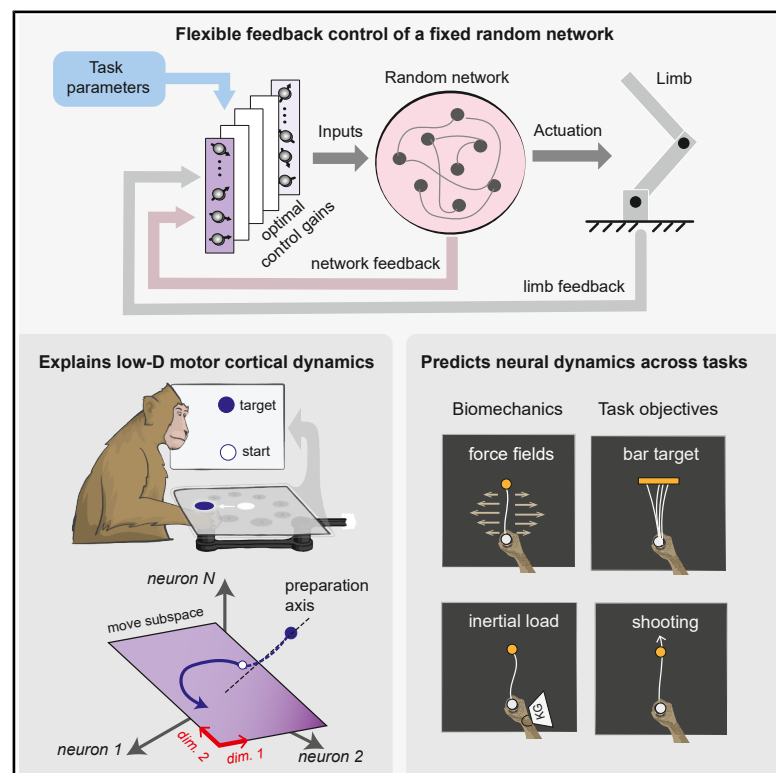


Feedback control of random networks as a model of flexible motor cortical dynamics across tasks

Graphical abstract



Authors

Hari Teja Kalidindi, Frédéric Crevecoeur

Correspondence

hari.kalidindi@donders.ru.nl (H.T.K.),
frederic.crevecœur@uclouvain.be (F.C.)

In brief

Kalidindi and Crevecoeur develop a computational framework linking feedback-controlled networks to limb dynamics. They demonstrate that optimal control of fixed network reproduces key motor cortical dynamics and predicts neural activity across tasks. Analytical results show low-dimensional patterns emerge from task and biomechanical complexity, thereby bridging neural dynamics with control theory.

Highlights

- Random artificial networks mimic motor cortical dynamics across multiple behaviors
- Analyses and simulations show limb feedback drives low-dimensional neural activity
- Model captures neural dynamics for reaching with loads and responses to perturbations
- Task and biomechanics jointly influence the dimensionality of neural dynamics



Article

Feedback control of random networks as a model of flexible motor cortical dynamics across tasks

Hari Teja Kalidindi^{1,2,3,*} and Frédéric Crevecoeur^{1,2,4,5,*}

¹Institute of Communication Technologies, Engineering and Applied Mathematics (ICTEAM), UCLouvain, Avenue Georges Lemaitre 4-6, 1348 Louvain-la-Neuve, Belgium

²Institute of Neuroscience, UCLouvain, Avenue E. Mounier 53, 1200 Brussels, Belgium

³Donders Centre for Cognition, Radboud University, Thomas Van Aquinostraat 4, 6525D Nijmegen, the Netherlands

⁴WEL Research Institute, Wavre, Belgium

⁵Lead contact

*Correspondence: hari.kalidindi@donders.ru.nl (H.T.K.), frederic.crevecœur@uclouvain.be (F.C.)

<https://doi.org/10.1016/j.celrep.2026.116991>

SUMMARY

Modern large-scale recordings show that motor cortex activity during reaching evolves along low-dimensional trajectories, often interpreted as neural bases for motor command generation. However, the origin of these patterns and how they flexibly reorganize across tasks remains unclear. Here, we show that hallmark features of motor cortical activity, including the encoding of several movement parameters during preparation and rotational dynamics during execution, emerge naturally in a linear model composed of a random network coupled to a biomechanical system. Under optimal control, the model achieves multiple behaviors through task-dependent mapping of sensory feedback onto the network. The corresponding feedback rule enables a rich motor repertoire while producing network activity consistent with motor cortical recordings. In this model, the relationship between feedback signals from the biomechanical plant and the low-dimensional network dynamics can be established analytically, which offers an interpretable framework to link neural dynamics with sensorimotor control across tasks.

INTRODUCTION

A central challenge in neuroscience is to explain how neural activity generates behavior and adapts to changing environments.¹ To address this, recent advances have shifted focus from individual neurons to the collective dynamics of populations of neurons.^{2–4} Empirical studies show that the population dynamics can be effectively captured by rather low-dimensional, time-varying signals, referred to as *neural factors*⁵ or *latent dynamical state*.^{6,7} These factors have been interpreted as signals that the brain must generate to orchestrate behavior.^{5,8} Yet, fundamental questions remain: what is the relationship between neural factors and biological or physical variables that characterize the movement and which factors are required for a given behavior?

Prevailing theories propose that neural factors emerge from recurrent connectivity among cortical and subcortical areas, forming attractors that shape population activity.⁶ Artificial recurrent neural networks (RNN) capture this idea by developing stable patterns—such as tonic or multi-phasic activity—generating muscle commands when optimized on sensorimotor tasks.^{9–17} However, generalization to other tasks remains challenging for these models, which rely on training based on novel relevant datasets.¹⁸ Their behavior is often difficult to interpret due to non-linearities and dependency on both training data¹⁸ and training algorithms.¹⁹ This raises the question: are simpler, more interpretable models sufficient to account for some key features of motor cortical dynamics?

In contrast to trained artificial networks, primates exhibit remarkable flexibility as they can rapidly adapt to novel contexts

or even update control within an ongoing movement.^{20–24} Although development certainly forms a wide range of behaviors akin to long-term training, recent computational modeling studies suggest that contextual inputs to the network can evoke patterns suitable to a given task.^{25–33} However, it remains unclear how inputs are computed and adjusted across diverse sensorimotor tasks, such as different instructions or loading conditions.

Here, we propose a unifying theoretical framework that explains both the emergence of low-dimensional factors in motor cortical dynamics and how inputs to the network should be computed to accomplish the task at hand. We developed a linear model coupling a random network to a biomechanical limb and simulated various tasks using stochastic optimal control.^{31,34,35} In this framework, the feedback control policy maps the feedback about the state of the whole system (network and limb) onto optimal network activations. We demonstrate that this approach directly links the parameters of the controller to the network activity across tasks without the need for example-driven retraining. Remarkably, our model reproduced key features observed in motor cortex, including preparatory activity anticipating the upcoming movement,^{36,37} rotational dynamics during movement,^{8,38} and activity during movement preparation and execution periods restricted to orthogonal subspaces.³⁹ Furthermore, this linear formulation offers an explanation for the origin of low-dimensional factors using the low-rank hypothesis of complex systems⁴⁰: although the network connectivity is high-dimensional, its activity is mainly driven by inputs carrying



sensory feedback and task parameters, yielding low-dimensional patterns of activation. Together, our results propose a formal link between sensorimotor control and low-dimensional population activity for a fixed random network, offering a powerful candidate framework to interpret the neural basis of flexible motor control.

RESULTS

We constructed a super-system composed of a randomly connected recurrent network ($n = 100$ nodes) coupled to a limb (Figure 1A). The network activity, $\mathbf{r}(t) \in \mathbb{R}^{100}$, was represented as a standard rate equation.^{9,12,26,31} The network output controlled one of the two limb models: a point-mass moving in a cartesian plane or a two-link planar arm. Both models featured two-dimensional actuation, with cartesian forces applied along the plane for the point-mass and joint torques applied to the shoulder and elbow joints for the planar arm. The limb state, $\mathbf{q}(t) \in \mathbb{R}^6$ encompassed position, velocity, and actuation forces/torques. The full details of the network and limb dynamics are provided in STAR Methods. For compact representation, the linear dynamics governing the network-body system were jointly described by:

$$\begin{bmatrix} \dot{\mathbf{r}} \\ \dot{\mathbf{q}} \end{bmatrix} = \begin{bmatrix} \mathbf{J} & \mathbf{0} \\ \mathbf{C} & \mathbf{M} \end{bmatrix} \begin{bmatrix} \mathbf{r} \\ \mathbf{q} \end{bmatrix} + \begin{bmatrix} \boldsymbol{\epsilon} \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{I} \\ \mathbf{0} \end{bmatrix} \mathbf{u}, \quad (\text{Equation 1})$$

where $\mathbf{J} \in \mathbb{R}^{100 \times 100}$ was the random state transition matrix, $\boldsymbol{\epsilon} \in \mathbb{R}^{100}$ was a constant input vector inducing heterogeneous spontaneous activity, $\mathbf{C}$ was the random linear readout that mapped the network activity to limb actuation, and $\mathbf{M}$ was the state transition matrix of limb dynamics (the explicit dependency on time was omitted from Equation 1 for clarity). The control input $\mathbf{u}(t) \in \mathbb{R}^{100}$ influenced only the network activity. The identity matrix $\mathbf{I}$ denotes that each control input component activated a single network unit.

