consistent with past models<sup>25</sup>, for one regime of user learning; however, it predicts a monotonic relationship between system convergence and decoder learning rate for other user learning rate regimes (Fig. 3d,e). This complex landscape illustrates that empirical optimization is





**Fig. 3 | Game-theoretic model of co-adaptive user–machine systems.**

**a**, Schematic of co-adaptive interfaces in which the user and decoder are modelled as adapting to minimize their own individual cost functions (plots in the inset show data for each agent). **b**, Visualization of the potential function that describes dynamics within the user–decoder game model (simplified 1D user and decoder; Methods). Two axes represent the scalar values of the user and decoder actions, and the vertical axis represents the value of the potential function. A two-dimensional projection of the potential function is also shown. **c**, Gradient field of the user and decoder cost functions (with equal penalty terms,  $\lambda_D = \lambda_E$ ).

The purple (user) and orange (decoder) curves are nullclines (where the agent's gradient equals 0), which intersect at stationary points (black stars). **d**, Heat map showing the error decay rate as a function of learning rates in equation (5) with penalty parameters  $\lambda_D, \lambda_E = 1/4$ . **e**, Error decay rate as a function of decoder learning rate ( $\alpha_D$ ) for two values of encoder learning rate ( $\alpha_E = 3/2$  (blue) and  $\alpha_E = 1/2$  (red)) corresponding to the horizontal slices in **d**. **f**, Stationary values of scalar decoder (orange) and encoder (purple) as a function of decoder penalty parameter ( $\lambda_D$ ) relative to the value of encoder penalty ( $\lambda_E$ ).

challenging and demonstrates the value of our computational framework for theory-guided optimization. The prediction that learning rate affects error decay rate is tested in Fig. 4.

We then explored the contribution of penalty terms  $\lambda_E, \lambda_D$ , which influence user–decoder phase portraits (Fig. 5a). Altering these penalties shifts the stationary points ( $E^*, D^*$ ) and, thus, affects the user and decoder efforts,  $f_E$  and  $f_D$ . We find that an inverse relationship holds between  $E^*$  and  $D^*$  for a range of  $\lambda_E$  and  $\lambda_D$ : fixing  $\lambda_E$  and varying  $\lambda_D$  from 0 to  $1/4\lambda_E$ ,  $D^*$  decreases monotonically and  $E^*$  increases monotonically (Fig. 3f). This prediction is tested in Fig. 5.

To summarize, our game-theoretic model generates the following predictions about user–decoder co-adaptation:

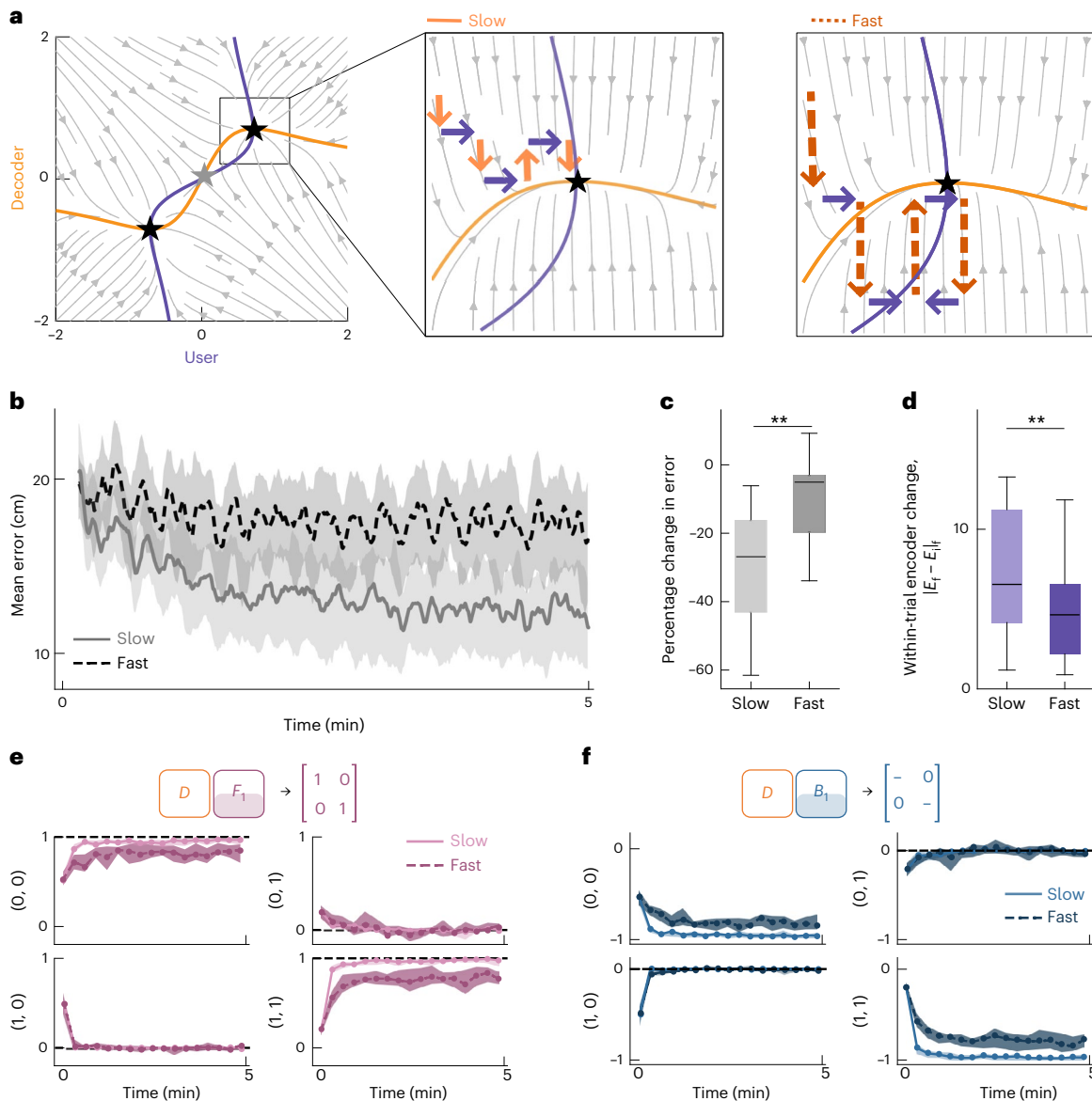
1. Overall system convergence and performance are influenced by both user and decoder learning rates.
2. Decoder penalty terms affect the location of stationary points in the overall system, which determines the user effort. This finding predicts that users and decoders, effectively, ‘trade-off’ effort to maintain system performance.
3. Because multiple stationary points exist, each learner's initialization will affect the final stationary point reached by the system. This predicts that decoder initialization will influence a user's final learned encoder.

### Decoder learning rates can disrupt co-adaptation

Our game-theoretic model emphasizes the importance of both user and decoder learning rates for convergence to stationary points. One agent adapting faster than the other may limit or prevent convergence to stationarity (Fig. 4a). Our model predictions echo results from previous models<sup>25</sup> and experimental findings from both invasive neural and non-invasive body–machine interfaces<sup>7,58</sup>. We examined the validity of

our model's predictions by testing the effect of decoder learning rates in our co-adaptive myoelectric interface system. Because the user's learning rate was unknown and could not be estimated a priori, we varied the decoder learning rate, testing two extremes: slow and fast (Methods). Our model makes two alternate predictions for the outcome of this manipulation, depending on user learning rate: if users learn relatively rapidly, these two extremes should yield similar performance; if users learn more slowly, fast decoder learning rates should reduce performance (Fig. 3e). Decoders for both learning rate conditions were initialized randomly, matching all other conditions, and then adapted according to the specified cost function. Task performance was worse for co-adaptive interfaces in which the decoder adapted with the fast learning rate compared with the slow learning rate throughout all trials (Fig. 4b), and performance improved more within the trial in the slow condition compared with the fast condition (Wilcoxon signed-rank test,  $P < 0.001$ ; Fig. 4c).

We found that the decoder learning rate also affected the user's encoders. We used our encoder estimates (Fig. 2) to quantify how much users' encoders changed over the course of the five-minute trial in each decoder condition (Methods). User encoders changed less within trials when the decoder adapted fast compared with slow (Wilcoxon signed-rank test,  $P < 0.001$ ; Fig. 4d), suggesting that the decoder may have adapted too fast for the user's encoder to match. We then analysed user–decoder interactions, which confirmed that rapid decoder adaptation disrupted co-adaptation. The encoder–decoder pairs in the slow condition approached the system ideal for perfect trajectory tracking and closed-loop stability, but the fast-condition encoder–decoder pairs deviated from the system's ideal state in both first-order feed-forward (Fig. 4e) and feedback conditions (Fig. 4f), showing that a fast learning rate negatively impacted the co-adaptive interactions for both trajectory tracking and closed-loop stability. Our



**Fig. 4 | Decoder learning rate influences interface performance and user learning.** **a**, Model predictions. All panels show the gradient field of the user and decoder cost functions. The purple (user) and orange (decoder) curves show nullclines (where the agent’s cost gradient equals 0) that intersect at stationary points (black stars). The slow (middle) and fast (right) panels zoom in on one stationary point and illustrate changes in the decoder (shades of orange) and encoder (purple) for different decoder learning rates. **b**, Average error as a function of time within the five-minute trial for fast (dashed dark orange) and slow (solid light orange) decoder learning rates ( $N = 14$ , median; shading shows the 25th–75th percentile). **c**, Percentage change in error from the start of the trial (first 30 s) to the end of the trial (last 30 s), separated by fast (dashed black) and slow (solid grey) learning rates ( $N = 14$ , centre shows the median; box shows the 25th–75th percentiles; whiskers extend to  $1.5 \times$  the interquartile range;

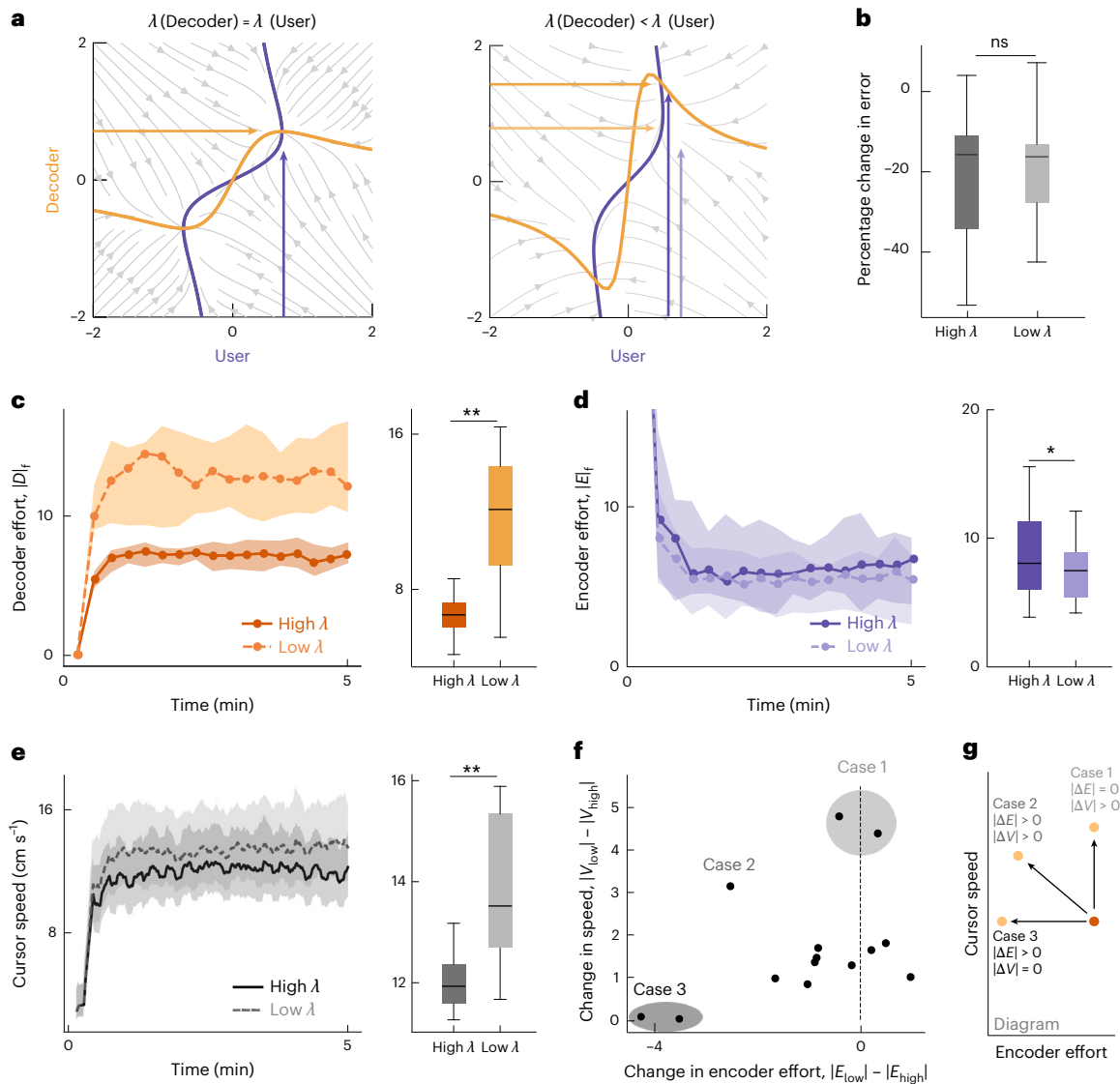
two-sided Wilcoxon signed-rank test,  $**P = 6.1 \times 10^{-5}$ ). **d**, Average change in the user’s encoder within trials ( $\|E_f - E_i\|$ ) for the slow and fast decoder learning rate conditions ( $N = 14$ , centre shows the median; box shows the 25th–75th percentiles; whiskers extend to  $1.5 \times$  the interquartile range; two-sided Wilcoxon signed-rank test,  $**P = 1.8 \times 10^{-4}$ ). **e**, Product of the decoder matrix with first-order feed-forward ( $F_1$ ) contributions of the encoder matrix ( $N = 14$ , median; shading shows the 25th–75th percentile) for fast (dashed dark pink) and slow (solid light pink) decoder learning rates. The black dashed lines represent values that yield perfect trajectory tracking. **f**, Product of the decoder matrix with first-order feedback ( $B_1$ ) contributions of the encoder matrix ( $N = 14$ , median; shading shows the 25th–75th percentile) for fast (dashed dark blue) and slow (solid light blue) decoder learning rates. The black dashed lines represent values that yield closed-loop stability.

experimental results corroborated our game-theoretic prediction that decoder learning rates influence the system behaviour, though precise validation experiments will require improved methods to estimate the user learning rates (see the ‘Discussion’ section).

### Decoder effort penalty influenced user effort without changing performance

Our game-theoretic framework includes the regularization terms,  $\|D\|$  and  $\|E\|$ , which correspond to what we refer to as ‘effort’. Including

regularization terms enforces convergence to a stable stationary point (see the ‘Game theory methods’ section). In the model, the scaling of the decoder regularization term, controlled by  $\lambda_D$ , influences the decoder effort. In turn, the user effort is also influenced by  $\lambda_D$  due to the linear relationship between the decoder and cursor velocity (equation (6)). As the decoder penalty changes in our model, the stationary point and encoder effort correspondingly shift (Fig. 5a). Our model, therefore, predicts that changing the decoder effort will not impact convergence to stationarity in the co-adaptive interface, but will influence



**Fig. 5 | Decoder penalty affects user effort and cursor velocity but not performance.** **a**, Model predictions. The panels show the gradient fields of the user and decoder cost functions (format as that in Fig. 4a) when the decoder penalty term is equal to (left) or less than (right). The arrows depict the expected stationary points of the user (purple) and decoder (orange). **b**, Percentage change in error from the trial start (first 30 s) to end (last 30 s), separated by high (dark grey) and low (light grey) decoder penalty term conditions ( $N = 14$ , median; shading shows the 25th–75th percentile; two-sided Wilcoxon signed-rank test,  $ns = 0.17$ ). **c**, Left: average magnitude of the decoder matrix (norm) over time in the trial for low (dashed light orange) and high (solid dark orange) decoder penalty terms ( $N = 14$ , median; shading shows the 25th–75th percentile). Right: box plots are average decoder efforts across the trial for each subject ( $N = 14$ , centre shows the median; box shows the 25th–75th percentiles; whiskers extend to  $1.5 \times$  the interquartile range; one-sided Wilcoxon signed-rank test,  $**P = 6.1 \times 10^{-5}$ ). **d**, Left: average magnitude of the user encoder matrix (norm) over time in the trial for low (dashed light purple) and high (solid dark purple) decoder penalty terms ( $N = 14$ , median; shading shows the 25th–75th percentile).

Right: box plots are average effort for each subject across the trial ( $N = 14$ , centre shows the median; box shows the 25th–75th percentiles; whiskers extend to  $1.5 \times$  the interquartile range; one-sided Wilcoxon signed-rank test,  $*P = 0.018$ ). **e**, Left: average cursor speed as a function of time in the trial for low (dashed light grey) and high (solid dark grey) decoder penalty terms ( $N = 14$ , median; shading shows the 25th–75th percentile). Right: box plots are the average cursor speed across the trial for each subject ( $N = 14$ , centre shows the median; box shows the 25th–75th percentiles; whiskers extend to  $1.5 \times$  the interquartile range; one-sided Wilcoxon signed-rank test,  $**P = 6.1 \times 10^{-5}$ ). **f**, Average difference in cursor speed between the low and high decoder penalty conditions plotted against the average difference in encoder efforts between the low and high decoder penalty conditions for each individual participant (black dots). The shading illustrates qualitatively identified groupings. **g**, Schematic illustrating the relationship between encoder effort and cursor speed when the decoder effort changes. The dots represent the encoder effort and velocity for high (dark orange) and low (light orange) decoder penalty terms. The arrows depict the potential scenarios of how the decoder penalty can shift the encoder effort and velocity.

user behaviour by shifting their effort. Past work suggests that users can adapt their encoder to trade off with a decoder in a co-adaptive interface in which the decoder only prioritized optimizing task performance<sup>10</sup>. Yet, many studies highlight limitations on the degree of user flexibility in a range of neural interfaces<sup>59,60</sup>. It is, therefore, unknown whether users will flexibly trade off with a decoder that aims to optimize multiple objectives, including its own individual effort.

We tested the impact of two penalty terms—low penalty and high penalty—on encoder–decoder co-adaptation in our myoelectric interface experiments. Decoders for both penalty conditions were initialized randomly, matching all other conditions, and then adapted according to the specified cost function (Methods). Performance was not impacted by the penalty terms (Wilcoxon signed-rank test,  $P > 0.05$ ; Fig. 5b), suggesting that the co-adaptive systems converged

similarly for both decoder conditions. Decoder effort was influenced by the penalty term as expected: higher decoder penalty resulted in decreased decoder effort (Wilcoxon signed-rank test,  $P < 0.001$ ; Fig. 5c). As predicted by our model, the decoder effort affected the encoder effort to reflect the trade-off in effort between the two learners. Thus, higher decoder penalty terms resulted in increases in the encoder effort (quantified by the norm of the  $F_1$  component of the encoder; Wilcoxon signed-rank test,  $P < 0.05$ ; Fig. 5d). These results held true regardless of the learning rate of the decoders (Extended Data Fig. 6).

Although our experimental results broadly matched model predictions, we noticed that the relative decrease in encoder magnitude with respect to the increase in decoder magnitude was smaller than predicted by the theory (Fig. 5c versus Fig. 5d). Importantly, decoder effort controls the magnitude of  $D$ , which defines the gain between the user EMG and cursor velocity. If users do not change their effort to perfectly mirror changes in the decoder effort, the cursor's speed—an aspect of task performance not captured in our model and not directly linked to task performance—would change. We found that the decoder penalty terms influenced the cursor speed during tracking, with a lower decoder penalty term leading to faster cursor speeds, on average, across users (Wilcoxon signed-rank test,  $P < 0.001$ ; Fig. 5e). Interestingly, these trade-offs differed across our participants (Fig. 5f), potentially capturing individual preferences. We noticed clear outlier participants who chose to decrease their effort and maintain a similar cursor speed, or alternately maintain a similar effort to go faster; the remainder of our participants changed both effort and speed to varying degrees. This suggests three broad groups of strategies adopted by users (Fig. 5g). These different strategies, however, did not correlate with the task performance, suggesting that users may identify compromise solutions to trade off with the decoder without sacrificing task performance. These findings highlight experimental deviations from our model predictions for strict encoder–decoder trade-offs. We also found only subtle evidence to support our model's predictions related to the influence of decoder initialization (prediction 3 from the preceding section, 'Game theory model generated predictions for co-adaptation outcomes'; Extended data Fig. 5; see the 'Discussion' section). Despite these deviations, our experimental results broadly confirmed our game-theoretic model predictions and highlighted the potential of decoder adaptation to influence the user's encoder during closed-loop co-adaptation.

## Discussion

We extended quantitative modelling techniques from control theory and game theory to reveal how decoding algorithms can shape user adaptation in neural interfaces. The critical first step was developing a myoelectric interface testbed that enabled systematic experiments, examining how users learn to control an unfamiliar interface alongside different adaptive algorithms. Our platform provides advantages over existing human-in-the-loop simulators and related approaches, which may not fully capture how people respond to adaptive algorithms in closed-loop settings<sup>61–65</sup>.

We then created control-theory-based estimations of user encoder transformations that allowed us to quantify critical closed-loop properties like trajectory tracking and stability (Fig. 2c,d) and illustrated how decoder design impacted these properties (Fig. 4e,f). Prior work has shown that encoder–decoder stability probably contributes to interface performance and user learning<sup>2,7,10</sup>, but efforts to quantify stability in adaptive neural interfaces have focused on the decoder alone<sup>27,28</sup>. By developing methods to analyse encoder–decoder pairs, we were able to characterize stability in the full co-adaptive system (Figs. 2 and 4e,f) and reveal how decoding algorithms can influence user adaptation. Although our approach used a simplified model of user closed-loop control policies, our results highlight the promise of these frameworks for designing and analysing neural interface systems. Expanding the accuracy of our models, by incorporating details like sensory processing delays, will probably lead to further advances.

We also developed a game-theoretic model to generate predictions about the behaviour of different co-adaptive systems. Designing co-adaptive interfaces requires confronting a vast space of algorithm parameters that can influence decoder and interface performance<sup>7,25,27,41,66</sup>. Our approach provides experimentally validated methods to quantitatively model and predict outcomes of co-adaptive dynamics. These tools will enable principled interface design and optimization by considering multiple objectives and user–decoder interactions. Current neural interfaces largely aim to maximize performance on a task without considering other aspects of interface performance, such as stability or robustness. Our results demonstrate an example of training algorithms to consider multiple objectives, balancing task error with decoder effort, which, in turn, influences user effort (Fig. 5). Future interfaces might be designed to consider additional objectives like stability, robustness or performance across multiple tasks. We foresee our game-theoretic optimization-based framework becoming especially useful as the functionalities of neural interface tasks grow.

Our game-theoretic model is particularly valuable for predicting user–decoder interactions that influence system performance, such as decoder learning rates (Fig. 3d–f). We found that a faster decoder learning rate yielded worse overall performance (Fig. 4c) and altered trajectory tracking and stability in encoder–decoder pairs (Fig. 4e,f) in our experimental myoelectric interface. Our results add to prior experimental observations from co-adaptive kinematic<sup>7,25</sup> and brain–computer interfaces<sup>25,58</sup>. Interestingly, many high-performance adaptive algorithms in invasive neural interfaces adapt rapidly<sup>13,28,67</sup>. Other co-adaptive interface models and experiments suggest that intermediate learning rates are optimal<sup>25</sup>. We found that a fast decoder learning rate condition disrupted co-adaptive dynamics in our interface, and our data suggest a more monotonic relationship between decoder learning rates and system performance. Our model highlights that the learning rates of both user and decoder determine system dynamics (Fig. 3d–f) and our experimental data are the most consistent with a regime in which users learned slowly. Although we cannot yet verify user learning rates, it is plausible that our naïve participants may have adapted at different rates than the more experienced participants in invasive brain–computer interface studies and participants using well-learned input interfaces (for example, a computer mouse) in a co-adaptive system<sup>25</sup>. Indeed, user learning rates may vary across different biosignal modalities<sup>20</sup>. Our framework provides a path to optimize neural interfaces by tailoring the algorithm adaptation rates based on user behaviour. This raises important future challenges to develop methods that can perform on-the-fly estimates of critical properties of user behaviour, such as learning rate.