The objective was to find inputs $\mathbf{u}(t)$ that steer the network activity to make the limb move optimally as defined by a cost function. We adopted the formalism of receding-horizon to allow modeling changes in task instructions, such as the unexpected occurrence of a GO cue presented during movement preparation (see STAR Methods). This problem was solved as follows: we determined the control inputs that minimized the expected cost ($\mathcal{J}(\mathbf{q}, \mathbf{u}, t)$) over a time-horizon Δh defined by:

$$\mathcal{J}(\mathbf{q}, \mathbf{u}, t) = \int_t^{t+\Delta h} \mathbf{q}(s)^T \mathbf{Q}(s) \mathbf{q}(s) + \mathbf{u}(s)^T \mathbf{R}(s) \mathbf{u}(s) ds. \quad (\text{Equation 2})$$

We then applied the first time step of the control law and repeated the process by moving the time horizon forward. Here, $\mathbf{Q}(\cdot)$ is a positive-semidefinite matrix penalizing deviations from desired limb states (note that the state vector includes the target, omitted here for simplicity, see STAR Methods), and $\mathbf{R}(\cdot)$ is a positive-definite matrix penalizing the large network inputs ($\mathbf{u}$).

We now simply consider the structure of the controller to interpret its effect on the network-body dynamics across task varia-

tions. For the given system dynamics (Equation 1) and quadratic cost function (Equation 2), the optimal control inputs for each time-horizon can be computed using the classical linear quadratic regulator (LQR) approach^{35,41} (see STAR Methods), which yields an optimal state-feedback control law that can be decomposed as follows:

$$\mathbf{u}(t) = -\mathbf{L}_{net}(\boldsymbol{\Theta}, t) \mathbf{r}(t) - \mathbf{L}_{limb}(\boldsymbol{\Theta}, t) \mathbf{q}(t). \quad (\text{Equation 3})$$

The set $\boldsymbol{\Theta}$ encompasses all the system and cost parameters (i.e., $\boldsymbol{\Theta} = \{\mathbf{Q}, \mathbf{R}, \mathbf{J}, \mathbf{C}, \mathbf{M}\}$), and $\mathbf{L}_{net}$ and $\mathbf{L}_{limb}$ are the control gains applied to the network and limb states, respectively. Substituting this feedback control law (Equation 3) into the system dynamics (Equation 1) gives the closed-loop system:

$$\begin{bmatrix} \dot{\mathbf{r}} \\ \dot{\mathbf{q}} \end{bmatrix} = \begin{bmatrix} \mathbf{J} - \mathbf{L}_{net}(\boldsymbol{\Theta}, t) & -\mathbf{L}_{limb}(\boldsymbol{\Theta}, t) \\ \mathbf{C} & \mathbf{M} \end{bmatrix} \begin{bmatrix} \mathbf{r} \\ \mathbf{q} \end{bmatrix} + \begin{bmatrix} \boldsymbol{\epsilon} \\ \mathbf{0} \end{bmatrix}. \quad (\text{Equation 4})$$

Rewriting the system in this closed-loop form offers an interpretation of the controller structure: this optimal control scheme enables multiple tasks (represented by $\boldsymbol{\Theta}$) by dynamically adjusting the coupling of network nodes ($\mathbf{J}$ became $\mathbf{J} - \mathbf{L}_{net}(\boldsymbol{\Theta}, t)$), and the feedback about the limb states through the matrix $-\mathbf{L}_{limb}(\boldsymbol{\Theta}, t)$ (Figure 1A). In essence, this approach leverages the linear structure of the system to compute the optimal network connectivity suitable for a given set of parameters without data-driven retraining. Thus, we may expect that different tasks (and their corresponding sets of parameters) induce different network activations, resulting in various shapes and trajectories of the network states representing neural population activity (Figure 1B). We show below that this model remarkably reproduces previously published properties (Figure 1C) and makes quantitative predictions of neural activities associated with standard tasks (Figure 1D).

Emergence of low-dimensional factors for preparation and execution

We first validate the model by showing that it reproduces hallmark features of monkey's primary motor cortex (M1) activity during a standard center-out reaching task^{8,39} (Figure 1C). Specifically, using a receding-horizon control scheme^{42,43} (see STAR Methods), the model generated realistic hand trajectories in a delayed prepare-to-reach paradigm, including bell-shaped velocity profiles following the "GO" cue (Figure 2A). At the level of individual network nodes, we observed heterogeneous activity patterns, with tonic levels of activity during the preparatory phase that quickly encoded the impending target, and a subsequent phasic activity linked to movement execution (Figure 2B). Both the amplitude and the phase of individual responses varied across nodes and movement directions. Notably, many nodes exhibited an overshoot immediately after target onset, resembling the M1 recordings.^{36,44}

To characterize the interactions between nodes across epochs, we first analyzed functional connectivity by computing pairwise correlations of time-series activity during the preparatory (150–50ms before the GO cue) and movement epochs (50–150ms after the GO cue). Figure 2C shows that the average pairwise correlations across targets observed

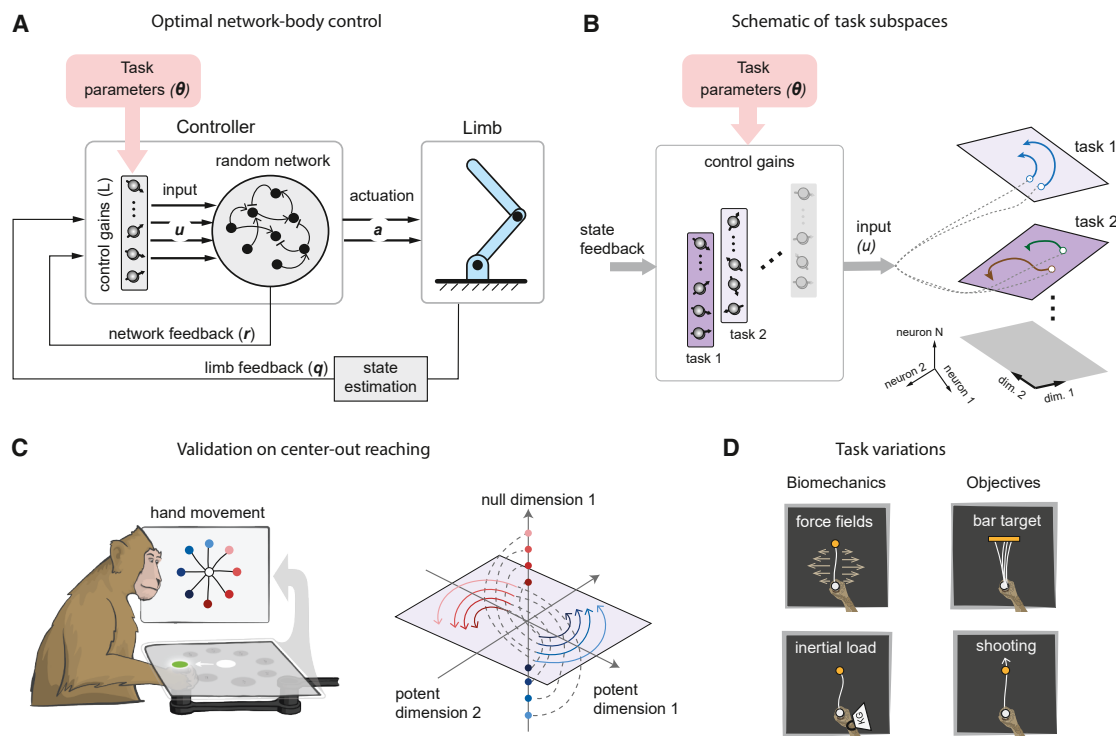


Figure 1. Schematic representation of the network-body system

(A) Task-dependent control of the network-body system through tunable control gains.

(B) Illustration of how the tunable control gains may reshape the low-dimensional neural trajectories based on task parameters.

(C) Illustration of a stereotypical prepare-to-reach task, showing how the neural population activity is driven into orthogonal subspaces for preparation and movement, where dots represent preparatory states, curves represent activity over time, and color coded to different directions of reaching.

(D) Selected benchmark tasks taken from standard motor control literature used to test the malleability of sensorimotor control in primates, dependent on changes in task objectives (flexibility) and biomechanics (adaptation).

during preparation were not maintained during movement. Consequently, the pairwise correlations among nodes during preparation were not linearly related to those observed during movement. This resulted in a low Pearson correlation coefficient between epochs (Figure 2D), indicating a reorganization of network interactions driven by the task objective to prepare or execute movement.

To examine the dominant features of population activity and link them to previously reported neural factors, we performed principal component analysis (PCA) on the network activity during preparatory and movement epochs (see STAR Methods). The dominant principal components (PCs) in the preparatory epoch encoded target information prior to movement onset (Figure 2E). This finding demonstrates that directional encoding of the upcoming movement occurred within a subspace of $\mathbb{R}^n$ that did not produce any motor output. This observation is consistent with experimental observations where neuronal activity unfolds along “output-null” dimensions that do not produce any movement.^{5,39} The same PCA applied to the population activity after the GO cue revealed rotational trajectories (Figure 2F), closely resembling rotations reported in M1.⁸ Notably, the control scheme showed distinct responses between preparatory versus movement epochs because the task objectives (Q) changed: during preparation, deviation from the start was penal-

ized, whereas during movement, deviation from the goal was penalized. As a result, the controller allowed the same state-feedback signals to produce different network trajectories depending on whether the system was preparing or executing a movement.

This task-dependent control scheme also captured another well-known feature of M1 activity: the orthogonality between preparatory and execution related activity.³⁹ Applying PCA to network activity in each epoch separately, the preparatory-PCs explained little of the variance observed during the movement, and vice versa. To quantify the degree of overlap, we computed the alignment index³⁹ (see STAR Methods), measuring how much variance in the preparatory epoch was explained by the first six PCs from the execution epoch, normalized by the variance explained by the first six PCs from the preparatory epoch. When the activities during the preparation and execution phases are confined to orthogonal subspaces, respectively, without any overlap, the alignment index is zero. In our model, the network activities exhibited a low alignment index, consistent with the recordings from M1³⁹ and significantly lower than surrogate datasets generated via the tensor maximization entropy (TME) procedure (Figure 2G, see STAR Methods).