Our game theory model and experimental results also revealed user–decoder interactions that influence user behaviour, such as decoder effort (Fig. 5d). This demonstrates the feasibility of using our framework to optimize interfaces for particular goals. In myoelectric interfaces, manipulating decoders to encourage users to learn encoders that use less muscle activity could be desirable to maximize interface usability for extended periods. Alternately, decoders that encourage users to increase muscle activation towards a desired target could be desirable for motor rehabilitation applications<sup>23,68</sup>. Interestingly, we saw that users responded differently to varying decoder penalty terms, potentially reflecting personal preferences for effort versus cursor speed (Fig. 5f). Customizing interfaces based on dynamic user preferences has been shown to be beneficial in exoskeleton interfaces<sup>69</sup>. Individual preferences for cursor dynamics have also been seen in invasive neural interfaces<sup>70</sup>. Extending our game-theoretic frameworks to include user preferences will enable the design of fully personalized, custom neural interfaces, which may make these technologies more widely accessible<sup>71</sup>.

Our game theory model highlighted that multiple encoder–decoder stationary points exist in neural interfaces. This raises the possibility that the initial decoder can affect the final stationary point

and, in turn, the final user encoder, without impacting performance (Extended Data Fig. 5a). Our experimental results confirmed that performance was not impacted by decoder initialization (Fig. 1e), consistent with prior work testing random decoder initializations in neural interfaces<sup>41</sup>. Although performance was not different, we found that initialization subtly impacted the stationary points as identified by the final encoder–decoder matrices (Extended Data Fig. 5e). Our findings suggested that decoder initialization might bias the encoders learned by users, which could be particularly beneficial for rehabilitation applications, which aim to shape users' behaviour towards a particular goal. Our results also suggest that it may be critical to carefully consider initial training protocols for any neural interface, as it may influence user learning trajectories.

Our experiments revealed occasional deviations from model predictions, including deviations from strict encoder–decoder trade-offs (Fig. 5f) and only subtle impacts of the decoder initialization (Extended Data Fig. 5). This finding suggests that the actual user's cost might differ from our model's assumptions. For instance, our model predicts that decoder initialization would lead users to completely invert the relationship between EMG/movement and cursor velocity (Extended Data Fig. 5a), which ignores the likely possibility that users have some bias in strategies to control the interface. Additionally, there may be multiple ways to represent effort, and these definitions may also become unclear in less embodied invasive neural interfaces. Prior work has considered that users minimize effort<sup>54,72</sup> via muscle activation<sup>73</sup>, metabolic cost<sup>74,75</sup> and motor variability<sup>76</sup>. Studies into how users learn to control neural interfaces suggest that other features, such as correlations between neurons or muscles, may be key factors<sup>59,60</sup>. Deeper insights into how users learn to control different kinds of interface will improve our ability to model their cost functions to predict user–decoder interactions.

By developing computational and experimental methods to study co-adaptive systems, our results reveal how interface algorithms can influence user behaviour and system performance. Specifically, our model and experimental findings show that the decoder learning rate, penalty terms and initialization influence encoder–decoder dynamics in co-adaptive interfaces. These observations are consistent with recent work suggesting that adaptive algorithms can influence neural representations learned in brain–computer interfaces<sup>77</sup>. Our framework provides paths to harness the complex interactions between users and algorithms in closed-loop interfaces. For example, our results hint at the eventual possibility of designing a 'curriculum' to help naive users learn to control neural interfaces, which is a task that some find challenging<sup>48</sup>. Decoder learning rates may need to be slower early in learning when users widely explore control strategies<sup>78–80</sup> and adapt as users master the interface. Strategic decoder manipulations during a user's initial exploration could be designed to bias users towards particular strategies. We can borrow concepts from training neural networks<sup>81</sup> to envision a decoder adaptation curriculum to intelligently shape user learning, fully harnessing the power of co-adaptive interfaces. Our framework provides a versatile and principled way to model and design the next generation of smart, personalized adaptive neural interfaces.

## Methods

### Experimental methods

**Participants.** Fourteen volunteers were recruited for this study and gave their written consent before experiments, according to study procedures approved by the University of Washington's Institutional Review Board (IRB number STUDY00014060). All participants were compensated monetarily for their time. All participants had no known motor disorders. Participant demographic information (including gender, weight, height, age and handedness) was collected via a demographics survey that participants completed before the experiment (Table 1). Only forearm circumference was measured by the experimenter; all other information was self-reported.

**Experimental design.** Participants were asked to control a cursor on the screen—using muscle activity from their forearms measured via EMG—to follow a two-dimensional continuous target trajectory as closely as possible (Extended Data Fig. 1a). Participants were told that they might not be able to control the cursor at the beginning of the trial but to expect that their cursor control would improve as the trial progressed. Participants received no other cues or information about the decoder or study conditions. If the participant's cursor was stuck in the corner or edge of the screen for longer than 3.33 s, the cursor was automatically reset to the centre of the screen. Participants sat in a chair with no restraints facing a computer screen (HP Compaq L2206tm, 1,900 × 1,600 pixels, 46.5 cm × 24.5 cm). The computer screen displayed a red target circle (RGB: 1, 0, 0). The cursor was a blue circle (RGB: 0, 0, 1). The target was three times the area of the cursor (Extended Data Fig. 1a). The target and cursor positions were updated and displayed at 60 Hz. The task was programmed using the `pygame` version 2.0.1 and `LabGraph` version 1.0.2 Python packages.

Following prior work<sup>39,40,49</sup>, the target traversed a pseudorandom trajectory, which was generated by a sum of sinusoids with randomized phases, with frequencies that were prime multiples of 0.05 Hz (*x*-axis frequencies, 0.10 and 0.25 Hz; *y*-axis frequencies, 0.15 and 0.35 Hz). We randomized the phase of the sum of sines so that the reference trajectory would be unpredictable to the participants. Distinct prime multiples were chosen in each direction to provide separability in the *x* and *y* axes for any frequency-based analysis. To ensure constant signal power, the magnitude of each frequency component was normalized by the frequency squared. The trajectory was different for every trial. Each trial was 5 min with a 5-s ramp period during which the cursor and target speed slowly increased from stationary to the experimentally prescribed speeds. This ramp period followed prior experiments<sup>49</sup> and was added to give participants time to recognize the starting cursor and target movements. Participants completed 16 five-minute trials in two blocks; each block consisted of eight trials. Participants were given a five-minute break in between each block but their EMG array and placement did not change throughout the experiment.

**EMG signal collection and preprocessing.** EMG signals were obtained using a Quattrocento system (Bioelettronica). A 64-channel high-density surface EMG electrode array (8-mm interelectrode spacing, 5 × 13 electrodes in a rectangular layout) was placed on the dominant forearm of each participant, targeting the extensor carpi radialis (Extended Data Fig. 1b). Electrodes were placed on the dominant arm for each participant. Once placed, the electrode array was wrapped with Coban self-adherent wrap (3M). The electrode cables from Quattrocento to the array were secured to minimize motion artefacts.

EMG signals were acquired using Biolite Software (Bioelettronica) at 2,048 Hz in the differential mode with the built-in low-pass (130 Hz) and high-pass (10 Hz) filters. We filtered and rectified the EMG data following prior preprocessing techniques to compute the EMG linear envelope<sup>82,83</sup>. A moving average filter was applied to downsample EMG signals from 2,048 Hz to 60 Hz (Extended Data Fig. 1c).

**Myoelectric interface decoder.** Real-time myoelectric control was implemented using a velocity-controlled Wiener filter that output a cursor velocity  $v_t \in \mathbb{R}^2$ :

$$v_t = D \cdot u_t, \quad (6)$$

where  $u_t \in \mathbb{R}^N$  is the processed EMG signals of  $N = 64$  channels and  $D \in \mathbb{R}^{2 \times N}$  is the decoder mapping. The cursor velocity is integrated to output a cursor position  $y_t \in \mathbb{R}^2$ :

$$y_t = y_{t-1} + v_t \Delta t, \quad (7)$$

where  $\Delta t$  is the time between cursor updates.

**Myoelectric interface decoder adaptation.** Decoder adaptation consisted of two steps: (1) calculate the optimal decoder  $D^*$  by minimizing the cost function  $c_D$  based on the previous 20 s of user and trial data:

$$D^* = \arg \min_D c_D(D); \tag{8}$$

(2) update the next decoder  $D$  following the SmoothBatch approach<sup>44</sup>, which uses a weighted combination of the prior decoder  $D^-$  and the optimal decoder  $D^*$ :

$$D = \alpha D^- + (1 - \alpha) D^*, \tag{9}$$

where  $\alpha \in (0, 1)$  is the learning rate. The decoder cost function was minimized using SciPy version 1.5.4 (ref. 84). To satisfy real-time timing constraints on decoder cost minimization, we initialized  $D_0^* = (\tau - y) \cdot u^\dagger$  using the previous 20 s of data<sup>26</sup>. At the start of each trial, the decoder was initialized by randomizing the decoder weightings from a uniform distribution (using `numpy.random.rand`) and multiplying by a scalar factor. Each user had two different decoder initializations, referred to as D1 and D2. Initial decoders for each participant’s trial were set to D1 or D2. Note, although we programmed the decoder to update every 20 s, due to slight software imprecision, the decoder update actually occurred approximately every 18 s.

**Decoder cost.** The decoder cost function was formulated in earlier work<sup>55</sup> and aims to minimize both tracking error and decoder effort as the decoder adapts. Minimizing velocity error is a common goal in user–machine interfaces, seen widely across neural interfaces<sup>10,85</sup>, body–machine interfaces<sup>86</sup> and myoelectric interfaces<sup>44,66</sup>. Decoder effort is considered as part of the decoder cost since our prior theoretical analysis suggested that a regularization term in the user and decoder costs is necessary to ensure convergence in the user–decoder co-adaptation game to stable stationary points<sup>55</sup>.

The decoder cost  $c_D$  is constructed as a linear combination of the task error and decoder effort:

$$\text{error} = \left\| D \cdot u - \frac{1}{\Delta t} (\tau - y) \right\|_2^2, \text{ effort} = \|D\|^2; \tag{10}$$

here  $\|x\|_2$  denotes the 2-norm of signal  $x : [0, t] \rightarrow \mathbb{R}^d$ , and  $\|X\|$  denotes the Frobenius norm of matrix  $X \in \mathbb{R}^{m \times n}$ . The decoder cost  $c_D$  is then

$$c_D = \left\| D \cdot u - \frac{1}{\Delta t} (\tau - y) \right\|_2^2 + \lambda_D \|D\|^2, \tag{11}$$

where  $\lambda_D$  is the penalty term of the decoder effort.

**Decoder conditions.** We varied the decoder cost function and parameters of decoder adaptation to determine how it influenced system performance and user behaviour. Specifically, we varied the (1) learning rate  $\alpha$ , (2) decoder effort penalty term  $\lambda_D$  and (3) decoder initialization. We tested two learning rates, namely, slow ( $\alpha = 0.75$ ) and fast ( $\alpha = 0.25$ ); two penalty terms, that is, low ( $\lambda = 10^2$ ) and high ( $\lambda = 10^3$ ); and two randomized initializations of the  $D$  decoder matrix (D1 and D2), namely, positive (matrix elements chosen uniformly at random in the range  $[0, 10^{-2}]$ ) and negative (elements chosen uniformly at random in  $[-10^{-2}, 0]$ ). We tested all combinations of learning rates, penalty terms and initializations, leading to a total of eight different decoder conditions.