The foregoing observation highlighted low alignment between preparatory and movement-related activities, as well as

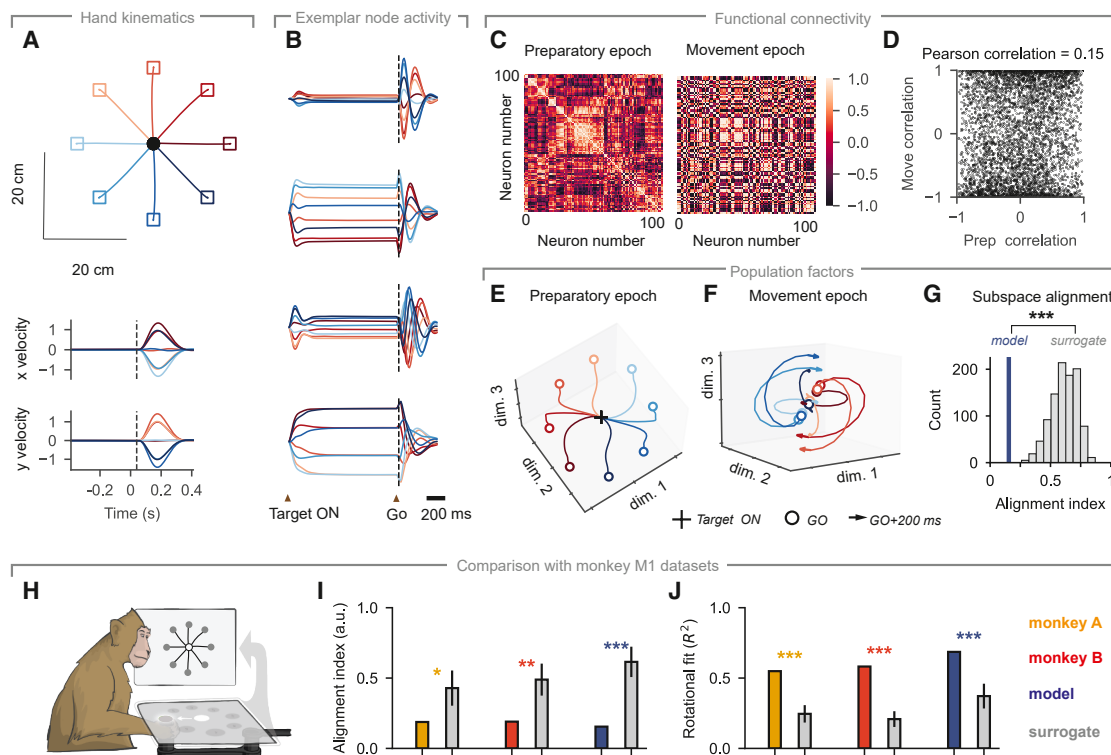


Figure 2. Feedback control produces meaningful behavior and reproduces M1-like preparation and movement-related “factors”

(A) Hand kinematics for the simulation of a point mass system.
 (B) Exemplar neural firing rates from selected units show that the activity of some neurons depends on the target of the pending movement.
 (C) Pairwise correlations in preparatory and movement periods.
 (D) Linear relationship between pairwise correlations from movement to the preparatory period. The lack of pattern shows that activities from one epoch are statistically independent of those of the other epoch.
 (E) Neural activity along the principal modes in the preparatory period.
 (F) Same as (E) for the movement period.
 (G) Alignment between preparatory and movement period activities. The histogram shows the distribution of alignment indices across surrogate data, and the vertical line represents the alignment of subspaces from the principal modes of preparatory and execution epochs (low alignment corresponds to orthogonal).
 (H) Comparison of the model with monkey datasets: monkey A⁸ and monkey B⁴⁵.
 (I) Comparison of the alignment index of the top six preparatory and movement period PCs.
 (J) The goodness of fit (R^2) obtained by fitting the top six PC trajectories to a rotational dynamic system using the jPCA procedure. Surrogate data are represented as mean $\pm$ SEM across samples. Asterisk symbols indicate the p values from one-tailed test: * p value < 0.05 , ** p value < 0.01 , *** p value < 0.005 .
 See also Figures S2–S4.

rotational trajectories. To evaluate the model’s resemblance to the motor cortex quantitatively, we compared its behavior with publicly available datasets of neural recordings in monkeys (Figure 2H).^{8,45} Specifically, we compared the alignment index and rotational dynamics in both the model and M1 data. Both the model and M1 recordings exhibited low alignment indices, substantially lower than those observed in surrogate datasets (Figure 2I). Rotational dynamics were quantified using the jPCA method,⁸ which was used to fit the top six PCs to a constrained dynamical system with purely rotational structure (see STAR Methods). The goodness of fit, measured by R^2 , was consistent across M1 recordings and model simulations and significantly larger than surrogate datasets generated via the TME procedure (Figure 2J).

To further characterize the rotational structure from the firing rate without dimensionality reduction, we computed

the complex-valued gyration number.⁴⁶ The real and imaginary parts of the gyration number capture the normalized decay and rotational components in the population activity, respectively (see STAR Methods). The model and M1 recordings exhibited comparable levels of rotation and decay, whereas shuffled datasets showed markedly reduced rotational component (Figure S1).

Thus, closed-loop control of the network-body system naturally gave rise to low-dimensional neural trajectories in orthogonal preparatory and movement subspaces, with rotational population dynamics closely paralleling the patterns of neural activities recorded in M1. Remarkably, these features emerged from a linear system combined with a quadratic cost. No additional constraint was required, such as enforcing simple network dynamics⁹ or minimizing trajectory tangling,^{15,47} as expressed in previous models.

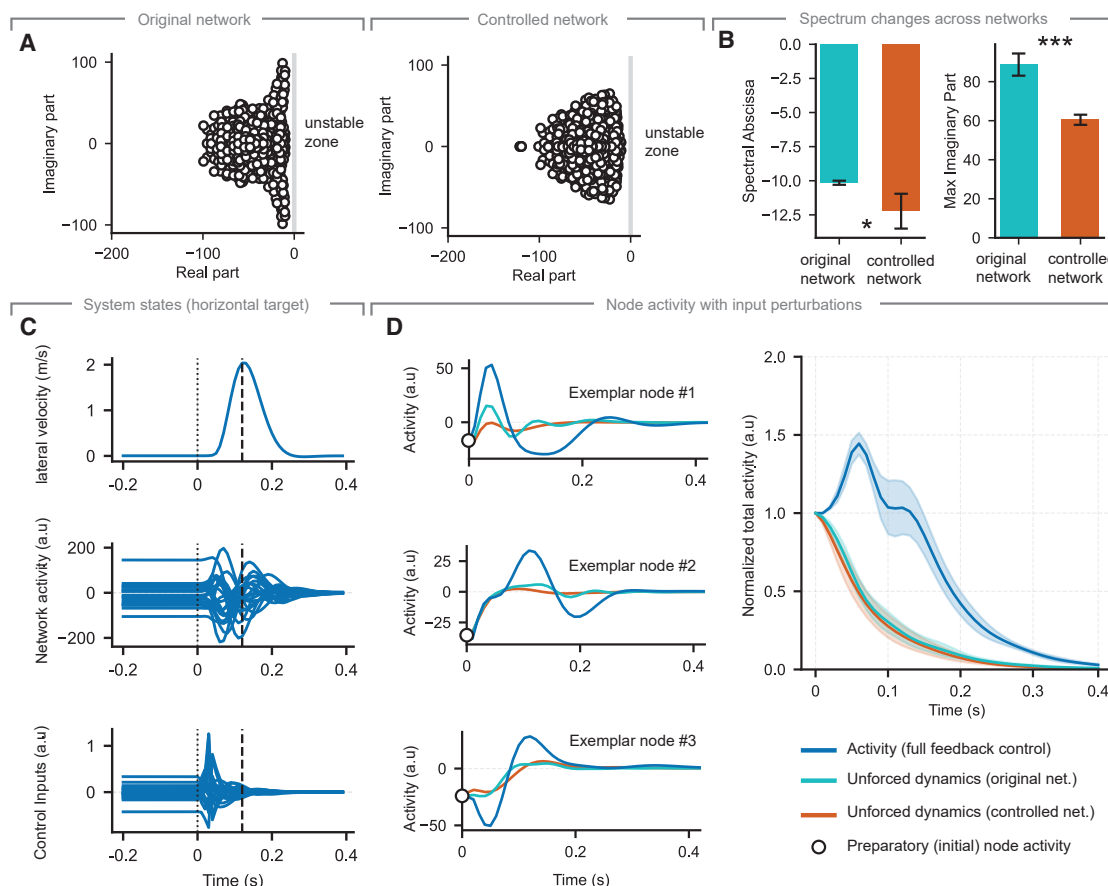


Figure 3. Model analysis

(A) Eigenvalue spectrum of an exemplar original network (left) and the same network when modified by the state feedback control law (right, Equation 4). (B) Change in spectral abscissa (maximum real part) and in the maximum imaginary part of the eigenvalues across 5 networks. (C) Simulated hand velocity, network activity, and control inputs while reaching to the target aligned with the x axis. Dotted and dashed vertical lines represent the time of the GO cue and the time of maximum velocity, respectively. (D) Left: node activity with full state feedback inputs compared with the unforced dynamics after the same initialization. Right: normalized total activity across nodes and reach directions. Data are represented as mean $\pm$ SEM across networks. * p value < 0.05 *** p value < 0.005 from paired two-sample t test across $n = 5$ different networks.

We made several additional observations that are reported in the supplementary material, which can be summarized as follows: our results appear robust against variations in the model structures for comparable networks and inputs. Figure S2 shows that the results for the center-out reaching task were similar when coupling the network to a two-link arm (see methods S1) instead of a point mass, as shown in Figure 2. Then, we also observed low alignment between preparatory and movement-related activity, as well as rotations, in fully connected recurrent networks without any excitation-inhibition balanced structure and randomly sampled Erdős-Rényi networks (see STAR Methods; Figures S3 and S4).

To gain further insight into the model's properties, we examined the effect of state feedback control on the network dynamics. First, spectral analysis of the state transition matrix revealed that the distribution of eigenvalues contained complex pairs of eigenvalues associated with oscillatory behavior (Figure 3A). In the initial, randomly configured networks, many ei-

genvalues exhibited large absolute imaginary components corresponding to high-frequency oscillations and real parts close to the imaginary axis, indicating slow decay. Interestingly, the closed-form expression ($J-L_{net}$) modified the distribution by reducing the maximum absolute imaginary part of the eigenvalues and decreasing the spectral abscissa (Figures 3A and 3B).

Next, we analyzed the role of state feedback inputs in driving the network activity. After the GO cue, inputs displayed a smooth, time-varying profile that persisted beyond peak velocity (Figure 3C). Interestingly, input signals ($u(t)$) decayed faster than the network activity, suggesting that movement execution was supported by a sustained activation generated by the network's intrinsic dynamics. To evaluate its contribution, we examined both the unforced dynamics of the initial, randomly configured network (J) and the dynamics of the controlled network ($J-L_{net}$) after setting the input to zero following the GO cue. The activity of individual nodes revealed that the unforced dynamics were

characterized by high-frequency, low-amplitude oscillations that decayed more quickly than the feedback-driven activity (Figure 3D, left sub-panel). We quantified the normalized total activity across nodes and reach directions, scaled to the initial preparatory state. Networks with state feedback inputs showed a rise in activity followed by a slow decay, whereas unforced dynamics exhibited a much faster decay (Figure 3D, right sub-panel). These results demonstrate how state feedback inputs interact with intrinsic network dynamics to shape the network activity during reaching movements.

Variation in movement dynamics

Building on the validation of the model in the center-out reaching task (as shown in Figure 2), we now illustrate its predictive power. Indeed, we can change any task parameter value (i.e., change a subset of Θ) and interrogate the model about the resulting neural activities. Using paradigmatic examples from motor control literature (Figure 1D), we demonstrate how task parameters shape neural trajectories, enabling quantitative predictions about neural population dynamics across a range of tasks. Two subsets of task parameters are often studied: (1) the task objectives (the weighting matrices Q and R penalizing states or inputs in the cost function, expressed in Equation 2) and (2) the system dynamics (the matrix M). Based on the fact that the control gains, and hence the network inputs, are modulated as a function of the above-mentioned task parameters (Equation 3), we can explore how these parameters shaped the network dynamics.

We first examine the influence of velocity-dependent force fields, widely used to study motor adaptation,⁴⁸ on the network dynamics. A velocity-dependent force field of the form $f_x = \alpha \dot{y}$ and $f_y = \alpha \dot{x}$ modifies the limb dynamics by coupling forward speed with lateral force and lateral speed with forward force. Assuming that the controller has knowledge of this force field, the model was able to produce straight hand paths toward the target after the GO signal (Figure 4A), countering the applied lateral forces with scaled hand forces (Figure 4A, bottom-right). Note that we do not model the process of adaptation, but we assume that the controller knows the force-field parameter. Although no movement occurred prior to the GO cue, a few individual nodes encoded the level of the force-field parameter (α) during the preparatory phase, without generating motor output (Figure 4B). Hence, network activity in the output-null space already encodes the strength of the force-field parameter. This was followed by a scaling in the amplitude of rate responses in individual nodes during the movement epoch (Figure 4B).

At the population level, the dominant PCs encoded force-field strength during the preparatory epoch, followed by rotational trajectories in the movement epoch (Figure 4C). Notably, these rotations occupied different planes but maintained a consistent direction even when the force field was applied in opposite directions (-5 to $+5$).

Next, we quantified the changes in neural population responses across force fields by computing the path changes compared to the baseline parameter level ($\alpha = 0$) (Figure 4G). The change in preparatory state was measured as the distance between the preparatory state at the GO signal and at the baseline force-field parameter level. Movement-related changes in

PCs were quantified as the sum of squared differences between points on the PC trajectory and the baseline trajectory. Across different networks, both preparatory and movement period PCs scale with the strength of the force-field parameter (α) (Figure 4H). Indeed, analyzing the corresponding control gains across varying force-field values revealed that these gains changed with α during both preparatory and execution epochs, thereby shifting network activity into different preparatory states that scaled with the force-field strength even when moving to the same spatial target.

We next asked whether this modulation in neural trajectories extended to other changes in limb dynamics. In a second set of simulations, we varied inertial loads by simulating different masses of the hand while reaching to the target (Figure 4D). Similar to the force-field task, individual nodes encoded the inertial load level m_{load} (Figure 4E) during both preparatory and movement epochs. Applying PCA again showed that the dominant PCs in each epoch reflected the load magnitude (Figure 4F). As before, both preparatory and movement PCs scaled with the level of loading (Figure 4I), confirming that parameters altering limb dynamics produced feedback modulation that reshaped the network activity.

To assess the model properties with multiple targets and loading conditions, we simulated reaching tasks involving eight spatial targets under varying velocity-dependent force fields and inertial loads (Figure S5A). In both conditions, preparatory PCs exhibited affine encoding: preparatory states varied with the magnitude of the load (m_{load}) or force-field parameter (α), while offsets were determined by the target location (Figure S5B). Notably, the radial organization shown in preparatory states is consistent with the empirical evidence from motor adaptation studies.^{49,50} Under force field conditions, preparatory shifts occurred tangentially (toward or away from adjacent targets), whereas inertial loading induced radial scaling of preparatory states. During movement, PCs traced rotational trajectories that varied systematically with both target direction and mechanical loading (Figure S5C), indicating an interaction between spatial goals and limb dynamics. Quantitatively, increasing the angular separation between target directions was related to a greater distance between preparatory states in the low-dimensional, projected space (Figure S5D). Overall, varying the limb dynamics led to a combination of scaling and directional variations in the PCs of network population activity.

Variation in task objectives

We now use the same approach to show a modulation of population activity in response to variation in task objectives (varying Q matrix in Equation 3). To test this, we simulated two tasks typically used in human behavioral experiments: the bar task^{51,52} and a shooting (or a hit) task.⁵³

In the bar task, the hand must reach a target with different levels of penalty on the lateral endpoint deviation, leading to variation in the lateral end position (Figure 5A). When a brief mechanical load was applied to perturb the movement laterally, the model corrected errors more vigorously in directions where errors were more costly (high lateral cost), leading to smaller deviations and higher corrective forces (Figure 5A,