**Data analysis methods**

**Performance metrics.** Our primary metric for quantifying performance was the tracking error, calculated as the Euclidean distance between the target and the cursor:  $\|\tau - y\|_2$ . We assessed changes in task performance within a trial by comparing the mean tracking error in the first (early) and last (late) 30 s of the trial (excluding the ramp-up

time). Because participants’ proficiency in tracking varied, we quantified improvements over time by calculating the relative error:  $\frac{\text{error}_{\text{fin}}}{\text{error}_{\text{ini}}} \times 100\%$ , where  $\text{error}_{\text{fin}} = \text{error}_{\text{final}} - \text{error}_{\text{initial}}$ .

**Statistical analyses.** All analyses treated participants as individual data points and computed the mean across decoder conditions (learning rate, penalty terms, initialization) and trials. To assess statistical significance across time and conditions for each participant, we used a one- or two-sided Wilcoxon signed-rank test (`scipy.stats.wilcoxon`), which is a paired, non-parametric test. We chose a non-parametric statistical test because of subject-to-subject variability. Figure box plots were plotted using `matplotlib.pyplot.boxplot` (centre line, median; box limits, upper and lower quartiles; whiskers, 1.5× the inter-quartile range; fliers not plotted).

**EMG analysis.** To quantify the relationship between user EMG activity and cursor movement, we computed the EMG direction tuning curves, a well-established analysis used for neural and myoelectric interfaces<sup>43–45</sup>. We first downsampled and grouped intended user cursor velocities into ten equally spaced directions and then fit raw EMG activity into the binned directions to form EMG tuning curves. These EMG tuning curves were created for each participant, each trial and each EMG signal (Fig. 1f). Note, since the EMG channels are differentially recorded, we selected 63 EMG signals for the EMG tuning analysis.

To quantify the within-trial EMG changes from early to late trial segments (Fig. 1g) we computed the norm difference between the early and late EMG tuning curves:

$$\|\Delta \text{EMG}\| = \|\text{EMG}_{\text{dir},i} - \text{EMG}_{\text{dir},j}\|, \tag{12}$$

where  $\text{EMG}_{\text{dir}}$  is the average EMG activity per direction and  $i$  and  $j$  indicate time points. All tuning curves were calculated by reconstructing 30-s segments of trial data that equally sampled all cursor directions. To measure how the EMG activity changed across the entire experiment per participant (Fig. 1h), we averaged the  $\|\Delta \text{EMG}\|_2$  across all channels for each participant to compute a mean  $\|\Delta \text{EMG}\|_2$  per participant per trial.

To find the preferred direction of the EMG activity, we performed cosine-fitting analysis<sup>43</sup>. Specifically, average EMG activity for each intended cursor direction was modelled as

$$\text{EMG} = B_1 \cos \theta + B_2 \sin \theta + B_3, \tag{13}$$

where  $\theta$  represents the intended cursor direction and  $B_1, B_2$  and  $B_3$  are model coefficients. The preferred direction (PD) is

$$\text{PD} = \arctan 2(B_2/B_1). \tag{14}$$

The change between early and late preferred directions (Fig. 1i) was calculated as  $|\Delta \text{PD}| = |\text{PD}_{\text{late}} - \text{PD}_{\text{early}}|$ .

**Encoder estimation methods**

To model the user’s transformation from task space to user signals, we estimated the user’s encoder  $E$ . We model the user’s signal as being produced by a combination of feed-forward ( $F$ ) control on position ( $\tau$ ) and velocity ( $\dot{\tau}$ ) data, as well as feedback ( $B$ ) control on position error and velocity error data ( $(\tau - y)$  and  $(\dot{\tau} - \dot{y})$ , respectively):

$$u = F_0 \cdot \tau + F_1 \cdot \dot{\tau} + B_0 \cdot (\tau - y) + B_1 \cdot (\dot{\tau} - \dot{y}). \tag{15}$$

Or, in the matrix formulation,

$$u = E \cdot P + \beta \cdot 1_t^T = [F_0 \ F_1 \ B_0 \ B_1] \cdot \begin{bmatrix} \tau \\ \dot{\tau} \\ \tau - y \\ \dot{\tau} - \dot{y} \end{bmatrix} + \beta \cdot 1_t^T, \tag{16}$$

where  $\mathbf{1}_t^T = [1 \dots 1] \in \mathbb{R}^{1 \times t}$ ,  $u \in \mathbb{R}^{N \times t}$ ,  $E \in \mathbb{R}^{N \times d}$ ,  $P \in \mathbb{R}^{4d \times t}$ ,  $N$  is the number of signals,  $d$  is the dimension of task information and  $\beta$  is an offset factor for estimation ( $\beta \in \mathbb{R}^{N \times 1}$ ). In this Article,  $N = 64$  EMG channel inputs,  $d = 2$  ( $x$  and  $y$  dimensions on a computer display). The selection of  $t$  determines the timescale on which  $E$  is estimated.

We numerically estimated users encoders  $E$  using linear regression of experimentally measured data for  $u$  and  $P$ . We estimated  $E$  every 20 s. This corresponds to the timescale on which decoders were updated, thereby generating matched encoder–decoder estimate pairs to analyse co-adaptive dynamics (for example, Fig. 2c,d). Linear regression was done by constructing a  $u$  matrix of EMG signals for  $t = 1, 200$  samples (20 s of data at 60 Hz) for the 64 EMG electrodes, and a corresponding matrix  $P$  of position, velocity, position error and velocity error ( $\tau \in \mathbb{R}^{2 \times t}$ ,  $\dot{\tau}$ ,  $\tau - y$  and  $\dot{\tau} - \dot{y}$ , respectively). These data were used to fit equation (16) using linear regression (`sklearn.linear_model.LinearRegression`) with an intercept offset.

We estimated the accuracy of the myoelectric encoder model by reconstructing EMG signals from the encoder estimation and compared the coefficient of determination ( $R^2$ ) for the cursor velocity and position decoded from the reconstructed EMG to the actual cursor velocity and position recorded in the trial. To establish a baseline for the predictive power of our encoder model, we also computed the accuracy of cursor velocity and position decoded from time-shuffled EMG data (Fig. 2b). The time-shuffled EMG data were created by permuting the EMG signals within each trial, so that the EMG data came from the same subject and trial but different time points.

**Closed-loop encoder–decoder predictions.** We used control theory principles to predict relationships between elements of encoder  $E$  and decoder  $D$ . The decoder is velocity based, meaning that the cursor position  $y$  is related to the user’s EMG signal  $u$  via  $\dot{y} = D \cdot u$ , where  $D$  is the decoder matrix. We model the user’s signal as being produced by a combination of feed-forward and feedback control on the position and velocity data:

$$u = F_0 \cdot \tau + F_1 \cdot \dot{\tau} + B_0 \cdot (\tau - y) + B_1 \cdot (\dot{\tau} - \dot{y}). \tag{17}$$

This controller can achieve exact tracking, defined as  $y = \tau$ , if  $D \cdot F_0 = 0$  and  $D \cdot F_1 = I$ , where  $0$  is the  $2 \times 2$  zero matrix and  $I$  is the  $2 \times 2$  identity matrix. To see this, suppose at time  $t$  that  $\tau(t) = y(t)$ . Then,

$$\dot{\tau}(t) = \dot{y}(t) = D \cdot u(t) = D \cdot F_0 \cdot \tau(t) + D \cdot F_1 \cdot \dot{\tau}(t) \iff D \cdot F_0 = 0, D \cdot F_1 = I. \tag{18}$$

Note that  $M$  is a simple integrator, so the condition for  $\dot{\tau}(t) = \dot{y}(t)$  in equation (18) together with  $\tau(t) = y(t)$  ensures  $\tau = y$  at all times. To ensure closed-loop stability, it suffices that  $D \cdot B_0$  and  $D \cdot B_1$  are negative-definite diagonal matrices (chapter 5 of ref. 51).

**Encoder–decoder pairs.** We calculated encoder–decoder pairs by multiplying the decoder ( $D \in \mathbb{R}^{2 \times 64}$ ) at each update with the feed-forward and feedback contributions to the encoder ( $E \in \mathbb{R}^{64 \times 8}$ ) at each update. Each encoder–decoder pair ( $D \cdot F_0, D \cdot F_1, D \cdot B_0$  and  $D \cdot B_1$ ) resulted in a  $2 \times 2$  matrix representing the dynamics between the encoder and decoder.

**Encoder differences.** We calculated changes in encoders and decoders over time using the principal angle between subspaces (range space of  $E$  and orthogonal complement to the null space of  $D$ )<sup>87</sup> and the norm difference of encoder matrices:  $\|E_i - E_j\|$ , where  $i$  and  $j$  indicate the time points. Since we started from a decoder that was initialized with randomized weightings and resulted in poor user control, the details of the first encoder  $E_0$  were not readily interpretable. We did not include  $E_0$  in computing or plotting the encoder differences in Fig. 2e–g. To compute encoder changes within a trial, we calculated the difference between

the final ( $E_f$ ) and initial ( $E_i$ ) encoders, estimated by averaging the first three and last three encoders in a trial, respectively (excluding  $E_0$ ). To compare data from consecutive encoders to the average change across a full trial (Fig. 2g), we computed the consecutive encoder difference by averaging across three subsequent, non-overlapping encoders and calculating the norm difference. This method ensured that the same number of encoder data points would be compared.

**Game theory methods**

We modelled the co-adaptive system in a game theory framework involving two agents: the user and the decoder. The decision variables of the user and decoder were the entries in the matrices  $E$  and  $D$ , respectively. For tractability, we limit our analysis to the scalar case  $E, D \in \mathbb{R}$ . Each agent was modelled as having a cost function defined by a linear combination of task error and agent effort. Agent effort was quantified using the square of the scalars  $E$  and  $D$ . Task error was defined as in equation (10) as the norm squared of  $D \cdot u - (\tau - y)/\Delta t$ . Considering the scalar case and focusing solely on feedback control (that is, setting  $u = B_0(\tau - y)$ ) yields error  $(DB_0\Delta t - 1)^2(\tau - y)^2/\Delta t$ . For simplicity in the analysis, we relabel  $E = B_0$ , normalize by  $(\tau - y)^2/\Delta t$ , and choose coordinates for  $D$  so that  $e(D, E) = (DE - 1)^2 = (1 - DE)^2$ .

**Stationary points.** In the one-dimensional game formulation, the potential function takes the form

$$\phi(E, D) = (1 - DE)^2 + \lambda_E E^2 + \lambda_D D^2, \tag{19}$$

where  $D, E \in \mathbb{R}$ . We solved the nonlinear equations  $\partial_E \phi(E, D) = 0$  and  $\partial_D \phi(E, D) = 0$  to compute stationary points as a function of penalty parameters  $E^*(\lambda_E, \lambda_D)$ ,  $D^*(\lambda_E, \lambda_D)$ , finding a saddle at  $E^* = D^* = 0$  and the following mirror pair of local minima:

$$E^*(\lambda_E, \lambda_D) = \pm \sqrt{\sqrt{\lambda_D/\lambda_E} - \lambda_D}, D^*(\lambda_E, \lambda_D) = \pm \sqrt{\sqrt{\lambda_E/\lambda_D} - \lambda_E}. \tag{20}$$

That these points are local minimizers was verified for  $\lambda_E, \lambda_D \in (0, 1]$  by numerically determining that the eigenvalues of the Hessian of the potential function  $\phi$  were positive at these points.

**Model for adaptation.** We modelled agent adaptation using the discrete-time dynamics in equation (5). To assess the convergence of this model for adaptation to stationary points, we linearize the discrete-time dynamics about a stationary point and compute the eigenvalues of this linearization. A sufficient condition for stability is that the magnitude of all eigenvalues is less than 1.

**Calculation of error decay rate.** Given a matrix  $A$  that is the linearization of the discrete-time gradient descent dynamics in equation (5), we compute the spectral radius as  $\rho = \max\{|\lambda|\}$ , where  $\lambda$  is an eigenvalue of  $A$ . Then, the error decay rate is equal to the spectral radius  $\rho$ , and satisfies the bound  $e(i) \leq e(0)\rho^i$ , where  $i$  is the iteration number.

**Reporting summary**

Further information on research design is available in the Nature Portfolio Reporting Summary linked to this article.

**Data availability**

All data are publicly available in a Code Ocean capsule at <https://doi.org/10.24433/CO.4049054.v3> (ref. 88).