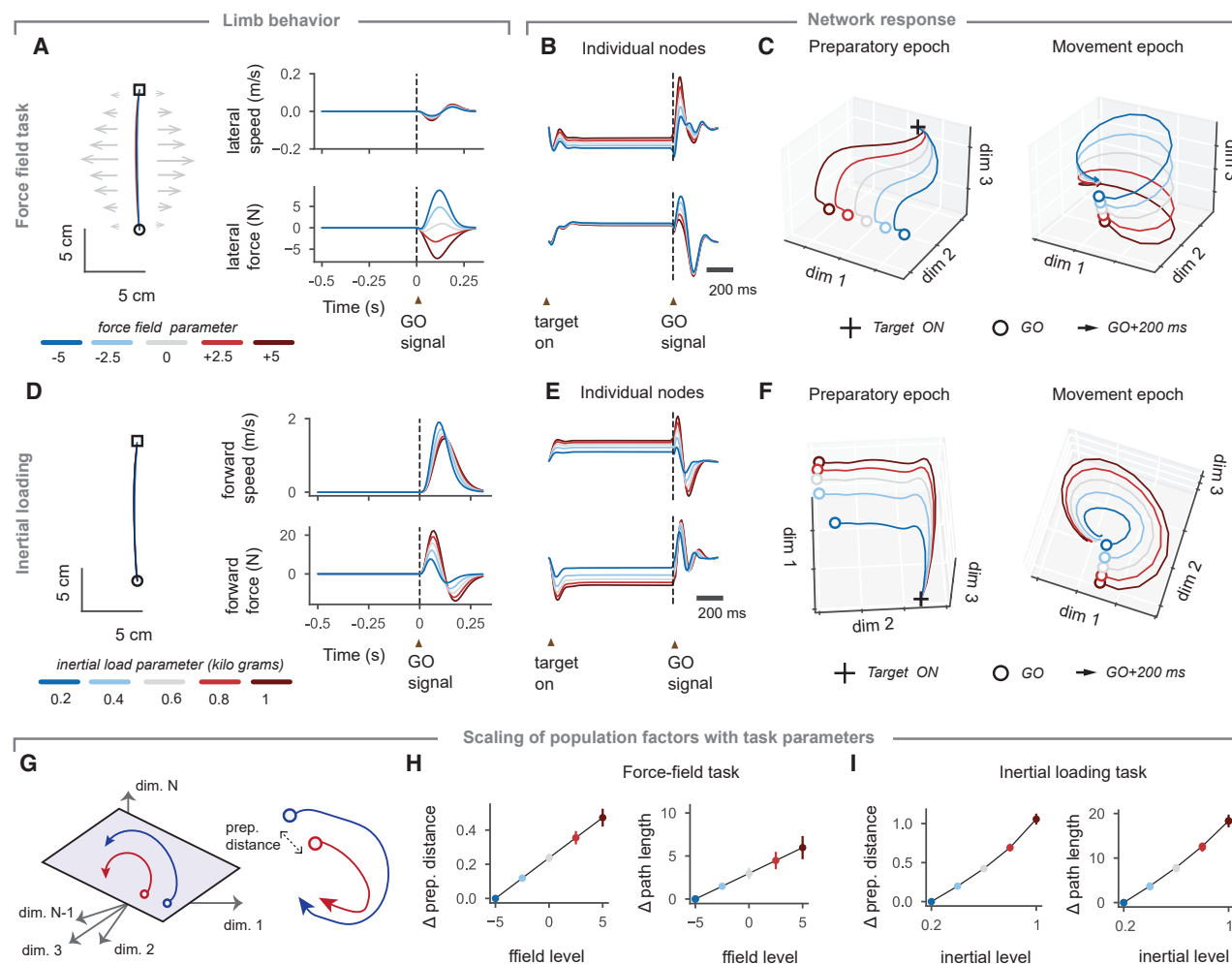


Figure 4. Flexible control of changes in biomechanical requirements

(A) Representation of the task with a clockwise or counterclockwise force field of different strengths and hand kinematics with varying levels of lateral force applied to counter the force field.

(B) Exemplar neuron firing rates.

(C) Projection of the network trajectories in the principal mode of the preparation and movement epochs for five different levels of orthogonal force-field parameter.

(D–F) Same as above for five different levels of limb inertia.

(G) Illustration of the key measures of variation in neural factors.

(H) Change in preparatory and movement-related factors for the force-field task.

(I) Same as (H) for inertial loading task. The circle markers depict the mean across $n = 5$ networks, and the error bars indicate the $\pm$ SEM.

See also Figure S5.

perturbed condition). In contrast, when the cost was low, it corrected less (smaller force, larger hand deviation) as expected under the minimal intervention principle.³⁴ Behaviorally, these error corrections recapitulate the effect of environmental factors shown by Nashed and colleagues,⁵¹ where the online correction was modulated by the target width. At the network level, despite the absence of motor output during preparation, some network nodes already encoded task objectives (i.e., varied as a function of the lateral cost parameter). At the population level, preparatory PCs separated as a function of the lateral cost parameter, followed by distinct trajectories during execution (Figure 5B).

In the shooting task, the penalty on terminal velocity of reaching the target was systematically reduced, leading to shooting movements that hit the target without stopping on target (Figure 5C). Notably, the differences in kinematics were minimal (Figure 5C), especially the differences in hand speed became apparent only near the end of the movement (Figure 5C top-right). At the population level, preparatory PCs reflected changes in the cost on terminal velocity before the actual movement, followed by distinct movement-related PCs (Figure 5D). Across different network instantiations, both preparatory and movement-related PC trajectories scaled with cost parameters for both tasks, demonstrating

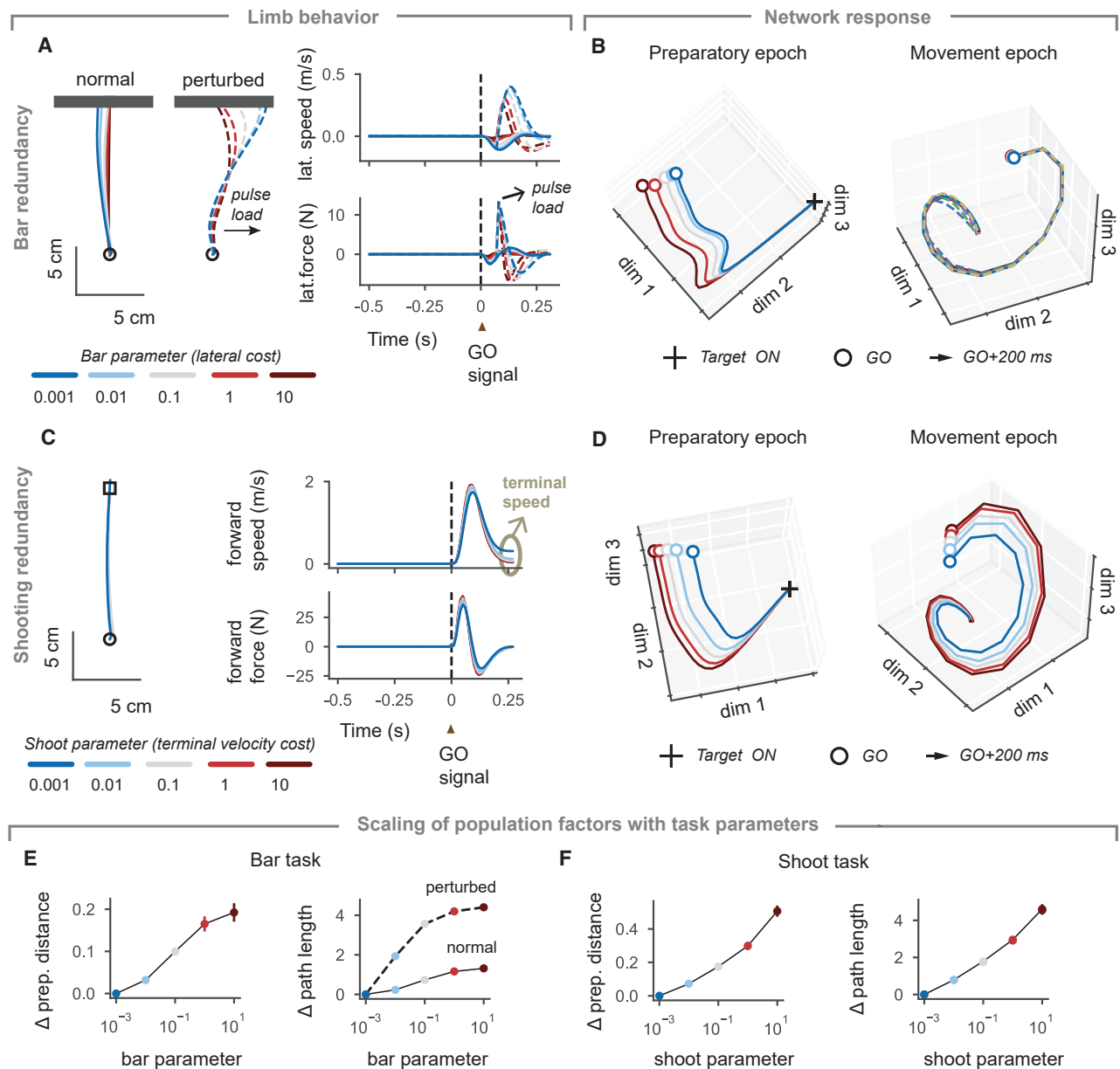


Figure 5. Flexible control of changes in task performance objectives

(A) Representation of the task with varying levels of penalty on the lateral endpoint coordinate, allowing modulation of the feedback response.

(B) Projection of the network trajectories on the three principal modes from preparatory and movement epochs for five different levels of lateral terminal displacement cost parameter.

(C and D) Same as above for five different levels of shooting parameter captured by a cost on terminal hand velocity.

(E) Change in preparatory and movement period factor properties for the bar task.

(F) Same as (E) for the shooting task. The circle markers depict the mean across $n = 5$ networks, and the error bars indicate the $\pm$ SEM.

See also Figure S6.

that these parameters can be extracted from the low-dimensional features of the network activity (Figures 5E and 5F). Movement PCs during perturbation showed larger path length modulation with the bar parameter (Figure 5E), highlighting how state feedback control integrates limb feedback to adjust network trajectories according to task objectives.

Interestingly, our model reproduced another unexpected feature of M1 activity linked to task-dependent consideration of limb feedback. We applied constant loads in eight radial directions during a prepare-to-reach task toward a target 20 cm from the start location (Figure S6). Loads were presented unexpectedly in two contexts: 500ms after target onset (preparatory epoch) and

100ms after the GO cue (execution epoch). In both cases, the controller rapidly corrected hand deviations. Simulations revealed that the magnitudes of individual nodes' responses to loading during preparatory (i.e., posture maintenance) and execution epochs were uncorrelated (Figures S6A and S6B). Such random changes in the load representations were consistent with the evidence of distinct cortical responses during posture maintenance and reach execution.⁵⁴ Note, however, that our model applied endpoint loads in cartesian space, unlike joint torques used in the above-mentioned experimental study.

Low rank hypothesis applied to network-body system

Our developments and simulations highlight a critical role of state feedback and task parameters in shaping population activity. First, the optimal input to the system clearly conveys feedback from the modeled limb into the network (Equation 3), which must influence the network trajectories. The analysis of the network's activity without feedback also confirmed the defining role of state feedback on the network trajectories (Figure 3). Next, our previous simulations show that both task parameters and perturbation responses together influence network trajectories (Figures 5E and S6). Note that we do not specify the source of inputs; in practice, biological systems must contend with delays and noise in sensory processing, and it is assumed that an estimate from sensory feedback or internal predictions is available.^{55,56} Now, we develop a more formal argument of why low-dimensional features emerge from limb feedback based on the spectral properties of the network-body system.

The dynamics of the full system (Equation 4) can be written compactly as $\dot{\mathbf{z}} = \mathbf{A}_c \mathbf{z}$, where $\mathbf{A}_c$ represents the combined system dynamics of the network, limb, and desired (target) limb states under closed-loop control. For simplicity, we ignored the constant input (ϵ) as it does not impact subsequent analyses. The matrix $\mathbf{A}_c$ has a block structure, consisting of a high-dimensional, partially random component ($\mathbf{A}_c^{(1)}$) and a low-dimensional component ($\mathbf{A}_c^{(2)}$) related to the limb feedback gains (Figure 6A, see STAR Methods).

indicates that when the spectral norm of $\mathbf{A}_c^{(1)}$ is sufficiently small, the singular values of the total connectivity matrix ($\mathbf{A}_c$) remain close to the singular values of its low-dimensional component ($\mathbf{A}_c^{(2)}$).⁴⁰ In this scenario, the total matrix ($\mathbf{A}_c$) also attains a low effective rank, meaning that a small number of dominant features capture most of the connectivity structure.

We examined the dimensionality of network structure under the application of feedback control in the center-out reaching task. In line with the low rank property (Equation 6), the singular values of the matrix $\mathbf{A}_c$ decreased rapidly and matched those of the limb-feedback connectivity the term $\mathbf{A}_c^{(2)}$ for the first four components (Figures 6A and 6B). Using a well-defined effective rank metric that is based on the uniformity of singular value distribution (see STAR Methods), we found that the dimensionality of the network structure closely tracked that of the limb-related input. Consequently, even though the system's initial connectivity was high-dimensional (*erank*: 69), the structure of the controller input forced low-dimensionality (*erank*: 3).

Thus, it is expected that the network activity will also be low-dimensional as its dimension is determined by the connectivity structure. Indeed, the effective rank of the network activity (STAR Methods) across time and targets was much lower (*erank*: 6.8) than the actual number of network nodes ($n = 100$) (Figure 6C).

The foregoing spectral analysis connected the rapid decay of the singular values of the system matrix $\mathbf{A}_c$ (i.e., its structural properties) to the low-dimensionality of the network activity (i.e., its dynamics). A direct corollary is that increasing the dimensionality of the input signal (through $\mathbf{A}_c^{(2)}$) containing task-relevant information should increase the effective rank of the network activity (dashed lines in Figure 6A). We tested this in two ways: (1) first, we increased task complexity by adding via-points along the movement path, and (2) then we modified the dimension of the plant by considering either a single joint system or a two-joint system.

We simulated the center-out reaching task with 0, 1, or 2 intermediate via-points. Although hand paths were generally similar

$$\underbrace{\begin{bmatrix} J - L_{net}(\Theta) & -L_{limb}(\Theta) \\ C & M \end{bmatrix}}_{\mathbf{A}_c} = \underbrace{\begin{bmatrix} J - L_{net}(\Theta) & 0 \\ C & M \end{bmatrix}}_{\mathbf{A}_c^{(1)}} + \underbrace{\begin{bmatrix} 0 & -L_{limb}(\Theta) \\ 0 & 0 \end{bmatrix}}_{\mathbf{A}_c^{(2)}}. \quad (\text{Equation 5})$$

Thibeault and colleagues⁴⁰ recently showed that the connectivity of many natural physical systems can be modeled as a combination of a random component and a low-dimensional component. They used Weyl's inequality to show such systems are effectively low dimensional, provided the spectrum of the random component is small in comparison with the low-dimensional component of the overall system connectivity. By rewriting the Weyl's inequality from Thibeault et al.⁴⁰ while considering Equation 5, we get:

$$|\sigma_i(\mathbf{A}_c) - \sigma_i(\mathbf{A}_c^{(2)})| \leq \|\mathbf{A}_c^{(1)}\|_2 \quad (\text{Equation 6})$$

for all $i \in \{1, 2, \dots, n\}$, where σ_i is the i^{th} singular value and $\|\cdot\|_2$ is the spectral norm (largest singular value) of a matrix. The inequality

(Figure 6D), the velocity profiles showed additional sub-movements as the number of via-points increased. We observed two key changes indicating an increase in the dimensionality of feedback input signals. First, the effective rank of the network connectivity component associated with the limb feedback ($\mathbf{A}_c^{(2)}$) increased with the number of via-points (Figure S7A). Second, the time series of network inputs also showed an increase in effective rank (Figure S7B). Given that the dimensionality of the network activity is tightly linked to the structure of limb feedback, we examined the changes in the singular values of the network activity (STAR Methods). As task complexity increased, the singular value spectrum decayed more slowly (Figure 6E). Consequently, the effective rank, equivalent to the dimensionality of

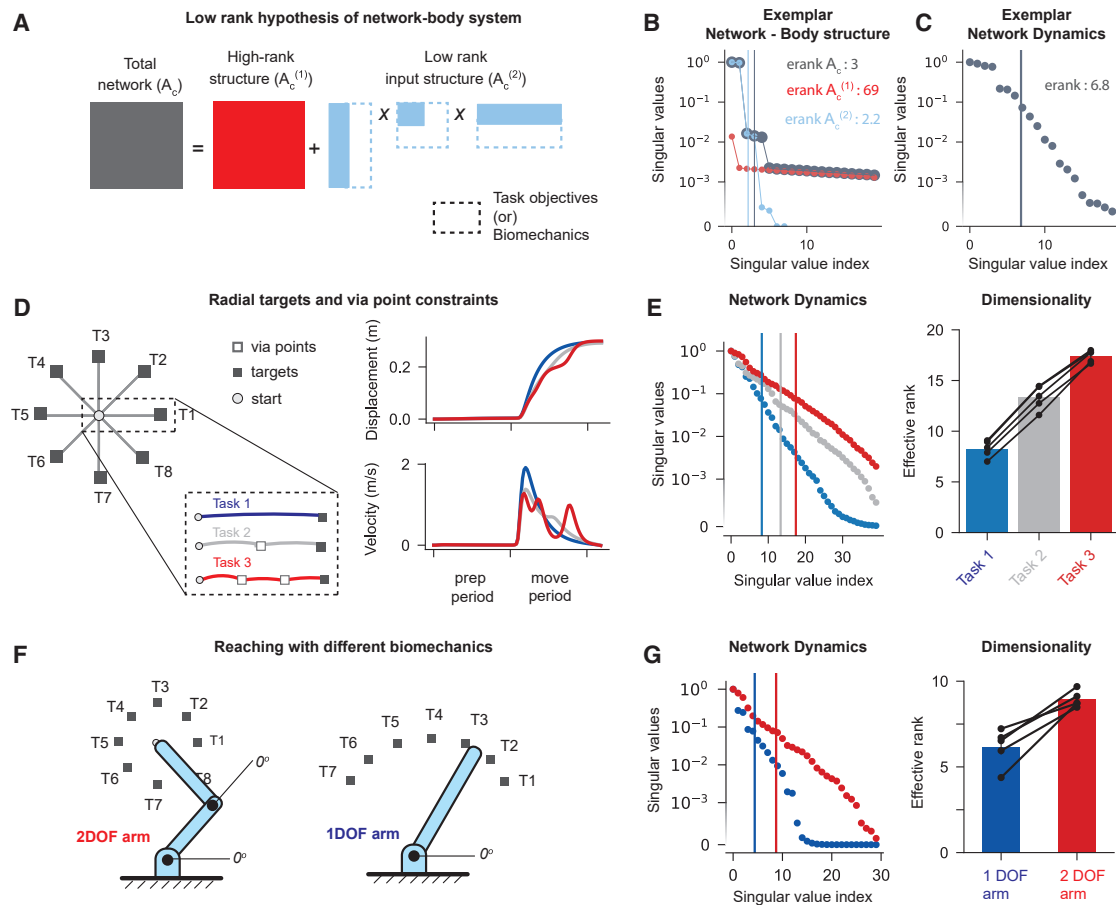


Figure 6. Testing low-rank hypothesis in the interacting network-body system

(A) Decomposing network structure into the default connectivity and the limb input structure resulting from the structure of the LQR controller. We indicated the low-dimensional structure of $A_c^{(2)}$ by illustrating the singular value decomposition of this matrix. The dashed lines indicate that the rank of the low-dimensional components can change depending on the task definition and dimension of the effector. See main text (Equation 5) for the mathematical description of these block matrices.

(B) Singular values of the composite matrix (in gray), of the default connectivity (in red), and the limb input matrix (in cyan) of an exemplar network.

(C) Singular values of exemplar network activity (dynamics) matrix obtained after concatenation of each node's temporal activity across the 8 different targets (STAR Methods). Observe that the effective rank (<7) was low in comparison with the network size ($n = 100$).

(D) Illustration of the task with a via-point used to modulate the complexity (left subpanel). Simulated hand displacement and velocity to a horizontal target in the three via-point tasks (right subpanels).

(E) Singular values of the exemplar network activity in the movement period (left) and effective ranks for five different instances of the model with different networks (right). The vertical lines are the effective ranks, and the horizontal axis was truncated for clarity. The light gray lines on the bar plots correspond to individual network instances.

(F) Illustration of the eight-directional reaching task with a 2-DOF and a 1-DOF arm.

(G) Singular values of the exemplar network activity in the movement period (left) and effective ranks for five different instances of the model with different networks (right). Data are represented as mean $\pm$ SEM across $n = 5$ networks.

See also Figures S7 and S8 and Methods S1.

the network activities, increased with task complexity (Figure 6E), although the network structure, eight reach directions, and movement duration remained unchanged. This suggests that increasing task demands selectively induced higher-dimensional feedback input signals, thereby increasing the dimensionality of the associated network dynamics.