**Code availability**

The data and analysis scripts needed to reproduce all figures and statistical results reported in the Article are publicly available via CodeOcean at <https://doi.org/10.24433/CO.4049054.v3> (ref. 88).

## References

1. 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).
2. Taylor, D. M., Tillery, S. I. H. & Schwartz, A. B. Direct cortical control of 3D neuroprosthetic devices. *Science* **296**, 1829–1832 (2002).
3. Carmena, J. M. et al. Learning to control a brain-machine interface for reaching and grasping by primates. *PLoS Biol.* **1**, e42 (2003).
4. Hochberg, L. R. et al. Neuronal ensemble control of prosthetic devices by a human with tetraplegia. *Nature* **442**, 164–171 (2006).
5. Pandarinath, C. & Bensmaia, S. J. The science and engineering behind sensitized brain-controlled bionic hands. *Physiol. Rev.* **102**, 551–604 (2022).
6. Dadarlat, M. C., Canfield, R. A. & Orsborn, A. L. Neural plasticity in sensorimotor brain-machine interfaces. *Annu. Rev. Biomed. Eng.* **25**, 51–76 (2023).
7. Danziger, Z., Fishbach, A. & Mussa-Ivaldi, F. A. Learning algorithms for human-machine interfaces. *IEEE Trans. Biomed. Eng.* **56**, 1502–1511 (2009).
8. DiGiovanna, J., Mahmoudi, B., Fortes, J., Principe, J. C. & Sanchez, J. C. Coadaptive brain-machine interface via reinforcement learning. *IEEE Trans. Biomed. Eng.* **56**, 54–64 (2009).
9. Jarrassé, N., Charalambous, T. & Burdet, E. A framework to describe, analyze and generate interactive motor behaviors. *PLoS ONE* **7**, e49945 (2012).
10. Orsborn, A. L. et al. Closed-loop decoder adaptation shapes neural plasticity for skillful neuroprosthetic control. *Neuron* **82**, 1380–1393 (2014).
11. Shenoy, K. V. & Carmena, J. M. Combining decoder design and neural adaptation in brain-machine interfaces. *Neuron* **84**, 665–680 (2014).
12. Hahne, J. M., Markovic, M. & Farina, D. User adaptation in myoelectric man-machine interfaces. *Sci. Rep.* **7**, 4437 (2017).
13. Brandman, D. M. et al. Rapid calibration of an intracortical brain-computer interface for people with tetraplegia. *J. Neural Eng.* **15**, 026007 (2018).
14. Silversmith, D. B. et al. Plug-and-play control of a brain-computer interface through neural map stabilization. *Nat. Biotechnol.* **39**, 326–335 (2021).
15. Rizzoglio, F., Casadio, M., De Santis, D. & Mussa-Ivaldi, F. A. Building an adaptive interface via unsupervised tracking of latent manifolds. *Neural Netw.* **137**, 174–187 (2021).
16. Gigli, A., Gijssberts, A. & Castellini, C. Unsupervised myocontrol of a virtual hand based on a coadaptive abstract motor mapping. In *Proc. 2022 International Conference on Rehabilitation Robotics (ICORR)* 1–6 (IEEE, 2022).
17. Hu, X. et al. Bridging human-robot co-adaptation via biofeedback for continuous myoelectric control. *IEEE Robot. Autom. Lett.* **8**, 8573–8580 (2023).
18. Ganguly, K. & Carmena, J. M. Emergence of a stable cortical map for neuroprosthetic control. *PLoS Biol.* **7**, e1000153 (2009).
19. Fetz, E. E. Volitional control of neural activity: implications for brain-computer interfaces. *J. Physiol.* **579**, 571–579 (2007).
20. Jackson, A. & Fetz, E. E. Interfacing with the computational brain. *IEEE Trans. Neural Syst. Rehabil. Eng.* **19**, 534–541 (2011).
21. Albert, S. T. & Shadmehr, R. The neural feedback response to error as a teaching signal for the motor learning system. *J. Neurosci.* **36**, 4832–4845 (2016).
22. Oby, E. R. et al. New neural activity patterns emerge with long-term learning. *Proc. Natl Acad. Sci. USA* **116**, 15210–15215 (2019).
23. De Santis, D. & Mussa-Ivaldi, F. A. Guiding functional reorganization of motor redundancy using a body-machine interface. *J. NeuroEng. Rehab.* **17**, 61 (2020).
24. Benabid, A. L. et al. An exoskeleton controlled by an epidural wireless brain-machine interface in a tetraplegic patient: a proof-of-concept demonstration. *Lancet Neurol.* **18**, 1112–1122 (2019).
25. Müller, J. S. et al. A mathematical model for the two-learners problem. *J. Neural Eng.* **14**, 036005 (2017).
26. Madduri, M. M. et al. Co-adaptive myoelectric interface for continuous control. *IFAC-PapersOnLine* **55**, 95–100 (2022).
27. Dangi, S., Orsborn, A. L., Moorman, H. G. & Carmena, J. M. Design and analysis of closed-loop decoder adaptation algorithms for brain-machine interfaces. *Neural Comput.* **25**, 1693–1731 (2013).
28. Hsieh, H.-L. & Shanechi, M. M. Optimizing the learning rate for adaptive estimation of neural encoding models. *PLoS Comput. Biol.* **14**, e1006168 (2018).
29. Chasnov, B., Ratliff, L., Mazumdar, E. & Burden, S. Convergence analysis of gradient-based learning in continuous games. In *Conference on Uncertainty in Artificial Intelligence* Vol. 115 (eds Adams, R. P. & Gogate, V.) 935–944 (PMLR, 2020).
30. Merel, J., Pianto, D. M., Cunningham, J. P. & Paninski, L. Encoder-decoder optimization for brain-computer interfaces. *PLoS Comput. Biol.* **11**, e1004288 (2015).
31. De Santis, D. A framework for optimizing co-adaptation in body-machine interfaces. *Front. Neurobot.* **15**, 662181 (2021).
32. Madduri, M. M., Burden, S. A. & Orsborn, A. L. Biosignal-based co-adaptive user-machine interfaces for motor control. *Curr. Opin. Biomed. Eng.* **27**, 100462 (2023).
33. Braun, D. A., Ortega, P. A. & Wolpert, D. M. Nash equilibria in multi-agent motor interactions. *PLoS Comput. Biol.* **5**, e1000468 (2009).
34. Li, Y., Carboni, G., Gonzalez, F., Campolo, D. & Burdet, E. Differential game theory for versatile physical human-robot interaction. *Nat. Mach. Intell.* **1**, 36–43 (2019).
35. Chasnov, B. J., Ratliff, L. J. & Burden, S. A. Human adaptation to adaptive machines converges to game-theoretic equilibria. *Sci. Rep.* **15**, 29364 (2025).
36. Krakauer, J. W., Hadjiosif, A. M., Xu, J., Wong, A. L. & Haith, A. M. in *Motor Learning* 613–663 (John Wiley & Sons, 2019).
37. Seo, G., Kishta, A., Mugler, E., Slutzky, M. W. & Roh, J. Myoelectric interface training enables targeted reduction in abnormal muscle co-activation. *J. NeuroEng. Rehab.* **19**, 67 (2022).
38. Portnova-Fahreeva, A. A., Rizzoglio, F., Mussa-Ivaldi, F. A. & Rombokas, E. Autoencoder-based myoelectric controller for prosthetic hands. *Front. Bioeng. Biotechnol.* **11**, 1134135 (2023).
39. Yamagami, M., Steele, K. M. & Burden, S. A. Decoding intent with control theory: comparing muscle versus manual interface performance. In *ACM Conference on Human Factors in Computing Systems (CHI)* 1–12 (ACM, 2020).
40. Yang, C. S., Cowan, N. J. & Haith, A. M. De novo learning versus adaptation of continuous control in a manual tracking task. *eLife* **10**, e62578 (2021).
41. Orsborn, A. L., Dangi, S., Moorman, H. G. & Carmena, J. M. 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 (2012).
42. Gilja, V. et al. A high-performance neural prosthesis enabled by control algorithm design. *Nat. Neurosci.* **15**, 1752–1757 (2012).
43. Georgopoulos, A. P., Schwartz, A. B. & Kettner, R. E. Neuronal population coding of movement direction. *Science* **233**, 1416–1419 (1986).
44. Radhakrishnan, S. M., Baker, S. N. & Jackson, A. Learning a novel myoelectric-controlled interface task. *J. Neurophysiol.* **100**, 2397–2408 (2008).
45. Couraud, M., Cattaert, D., Pacllet, F., Oudeyer, P. Y. & de Rugy, A. Model and experiments to optimize co-adaptation in a simplified myoelectric control system. *J. Neural Eng.* **15**, 026006 (2018).

46. McRuer, D. T. & Jex, H. R. A review of quasi-linear pilot models. *IEEE Trans. Hum. Factors Electron.* **8**, 231–249 (1967).
47. Drop, F. M., Pool, D. M., Damveld, H. J., van Paassen, M. M. & Mulder, M. Identification of the feedforward component in manual control with predictable target signals. *IEEE Trans. Cybern.* **43**, 1936–1949 (2013).
48. Zhang, X., Wang, S., Hoagg, J. B. & Seigler, T. M. The roles of feedback and feedforward as humans learn to control unknown dynamic systems. *IEEE Trans. Cybern.* **48**, 543–555 (2018).
49. Yamagami, M., Peterson, L. N., Howell, D., Roth, E. & Burden, S. A. Effect of handedness on learned controllers and sensorimotor noise during trajectory-tracking. *IEEE Trans. Cybern.* **53**, 2039–2050 (2023).
50. Cowan, N. J. et al. Feedback control as a framework for understanding tradeoffs in biology. *Integr. Comp. Biol.* **54**, 223–237 (2014).
51. Åström, K. J. & Murray, R. M. *Feedback Systems: An Introduction for Scientists and Engineers* (Princeton Univ. Press, 2008).
52. Başar, T. & Olsder, G. J. *Dynamic Noncooperative Game Theory* 2nd edn (SIAM, 1999).
53. Hespanha, J. P. *Noncooperative Game Theory: An Introduction for Engineers and Computer Scientists* (Princeton Univ. Press, 2017).
54. Todorov, E. & Jordan, M. I. Optimal feedback control as a theory of motor coordination. *Nat. Neurosci.* **5**, 1226–1235 (2002).
55. Madduri, M. M., Burden, S. A. & Orsborn, A. L. A game-theoretic model for co-adaptive brain-machine interfaces. In *Proc. 2021 IEEE Conference on Neural Engineering (NER)* 327–330 (IEEE, 2021).
56. Monderer, D. & Shapley, L. S. Potential games. *Games Econ. Behav.* **14**, 124–143 (1996).
57. Strogatz, S. H. *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering* 2nd edn (Westview Press, 2015).
58. Orsborn, A. L., Dangi, S., Moorman, H. G. & Carmena, J. M. Exploring time-scales of closed-loop decoder adaptation in brain-machine interfaces. In *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society* 5436–5439 (IEEE, 2011).
59. Sadtler, P. T. et al. Neural constraints on learning. *Nature* **512**, 423–426 (2014).
60. Berger, D. J., Gentner, R., Edmunds, T., Pai, D. K. & d’Avella, A. Differences in adaptation rates after virtual surgeries provide direct evidence for modularity. *J. Neurosci.* **33**, 12384–12394 (2013).
61. Cheng, S. & Sabes, P. N. Modeling sensorimotor learning with linear dynamical systems. *Neural Comput.* **18**, 760–793 (2006).
62. Chase, S. M., Schwartz, A. B. & Kass, R. E. Bias, optimal linear estimation, and the differences between open-loop simulation and closed-loop performance of spiking-based brain-computer interface algorithms. *Neural Netw.* **22**, 1203–1213 (2009).
63. Cunningham, J. P. et al. A closed-loop human simulator for investigating the role of feedback control in brain-machine interfaces. *J. Neurophysiol.* **105**, 1932–1949 (2011).
64. Willett, F. R. et al. Principled BCI decoder design and parameter selection using a feedback control model. *Sci. Rep.* **9**, 8881 (2019).
65. Awasthi, P., Lin, T.-H., Bae, J., Miller, L. E. & Danziger, Z. C. Validation of a non-invasive, real-time, human-in-the-loop model of intracortical brain-computer interfaces. *J. Neural Eng.* **19**, 056038 (2022).
66. Hahne, J. M., Dähne, S., Hwang, H.-J., Müller, K.-R. & Parra, L. C. Concurrent adaptation of human and machine improves simultaneous and proportional myoelectric control. *IEEE Trans. Neural Syst. Rehabil. Eng.* **23**, 618–627 (2015).
67. Shanechi, M. M., Orsborn, A. L. & Carmena, J. M. Robust brain-machine interface design using optimal feedback control modeling and adaptive point process filtering. *PLoS Comput. Biol.* **12**, e1004730 (2016).
68. Hung, N.-T. et al. Wearable myoelectric interface enables high-dose, home-based training in severely impaired chronic stroke survivors. *Ann. Clin. Transl. Neurol.* **8**, 1895–1905 (2021).
69. Ingraham, K. A., Remy, C. D. & Rouse, E. J. The role of user preference in the customized control of robotic exoskeletons. *Sci. Robot.* **7**, eabj3487 (2022).
70. Willett, F. R. et al. Feedback control policies employed by people using intracortical brain-computer interfaces. *J. Neural Eng.* **14**, 016001 (2016).
71. Yamagami, M., Portnova-Fahreva, A. A., Kong, J., Wobbrock, J. O. & Mankoff, J. How do people with limited movement personalize upper-body gestures? Considerations for the design of personalized and accessible gesture interfaces. In *Proc. 25th International ACM SIGACCESS Conference on Computers and Accessibility 1* (ACM, 2023).
72. Flash, T. & Hogan, N. The coordination of arm movements: an experimentally confirmed mathematical model. *J. Neurosci.* **5**, 1688–1703 (1985).
73. Emken, J. L., Benitez, R., Sideris, A., Bobrow, J. E. & Reinkensmeyer, D. J. Motor adaptation as a greedy optimization of error and effort. *J. Neurophysiol.* **97**, 3997–4006 (2007).
74. Finley, J. M., Bastian, A. J. & Gottschall, J. S. Learning to be economical: the energy cost of walking tracks motor adaptation. *J. Physiol.* **591**, 1081–1095 (2013).
75. McDonald, K. A., Cusumano, J. P., Hieronymi, A. & Rubenson, J. Humans trade off whole-body energy cost to avoid overburdening muscles while walking. *Proc. R. Soc. B Biol. Sci.* **289**, 20221189 (2022).
76. Harris, C. M. & Wolpert, D. M. Signal-dependent noise determines motor planning. *Nature* **394**, 780–784 (1998).
77. Rajeswaran, P., Payeur, A., Lajoie, G. & Orsborn, A. L. Assistive sensory-motor perturbations influence learned neural representations. Preprint at *bioRxiv* <https://doi.org/10.1101/2024.03.20.585972> (2025).
78. Herzfeld, D. J., Vaswani, P. A., Marko, M. K. & Shadmehr, R. A memory of errors in sensorimotor learning. *Science* **345**, 1349–1353 (2014).
79. Athalye, V. R., Ganguly, K., Costa, R. M. & Carmena, J. M. Emergence of coordinated neural dynamics underlies neuroprosthetic learning and skillful control. *Neuron* **93**, 955–970.e5 (2017).
80. Dhawale, A. K., Smith, M. A. & Ölveczky, B. P. The role of variability in motor learning. *Annu. Rev. Neurosci.* **40**, 479–498 (2017).
81. Bengio, Y., Louradour, J., Collobert, R. & Weston, J. Curriculum learning. In *Proc. 26th International Conference on Machine Learning (ICML ’09)* 41–48 (ACM, 2009).
82. Yamagami, M. et al. Assessment of dry epidermal electrodes for long-term electromyography measurements. *Sensors* **18**, 1269 (2018).
83. Chvatal, S. A. P. D. & Ting, L. H. P. D. Common muscle synergies for balance and walking. *Front. Comput. Neurosci.* **7**, 48 (2013).
84. Virtanen, P. et al. SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nat. Methods* **17**, 261–272 (2020).
85. Wodlinger, B. et al. Ten-dimensional anthropomorphic arm control in a human brain-machine interface: difficulties, solutions, and limitations. *J. Neural Eng.* **12**, 016011 (2014).
86. Seáñez, I. & Mussa-Ivaldi, F. A. A body-machine interface for the control of a 2D cursor. In *Proc. 2013 International Conference on Rehabilitation Robotics (ICORR)* 6650508 (IEEE, 2013).

87. Knyazev, A. V. & Argentati, M. E. Principal angles between subspaces in an  $a$ -based scalar product: algorithms and perturbation estimates. *SIAM J. Sci. Comput.* **23**, 2008–2040 (2002).
88. Madduri, M., Yamagami, M., Li, S.J., Burckhardt, S., Burden, S.A. & Orsborn, A.L. Computational framework to predict and shape human-machine interactions in closed-loop, co-adaptive neural interfaces. *CodeOcean* <https://doi.org/10.24433/CO.4049054.v3> (2025).

## Acknowledgements

This work was supported by Meta Reality Labs Research, the National Science Foundation (NSF) award number 2045014 (S.A.B.), NSF award number 2338662 (A.L.O.) and NSF award number 2124608 (A.L.O. and S.A.B.). M.M.M. was funded in part by the Department of Defense, National Defense Science and Engineering Graduate Fellowship (NDSEG). The funders had no role in the study design, data collection and analysis, decision to publish or preparation of the Article. We thank A. X. T. Millevolte for experimental setup support.

## Author contributions

M.M.M., M.Y., S.A.B. and A.L.O. were responsible for the experimental design. M.M.M., M.Y. and S.J.L. were responsible for the experimental setup. M.M.M. and S.B. conducted the experiments. M.M.M. analysed the experimental data and prepared the figures. M.M.M., S.A.B. and A.L.O. wrote the paper.

## Competing interests

A.L.O. declares the following competing interest: employment as a scientific advisor for Meta Reality Labs. The remaining authors declare no competing interests.

## Additional information

**Extended data** is available for this paper at <https://doi.org/10.1038/s42256-026-01194-z>.

**Supplementary information** The online version contains supplementary material available at <https://doi.org/10.1038/s42256-026-01194-z>.

**Correspondence and requests for materials** should be addressed to Amy L. Orsborn.

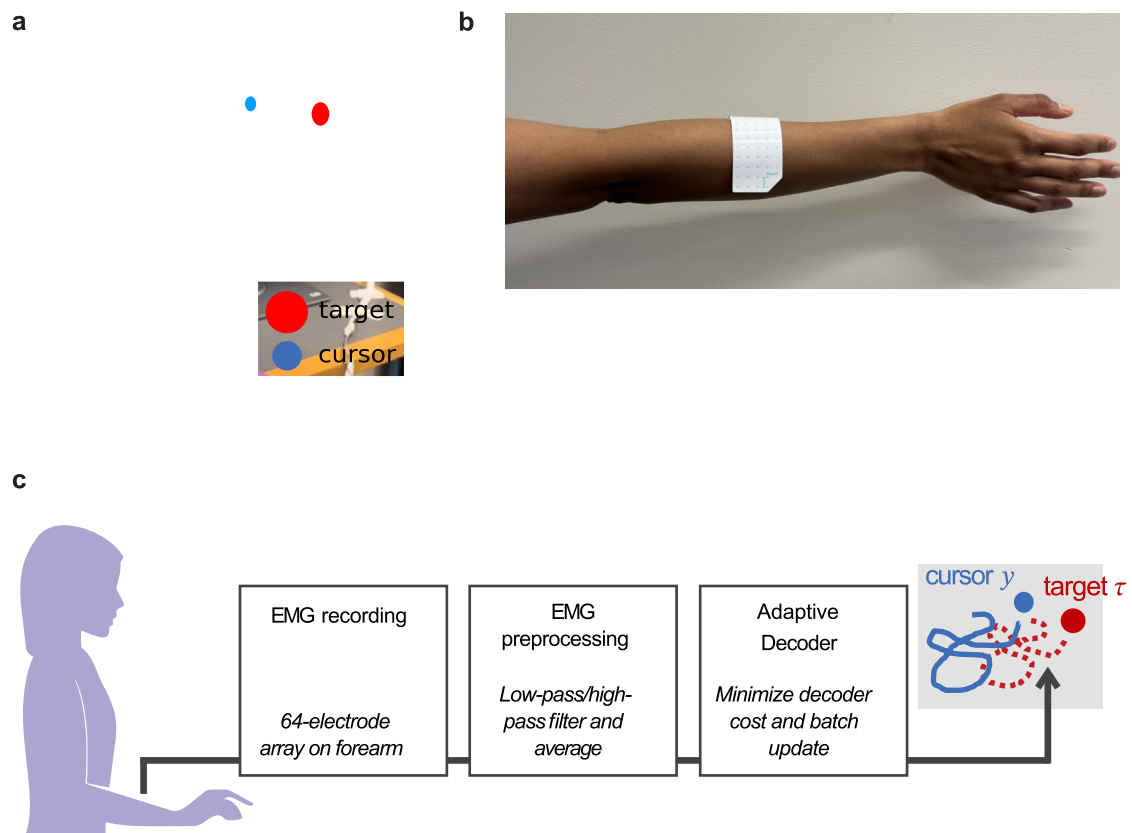
**Peer review information** *Nature Machine Intelligence* thanks Marco Capogrosso and Noah Cowan for their contribution to the peer review of this work.

**Reprints and permissions information** is available at [www.nature.com/reprints](http://www.nature.com/reprints).

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

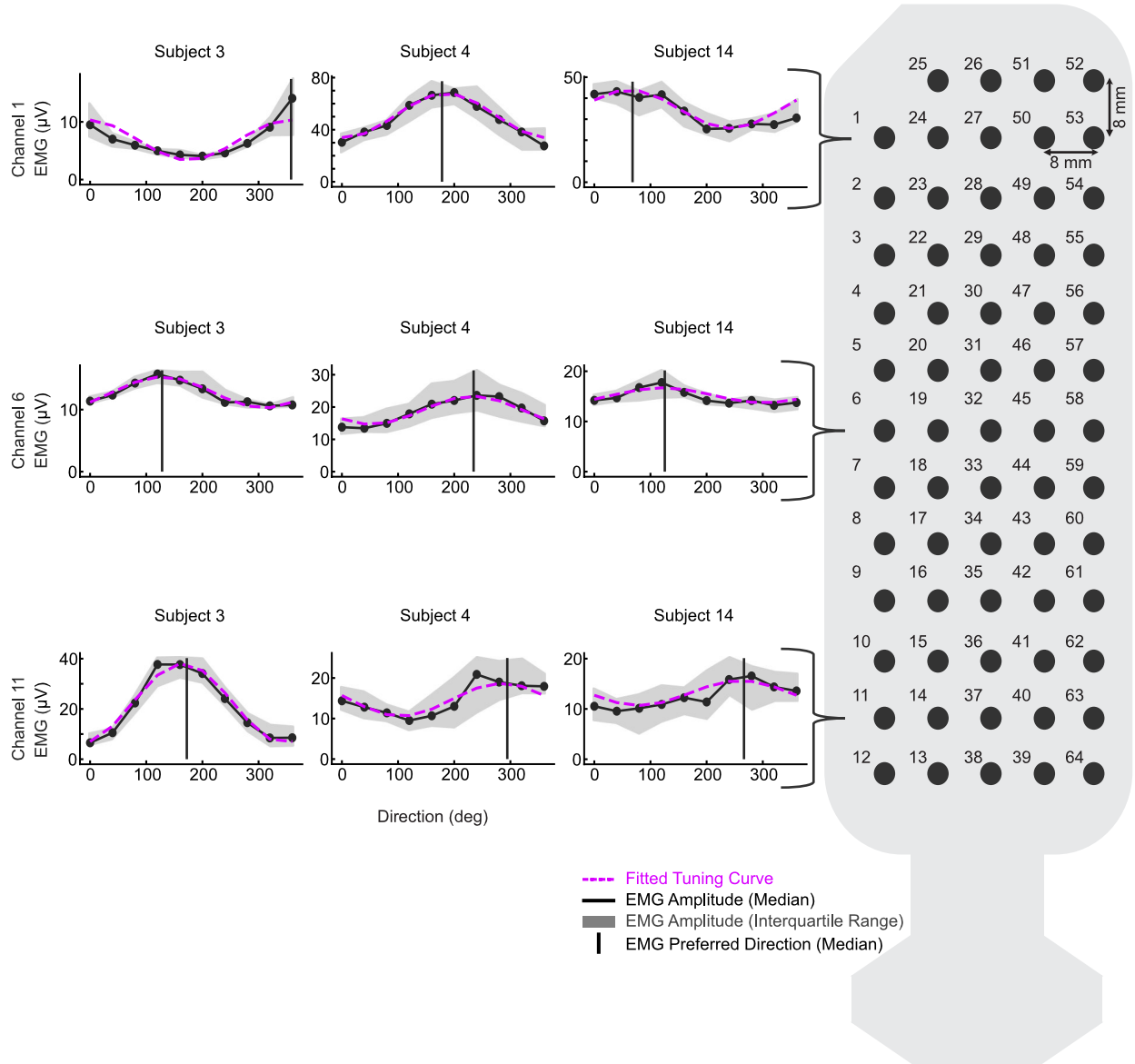
**Open Access** This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2026