We next varied limb biomechanics by simulating movements with either a one degree of freedom (1-DOF) or a two degree of freedom (2-DOF) limb to reach eight radial targets (Figure 6F). The number of DOF directly modified the size of the state vector

and of the feedback gain matrix $L_{limb}(\Theta)$. Again, the effective rank of the network connectivity component associated with limb feedback and the time series of network inputs increased with the added biomechanical dimension (Figures S7C and S7D). Singular value decomposition of the network activity (STAR Methods) showed that the 1-DOF simulations exhibited a steeper singular value decay, resulting in a lower effective rank than the 2-DOF simulations (Figure 6G). Thus, increasing the biomechanical dimensions also increased the dimensionality of the corresponding network dynamics.

Together, these results support the view that low-rank motor cortical dynamics emerge naturally from optimal control of the network-body system, and provide a formal connection between the dimensionality of the input to the network, such as the number of DOFs or the presence of intermediate goals and the neural population activity. Finally, as a corollary of the low-rank hypothesis described above, we observed that for similar tasks and biomechanics, and therefore similar inputs to networks, the PCs were preserved across variations in the network connectivity (Figure S8). This observation is consistent with the observed similarity in neural population trajectories across individuals performing a task,⁵⁷ assuming that different brains can be characterized by a similar network topology.

DISCUSSION

Our results demonstrate that several features of neural population dynamics naturally emerge from optimal control of a simple, linear network-body system. By coupling a high-dimensional random network to a biomechanical plant and computing optimal control gains for each task, we replicated several hallmark features of motor cortical activity, including rotational dynamics,^{8,38} preparatory states encoding the parameters of impending movements,^{5,37} and orthogonal preparatory and execution subspaces.³⁹ Our model captured the fact that population activity reflects variations in task parameters, such as altered force fields,^{49,58} inertial load,⁵⁹ or task demands before movement onset. Finally, we demonstrated that this linear formulation establishes the critical role of the input to the network (here, the state-feedback control signal) in the shaping of the network dynamics⁶⁰ based on the low-rank hypothesis of complex systems.⁴⁰

The model architecture corresponds to assumptions that are consistent with experimental evidence from motor cortical recordings. First, the relationship between network and muscle activation is fixed (Equation 1), which is consistent with the evidence that neural activation patterns exhibit a fixed relationship with muscle activation when changing arm posture or adapting to a force field.^{61,62} Likewise, network units exhibited changes in response magnitudes dependent on the task.⁵⁴ However, we did not model the effect of delays and nonlinearities that are known to determine neural responses during reaching and mechanical loading.^{13,63} These simplifying assumptions are not fundamental limitations since they can be addressed in more complex instances of the network-body model by modifying the block structure of the network,⁶⁴ including nonlinearities in the muscle and body dynamics^{13,65} or adding delays and noise following standard techniques.^{66,67} In this light, the simplifying assumptions discussed above can be seen as an asset: it allowed us to comprehensively demonstrate the remarkable explanatory power of the simplest instance of the model, thereby showing that the first basic, linear approximation already provides deep insight along with testable predictions about the shape and dimensions of neural factors that one may expect for any experiment that could be modeled in this framework.

Our approach differs from conventional neural network models, which rely on data-driven training,^{9–13,15–17,19,68–72} with task-dependent network trajectories that can be recalled

by contextual inputs²⁹ when similar situations arise. Here, because we formulated a linear model, the optimal network weights could be calculated analytically as optimal feedback control gains, modifying both the way state feedback was fed into the network as well as how different network nodes needed to be decoupled. This approach offers an original interpretation of how optimal controllers can be synthesized in the brain: variations in model parameters (Θ) yield distinct feedback control gains that modify input to the system and coupling of network nodes. Thus, task selection (e.g., hold then move),^{39,54} task-dependent control,^{15,73–79} and motor adaptation^{49,59,80} can in principle be achieved in a fixed-network architecture by efficiently modulating the projections of somatosensory signals into the network according to the task parameters.

This view provides a mechanism by which humans can express flexible, task-dependent control across fast timescales, i.e., between or even within trials.^{20,21,53,81–84} While connectivity may be reshaped over longer timescales through synaptic plasticity,^{14,70,85,86} rapid adjustments may be plausibly explained by swiftly modifying feedback control inputs through a fixed network architecture.^{31,32} We do not rule out that real sensorimotor circuits may develop specialized connectivity for commonly encountered tasks over time,⁸⁷ owing to plasticity in cortical and subcortical regions.⁸⁸ Rather, we propose that such experience-driven plasticity coexists with fast, task-dependent modulation of feedback-driven input to the network, enabling primates to deploy flexible control policies across a wide range of contexts very rapidly.

Another key insight offered by the linear formulation was that low-dimensional neural factors can emerge because limb and task parameters provide a dominant, low-rank input to the otherwise high-dimensional recurrent network. This aligns with the low-rank hypothesis of complex systems,⁴⁰ which posits that a small subset of principal modes embedded within a large random network yields a low effective rank, making the system trajectories low-dimensional. The block structure of the network-body dynamics and the expression of optimal control gains (Equation 5) allowed decomposing the closed-loop system into components linked to the network or feedback input and studying the spectral properties of each component as well as their effect on the dimension of the network trajectories. A direct consequence of this relationship was that the dimensionality of neural trajectories scaled with task complexity or with the dimensionality of the effector (Figure 6, via points and degrees of freedom, respectively). Overall, this model portrays motor cortex as an input-driven dynamical system, consistent with experimental evidence,^{60,89–93} while additionally highlighting the role of state feedback and task parameters in the computation of optimal network inputs.

In all, this study shows that linear models as a first approximation can provide a powerful test bed to make interpretable predictions about the shape and dimensions of neural trajectories, providing a candidate theoretical link between neural activity and sensorimotor control. This framework predicts that variations in task parameters lead to measurable changes in population activities, reflecting distinct control policies and biomechanical constraints. These normative insights may inform fundamental research and brain-machine development to

overcome the challenges posed by increasingly large neural datasets.

Limitations of the study

Clearly, the model was based on very simplifying assumptions: we assumed a fully observable state, direct readout for the actuation signals, and perfect knowledge of the network connectivity to derive the optimal controller. We believe that these assumptions were justified to formulate the basic properties of such a network-body system in the simplest case, and expect that future works will unveil the properties of more biologically plausible models in detail. Clearly, considering a partially observable state (i.e., with delays and noise⁹⁴) and indirect connections between the body and the network may enrich the predictions with insight into distributed processes across different brain regions,^{11,76,90} including cerebellum,^{89,95} somatosensory and parietal areas,^{67,90,96,97} and spinal cord.^{16,27,59} Likewise, the question of robustness against imperfect knowledge of the network structure in the controller is an important challenge for prospective studies.

RESOURCE AVAILABILITY

Lead contact

Requests for further information and resources should be directed to and will be fulfilled by the lead contact, Frédéric Crevecoeur (frederic.crevecœur@uclouvain.be).

Materials availability

This study did not generate any new materials.

Data and code availability

- Data from numerical simulations have been deposited at the Figshare repository and are publicly available. The DOI is listed in the [key resources table](#).
- This paper analyzes existing, publicly available data of neural recordings from monkey motor cortex. The DOIs are listed in the [key resources table](#).
- The original code for running the model and analyzing the results has been deposited in GitHub (and Zenodo) and is publicly available. The access link and the DOI are listed in the [key resources table](#).
- Any additional information required to reanalyze the data reported in this paper is available from the [lead contact](#) upon request.

ACKNOWLEDGMENTS

We thank Dr. Kevin Cross, Dr. Laureline Loggiaco, and our lab members for providing valuable feedback on the earlier versions of the manuscript. We thank Jessica Brand for her contribution to the development of the graphical illustration and the schematic. H.T.K. was supported by the Fonds de la Recherche Scientifique (FRS-FNRS) Chargé de recherche grant CR 252 (FC 043127). F.C. was supported by the F.R.S. - FNRS grant 1.C.033.18 (FC 036239). This work was additionally supported by a Concerted Research Action of Université Catholique de Louvain (ARC; “coaction”).

AUTHOR CONTRIBUTIONS

Conceptualization, H.T.K. and F.C.; methodology, H.T.K. and F.C.; investigation, H.T.K.; data curation: H.T.K.; formal analysis: H.T.K.; writing – original draft, H.T.K. and F.C.; writing – review & editing, H.T.K. and F.C.; funding acquisition, H.T.K. and F.C.; resources, F.C.; and supervision, F.C.

DECLARATION OF INTERESTS

The authors declare no conflict of interest.

DECLARATION OF GENERATIVE AI AND AI-ASSISTED TECHNOLOGIES IN THE WRITING PROCESS

During the preparation of this work, the author(s) used Grammarly and Gemini to check the grammar and refine the text. After using this tool or service, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

STAR★METHODS

Detailed methods are provided in the online version of this paper and include the following:

- **KEY RESOURCES TABLE**
- **METHOD DETAILS**
 - Network dynamics
 - Limb dynamics
 - The limb state evolves according to the differential equation $\dot{q} = Mq(t) + Fa(t)$, with
 - Receding horizon control in a Linear Quadratic Regulator (LQR) form
 - Prepare-to-reach task
 - Changes in limb mechanics
 - Changes in task objectives
- **QUANTIFICATION AND STATISTICAL ANALYSIS**
 - PCA in preparatory and movement epochs
 - Alignment index
 - jPCA analysis
 - Quantifying rotations and decay in data
 - Tensor maximum entropy (TME)
 - Singular value decomposition analysis
 - Dimensionality

SUPPLEMENTAL INFORMATION

Supplemental information can be found online at <https://doi.org/10.1016/j.celrep.2026.116991>.