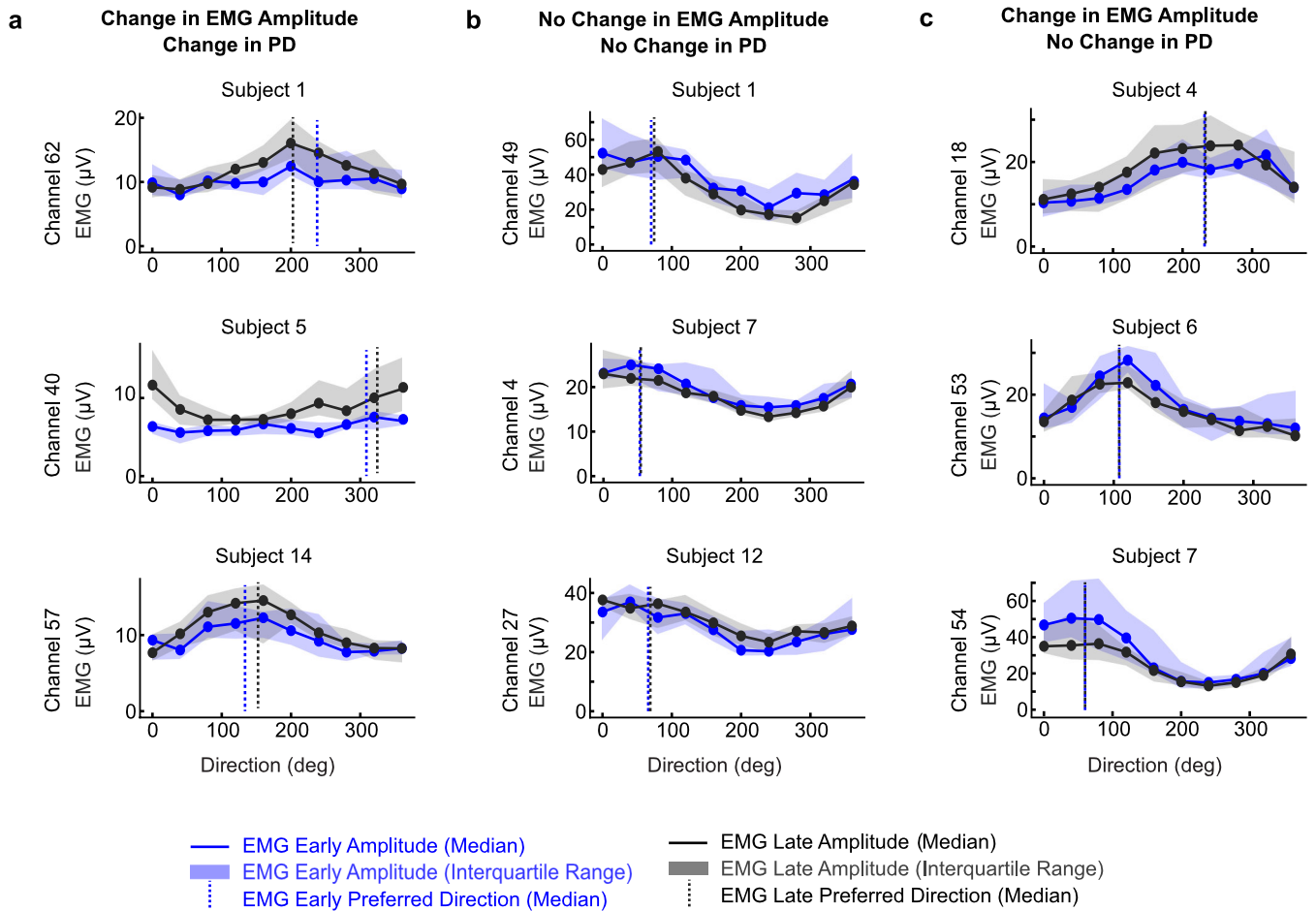
**Extended Data Fig. 1 | Experimental testbed.** **a.** Photograph of experiment. User tracks target (dashed red) by controlling the velocity of a 2D cursor (solid blue) with their forearm muscle activity. Muscle activity is collected via surface EMG electrodes that are placed on the participant's dominant forearm and wrapped with Coban tape. **b.** Electrode placement on dominant forearm. Surface

EMG electrodes are placed on user dominant forearm to target Extensor Carpi Radialis. **c.** Schematic of EMG preprocessing and decoding pipeline. EMG signals recorded from the user forearm are preprocessed and then input to the adaptive decoder. The decoder output is the cursor velocity that is integrated to display cursor position (blue) on the screen.



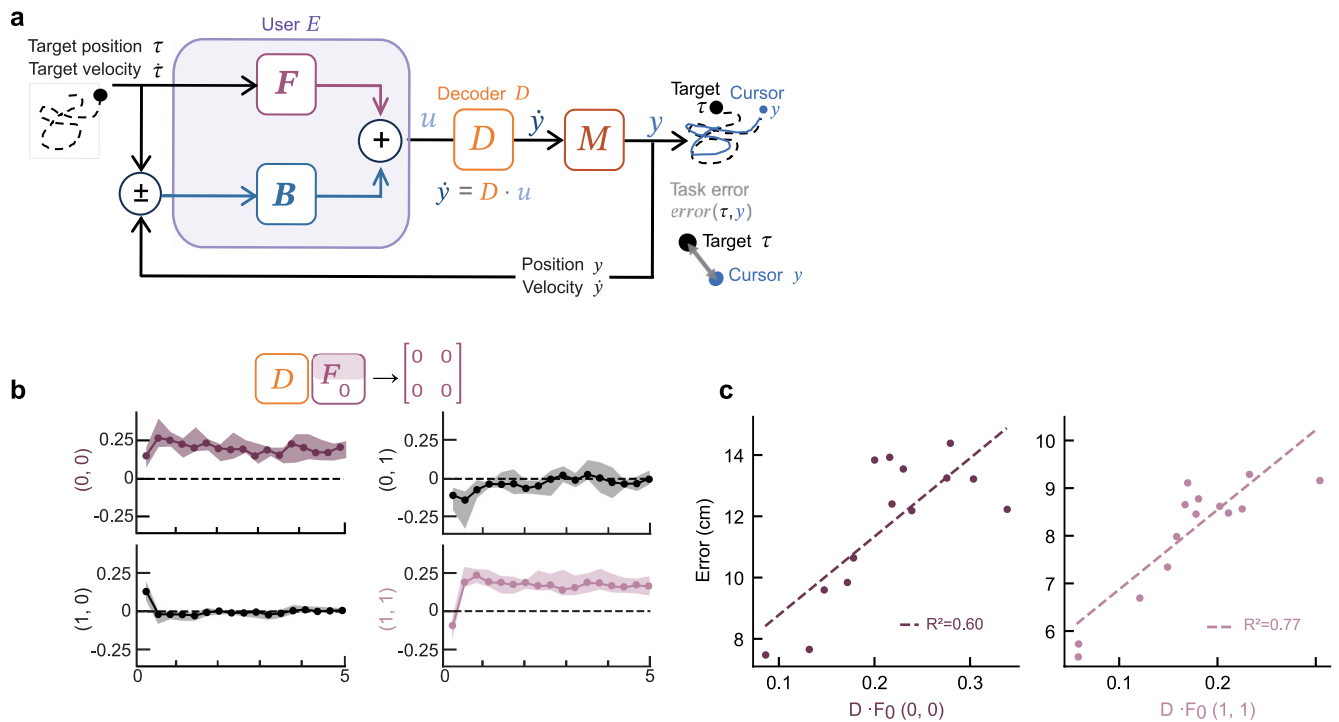
**Extended Data Fig. 2 | Surface EMG electrode and example tuning curves.** Diagram of surface EMG electrode and sample EMG tuning curves from representative subjects taken from late segments within trial, black line

represents median across trials with shaded area indicating the the 25th - 75th percentiles. Pink dashed lines show fitted tuning curves. Vertical lines represent preferred direction from late trial segments.



**Extended Data Fig. 3 | Example tuning curve changes from participants.** Sample EMG tuning curves from range of participants taken from early (blue) and late (black) trial segments. For all plots, lines represent the median across trials with shaded area indicating the 25th - 75th percentiles. Vertical dashed lines represent preferred direction (PD) from early (blue) and late (gray) trial segments. Examples were chosen to illustrate the diversity of qualitative changes

observed. **a.** Example EMG tuning curves showing qualitative differences in EMG amplitude and PD from early to late in the trial. **b.** Example EMG tuning curves showing minimal qualitative differences in EMG amplitude and PD from early to late in the trial. **c.** Example EMG tuning curves showing qualitative differences in EMG amplitude but minimal to no differences PD from early to late in the trial.

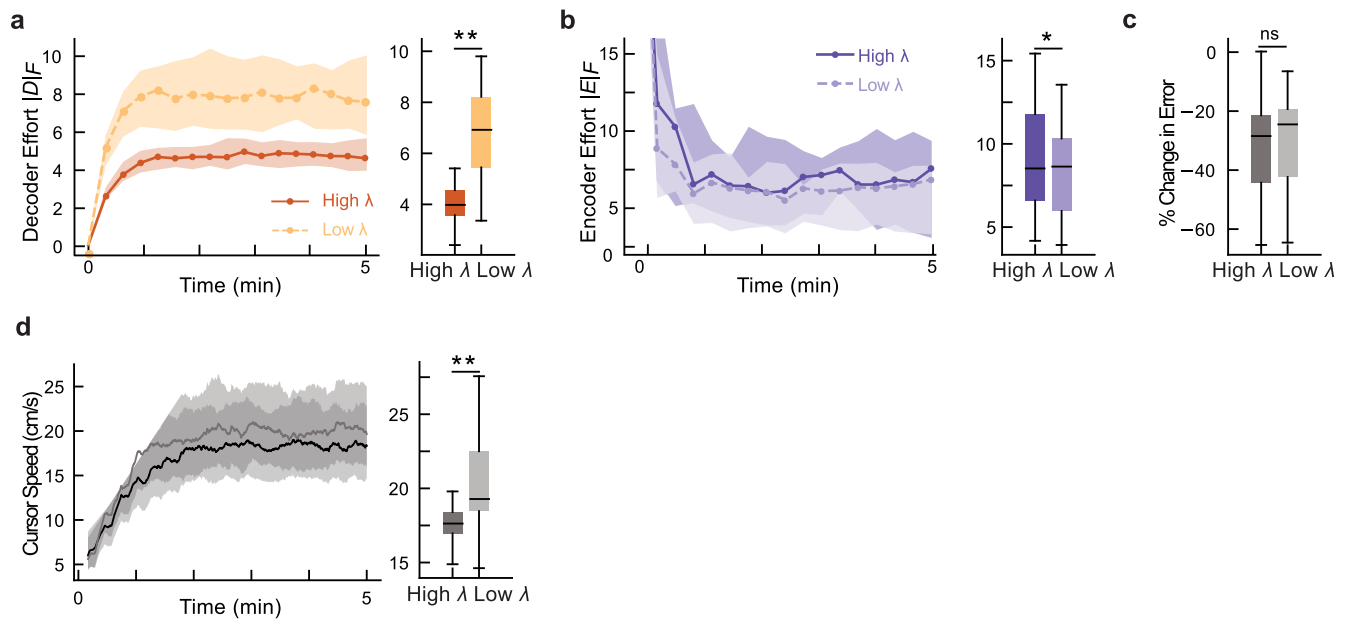


#### Extended Data Fig. 4 | Effect of error on Decoder-Encoder Feedforward Pairs.

**a.** Block diagram model of closed-loop system. The user's encoder  $E$  outputs myoelectric activity  $u$  to the decoder  $D$  to follow the target's position and velocity. The decoder outputs velocity, which is then integrated by system dynamics  $M$  to generate a cursor position. The user sees their cursor moving with some velocity  $\dot{y}$  to position  $y$ . Error in the cursor position and velocity (relative to the target) acts as closed-loop feedback to the user. (Duplicated from Fig. 2a for

completeness.) **b.** Product of the decoder matrix with first-order feedforward contributions of the encoder matrix ( $N=14$ , median; shading shows the 25th - 75th percentile). Black dashed lines represent values that yield perfect trajectory tracking. Purple dashed lines represent values that yield near perfect trajectory tracking. (Duplicated from Fig. 2c, left for completeness). **c.** Average tracking error (left: x-direction, right: y-direction) vs average  $D \cdot F_0$  value for each individual participant (dot).





**Extended Data Fig. 6 | Effect of penalty parameter in slow learning rate condition only.** **a.** Left: Average magnitude of the decoder matrix (norm) as a function of time in the trial for low (dashed light orange) and high (solid dark orange) decoder penalty terms (N=14, median; shading shows the 25th - 75th percentile). Right: Boxplots are average decoder effort across the trial for each subject (N=14, center shows median; box shows 25th-75th percentiles; whiskers extend to  $1.5 \times$  this interquartile range; one-sided Wilcoxon signed-rank test,  $**p = 6.1e - 5$ ). **b.** Left: Average magnitude of the user encoder matrix (norm) as a function of time in the trial for low (dashed light purple) and high (solid dark purple) decoder penalty terms (N=14, median; shading shows the 25th - 75th percentile). Right: Boxplots are average effort for each subject across the trial

(N=14, center shows median; box shows 25th-75th percentiles; whiskers extend to  $1.5 \times$  this interquartile range; one-sided Wilcoxon signed-rank test,  $*p = 0.018$ ). **c.** Percent change in error from the start of the trial (first 30 seconds) to the end of the trial (last 30 seconds), separated by high (dark gray) and low (light gray) decoder penalty term conditions (N=14, median; shading shows the 25th - 75th percentile; two-sided Wilcoxon signed-rank test, ns = 0.24). **d.** Left: Average cursor speed as a function of time in the trial for low (light gray) and high (dark gray) decoder penalty terms (N=14, median; shading shows the 25th - 75th percentile). Right: Boxplots are average cursor speed across the trial for each subject (N=14, center shows median; box shows 25th-75th percentiles; whiskers extend to  $1.5 \times$  this interquartile range; Wilcoxon signed-rank test,  $**p = 6.1e - 5$ ).

# Computational Framework to Predict and Shape Human-Machine Interactions in Closed-loop, Co-adaptive Neural Interfaces

November 21, 2025

## 1 Extended Data

Extended Table 1: List of Model Variables

Description	Symbol
Time	$t$
Task Dimensionality	$d = 2$
Number of EMG Channels	$N$
User Activity (sEMG Activity)	$u \in \mathbb{R}^{N \times t}$
Encoder/User	$E \in \mathbb{R}^{N \times 8}$
Encoder Feedforward	$F \in \mathbb{R}^{N \times 4}$
Encoder Feedback	$B \in \mathbb{R}^{N \times 4}$
Decoder	$D \in \mathbb{R}^{d \times N}$
Target Position	$\tau \in \mathbb{R}^d$
Target Velocity	$\dot{\tau} \in \mathbb{R}^d$
Cursor Position	$y \in \mathbb{R}^d$
Cursor Velocity	$\dot{y} \in \mathbb{R}^d$

## Reporting Summary

Nature Portfolio wishes to improve the reproducibility of the work that we publish. This form provides structure for consistency and transparency in reporting. For further information on Nature Portfolio policies, see our [Editorial Policies](#) and the [Editorial Policy Checklist](#).

### Statistics

For all statistical analyses, confirm that the following items are present in the figure legend, table legend, main text, or Methods section.

n/a	Confirmed
<input type="checkbox"/>	<input checked="" type="checkbox"/> The exact sample size ( $n$ ) for each experimental group/condition, given as a discrete number and unit of measurement
<input type="checkbox"/>	<input checked="" type="checkbox"/> A statement on whether measurements were taken from distinct samples or whether the same sample was measured repeatedly
<input type="checkbox"/>	<input checked="" type="checkbox"/> The statistical test(s) used AND whether they are one- or two-sided <i>Only common tests should be described solely by name; describe more complex techniques in the Methods section.</i>
<input type="checkbox"/>	<input checked="" type="checkbox"/> A description of all covariates tested
<input type="checkbox"/>	<input checked="" type="checkbox"/> A description of any assumptions or corrections, such as tests of normality and adjustment for multiple comparisons
<input type="checkbox"/>	<input checked="" type="checkbox"/> A full description of the statistical parameters including central tendency (e.g. means) or other basic estimates (e.g. regression coefficient) AND variation (e.g. standard deviation) or associated estimates of uncertainty (e.g. confidence intervals)
<input type="checkbox"/>	<input checked="" type="checkbox"/> For null hypothesis testing, the test statistic (e.g. $F$ , $t$ , $r$ ) with confidence intervals, effect sizes, degrees of freedom and $P$ value noted <i>Give <math>P</math> values as exact values whenever suitable.</i>
<input checked="" type="checkbox"/>	<input type="checkbox"/> For Bayesian analysis, information on the choice of priors and Markov chain Monte Carlo settings
<input checked="" type="checkbox"/>	<input type="checkbox"/> For hierarchical and complex designs, identification of the appropriate level for tests and full reporting of outcomes
<input checked="" type="checkbox"/>	<input type="checkbox"/> Estimates of effect sizes (e.g. Cohen's $d$ , Pearson's $r$ ), indicating how they were calculated

*Our web collection on [statistics for biologists](#) contains articles on many of the points above.*

### Software and code

Policy information about [availability of computer code](#)

Data collection	All data collection was performed with custom Python code (version 3.6.8) using Pygame (version 2.1.0) and LabGraph packages (publicly available). EMG data was collected using QTBioLabLite, version 1.1.3.
Data analysis	All data analysis was performed with custom Python code (Version information: python 3.8.12, scipy 1.7.3; numpy 1.22.1), which is publicly available via CodeOcean: DOI: 10.24433/CO.4049054.v2

For manuscripts utilizing custom algorithms or software that are central to the research but not yet described in published literature, software must be made available to editors and reviewers. We strongly encourage code deposition in a community repository (e.g. GitHub). See the Nature Portfolio [guidelines for submitting code & software](#) for further information.

### Data

Policy information about [availability of data](#)

All manuscripts must include a [data availability statement](#). This statement should provide the following information, where applicable:

- Accession codes, unique identifiers, or web links for publicly available datasets
- A description of any restrictions on data availability
- For clinical datasets or third party data, please ensure that the statement adheres to our [policy](#)

All data are publicly available in a Code Ocean capsule, DOI: 10.24433/CO.4049054.v2

## Human research participants

Policy information about [studies involving human research participants and Sex and Gender in Research](#).

Reporting on sex and gender	Gender was determined based on self-reported data from participants. Our participant pool included 9 women, 4 men, 1 prefer not to answer. Disaggregate data is not reported and the study was not designed to determine potential sex or gender differences in outcomes.
Population characteristics	Participant demographic information (including gender, weight, height, age, and handedness) was collected via a demographics survey that participants completed before the experiment (see below). Only forearm circumference was measured by the experimenter, all other information was self-reported.  Gender: 9 women, 4 men, 1 prefer not to answer weight (mean +/- std, lbs.): 140 +/- 21.6 Height (mean +/- std, in.): 65 +/- 3.4 Age (mean +/- std, years): 23 +/- 3.9 Forearm circumference (mean +/- std., in.): 10 +/- 0.8 Handedness: 12 right-handed, 2 left-handed level of ambidexterity: 8 dominant, 5 mostly dominant, 1 ambidextrous
Recruitment	Participants were recruited to our study through on-campus advertisements and electronic advertisements (e.g., emails, social media posts). The sample is therefore biased towards the populations on the University of Washington campus (e.g., undergraduates, graduate students), which may influence the generality of findings.
Ethics oversight	University of Washington Institutional Review Board (IRB #STUDY00014060).

Note that full information on the approval of the study protocol must also be provided in the manuscript.

## Field-specific reporting

Please select the one below that is the best fit for your research. If you are not sure, read the appropriate sections before making your selection.

Life sciences  Behavioural & social sciences  Ecological, evolutionary & environmental sciences

For a reference copy of the document with all sections, see [nature.com/documents/nr-reporting-summary-flat.pdf](https://nature.com/documents/nr-reporting-summary-flat.pdf)

## Behavioural & social sciences study design

All studies must disclose on these points even when the disclosure is negative.

Study description	The study primarily uses quantitative methods to analyze user behavior based on task performance and electrophysiology measures (electromyography) as they interact with a dynamic algorithm.
Research sample	The research sample is from undergraduate and graduate students at the University of Washington Seattle. Based on available demographic information about students on campus, our sample is slightly biased towards females (64% in our sample vs. 54-56% female in undergraduate and graduate populations). Our age range appears representative of the population.
Sampling strategy	The rationale for the sampling strategy and the sample size was based on prior studies of muscle-controlled interfaces (e.g., Yamagami et al., CHI 2020, N = 11; Saponas et al., CHI 2008, N = 13; Saponas et al., UIST 2009, N = 12; Madduri et al., CPHS 2022, N = 7; Huang et al., UIST 2015 N = 7). Sample sizes were pre-determined before beginning data collection based on these prior studies.
Data collection	Participants were asked to track a continuously-moving target with a cursor controlled with their muscle activity (electromyography, EMG). EMG data was collected from non-invasive sensors applied to the forearm. Behavioral data was collected through the custom python interface to monitor task performance. Participants performed the task in the presence of the researcher(s) (1-2 people) performing data collection. No other people were present in the room during data collection. Study conditions were randomized via a computer program, and participants were blinded to conditions but researchers were not.
Timing	All data for each participant was collected in a single session lasting 2-3 hours. Data for all 14 participants was collected over the span of approximately 6 months based on availability of researchers and participants.
Data exclusions	No data was excluded.
Non-participation	No participants withdrew from the study.
Randomization	Participants were not allocated into experimental groups.

# Reporting for specific materials, systems and methods

We require information from authors about some types of materials, experimental systems and methods used in many studies. Here, indicate whether each material, system or method listed is relevant to your study. If you are not sure if a list item applies to your research, read the appropriate section before selecting a response.

## Materials & experimental systems

n/a	Included in the study
<input checked="" type="checkbox"/>	<input type="checkbox"/> Antibodies
<input checked="" type="checkbox"/>	<input type="checkbox"/> Eukaryotic cell lines
<input checked="" type="checkbox"/>	<input type="checkbox"/> Palaeontology and archaeology
<input checked="" type="checkbox"/>	<input type="checkbox"/> Animals and other organisms
<input checked="" type="checkbox"/>	<input type="checkbox"/> Clinical data
<input checked="" type="checkbox"/>	<input type="checkbox"/> Dual use research of concern

## Methods

n/a	Included in the study
<input checked="" type="checkbox"/>	<input type="checkbox"/> ChIP-seq
<input checked="" type="checkbox"/>	<input type="checkbox"/> Flow cytometry
<input checked="" type="checkbox"/>	<input type="checkbox"/> MRI-based neuroimaging