Received: June 2, 2025

Revised: October 24, 2025

Accepted: January 20, 2026

Published: February 12, 2026

REFERENCES

1. Krakauer, J.W., Ghazanfar, A.A., Gomez-Marín, A., MacIver, M.A., and Poeppel, D. (2017). Neuroscience Needs Behavior: Correcting a Reductionist Bias. *Neuron* 93, 480–490. <https://doi.org/10.1016/j.neuron.2016.12.041>.
2. Urai, A.E., Doiron, B., Leifer, A.M., and Churchland, A.K. (2022). Large-scale neural recordings call for new insights to link brain and behavior. *Nat. Neurosci.* 25, 11–19. <https://doi.org/10.1038/s41593-021-00980-9>.
3. Gao, P., and Ganguli, S. (2015). On simplicity and complexity in the brave new world of large-scale neuroscience. *Curr. Opin. Neurobiol.* 32, 148–155. <https://doi.org/10.1016/j.conb.2015.04.003>.
4. Stringer, C., and Pachitariu, M. (2024). Analysis methods for large-scale neuronal recordings. *Science* 386, eadp7429. <https://doi.org/10.1126/science.adp7429>.
5. Churchland, M.M., and Shenoy, K.V. (2024). Preparatory activity and the expansive null-space. *Nat. Rev. Neurosci.* 25, 213–236. <https://doi.org/10.1038/s41583-024-00796-z>.
6. Vyas, S., Golub, M.D., Sussillo, D., and Shenoy, K.V. (2020). Computation Through Neural Population Dynamics. *Annu. Rev. Neurosci.* 43, 249–275. <https://doi.org/10.1146/annurev-neuro-092619-094115>.

7. Gallego, J.A., Perich, M.G., Miller, L.E., and Solla, S.A. (2017). Neural Manifolds for the Control of Movement. *Neuron* 94, 978–984. <https://doi.org/10.1016/j.neuron.2017.05.025>.
8. Churchland, M.M., Cunningham, J.P., Kaufman, M.T., Foster, J.D., Nuyujukian, P., Ryu, S.I., and Shenoy, K.V. (2012). Neural population dynamics during reaching. *Nature* 487, 51–56. <https://doi.org/10.1038/nature11129>.
9. Sussillo, D., Churchland, M.M., Kaufman, M.T., and Shenoy, K.V. (2015). A neural network that finds a naturalistic solution for the production of muscle activity. *Nat. Neurosci.* 18, 1025–1033. <https://doi.org/10.1038/nn.4042>.
10. Feulner, B., Perich, M.G., Chowdhury, R.H., Miller, L.E., Gallego, J.A., and Clopath, C. (2022). Small, correlated changes in synaptic connectivity may facilitate rapid motor learning. *Nat. Commun.* 13, 5163. <https://doi.org/10.1038/s41467-022-32646-w>.
11. Michaels, J.A., Schaffelhofer, S., Agudelo-Toro, A., and Scherberger, H. (2020). A goal-driven modular neural network predicts parietofrontal neural dynamics during grasping. *Proc. Natl. Acad. Sci. USA* 117, 32124–32135. <https://doi.org/10.1073/pnas.2005087117>.
12. Kalidindi, H.T., Cross, K.P., Lillicrap, T.P., Omrani, M., Falotico, E., Sabes, P.N., and Scott, S.H. (2021). Rotational dynamics in motor cortex are consistent with a feedback controller. *eLife* 10, e67256. <https://doi.org/10.7554/eLife.67256>.
13. Lillicrap, T.P., and Scott, S.H. (2013). Preference Distributions of Primary Motor Cortex Neurons Reflect Control Solutions Optimized for Limb Biomechanics. *Neuron* 77, 168–179. <https://doi.org/10.1016/j.neuron.2012.10.041>.
14. Gurnani, H., Liu, W., and Brunton, B.W. (2024). Feedback control of recurrent dynamics constrains learning timescales during motor adaptation. Preprint at bioRxiv. <https://doi.org/10.1101/2024.05.24.595772>.
15. Saxena, S., Russo, A.A., Cunningham, J., and Churchland, M.M. (2022). Motor cortex activity across movement speeds is predicted by network-level strategies for generating muscle activity. *eLife* 11, e67620. <https://doi.org/10.7554/eLife.67620>.
16. Guang, H., Nashed, J.Y., Pruszyński, J.A., Kalidindi, H.T., Crevecoeur, F., Cross, K.P., Blohm, G., and Scott, S.H. (2024). Parallel feedback processing for voluntary control: knowledge of spinal feedback in motor cortex. Preprint at bioRxiv. <https://doi.org/10.1101/2024.05.12.593756>.
17. Aldarondo, D., Merel, J., Marshall, J.D., Hasenclever, L., Klibaite, U., Gelis, A., Tassa, Y., Wayne, G., Botvinick, M., and Ölveczky, B.P. (2024). A virtual rodent predicts the structure of neural activity across behaviours. *Nature* 632, 594–602. <https://doi.org/10.1038/s41586-024-07633-4>.
18. Zador, A., Escola, S., Richards, B., Ölveczky, B., Bengio, Y., Boahen, K., Botvinick, M., Chklovskii, D., Churchland, A., Clopath, C., et al. (2023). Catalyzing next-generation Artificial Intelligence through NeuroAI. *Nat. Commun.* 14, 1597. <https://doi.org/10.1038/s41467-023-37180-x>.
19. Codol, O., Krishna, N.H., Lajoie, G., and Perich, M.G. (2024). Brain-like neural dynamics for behavioral control develop through reinforcement learning. Preprint at bioRxiv. <https://doi.org/10.1101/2024.10.04.616712>.
20. Braun, D.A., Aertsen, A., Wolpert, D.M., and Mehring, C. (2009). Learning Optimal Adaptation Strategies in Unpredictable Motor Tasks. *J. Neurosci.* 29, 6472–6478. <https://doi.org/10.1523/JNEUROSCI.3075-08.2009>.
21. Crevecoeur, F., Mathew, J., Bastin, M., and Lefèvre, P. (2020). Feedback Adaptation to Unpredictable Force Fields in 250 ms. *eNeuro* 7, 0400–0419. <https://doi.org/10.1523/ENEURO.0400-19.2020>.
22. Kalidindi, H.T., and Crevecoeur, F. (2023). Human reaching control in dynamic environments. *Curr. Opin. Neurobiol.* 83, 102810. <https://doi.org/10.1016/j.conb.2023.102810>.
23. Lake, B.M., Ullman, T.D., Tenenbaum, J.B., and Gershman, S.J. (2016). Building Machines That Learn and Think Like People. Preprint at arXiv. <https://doi.org/10.48550/arXiv.1604.00289>.
24. Saxe, A., Nelli, S., and Summerfield, C. (2021). If deep learning is the answer, what is the question? *Nat. Rev. Neurosci.* 22, 55–67. <https://doi.org/10.1038/s41583-020-00395-8>.
25. Beiran, M., Meirhaeghe, N., Sohn, H., Jazayeri, M., and Ostoic, S. (2023). Parametric control of flexible timing through low-dimensional neural manifolds. *Neuron* 111, 739–753.e8. <https://doi.org/10.1016/j.neuron.2022.12.016>.
26. Kao, T.-C., Sadabadi, M.S., and Hennequin, G. (2021). Optimal anticipatory control as a theory of motor preparation: A thalamo-cortical circuit model. *Neuron* 109, 1567–1581.e12. <https://doi.org/10.1016/j.neuron.2021.03.009>.
27. Lindén, H., Petersen, P.C., Vestergaard, M., and Berg, R.W. (2022). Movement is governed by rotational neural dynamics in spinal motor networks. *Nature* 610, 526–531. <https://doi.org/10.1038/s41586-022-05293-w>.
28. Logiaco, L., Abbott, L.F., and Escola, S. (2021). Thalamic control of cortical dynamics in a model of flexible motor sequencing. *Cell Rep.* 35, 109090. <https://doi.org/10.1016/j.celrep.2021.109090>.
29. Mante, V., Sussillo, D., Shenoy, K.V., and Newsome, W.T. (2013). Context-dependent computation by recurrent dynamics in prefrontal cortex. *Nature* 503, 78–84. <https://doi.org/10.1038/nature12742>.
30. Remington, E.D., Narain, D., Hosseini, E.A., and Jazayeri, M. (2018). Flexible Sensorimotor Computations through Rapid Reconfiguration of Cortical Dynamics. *Neuron* 98, 1005–1019.e5. <https://doi.org/10.1016/j.neuron.2018.05.020>.
31. Schimel, M., Kao, T.-C., and Hennequin, G. (2024). When and why does motor preparation arise in recurrent neural network models of motor control? *eLife* 12. <https://doi.org/10.7554/eLife.89131>.
32. Sohn, H., Meirhaeghe, N., Rajalingham, R., and Jazayeri, M. (2021). A Network Perspective on Sensorimotor Learning. *Trends Neurosci.* 44, 170–181. <https://doi.org/10.1016/j.tins.2020.11.007>.
33. Stroud, J.P., Porter, M.A., Hennequin, G., and Vogels, T.P. (2018). Motor primitives in space and time via targeted gain modulation in cortical networks. *Nat. Neurosci.* 21, 1774–1783. <https://doi.org/10.1038/s41593-018-0276-0>.
34. Todorov, E., and Jordan, M.I. (2002). Optimal Feedback Control as a Theory of Motor Coordination. *Nat. Neurosci.* 5, 1226–1235.
35. Åström, K.J. (1970). *Introduction to Stochastic Control Theory* (Academic Press).
36. Kaufman, M.T., Churchland, M.M., Ryu, S.I., and Shenoy, K.V. (2014). Cortical activity in the null space: permitting preparation without movement. *Nat. Neurosci.* 17, 440–448. <https://doi.org/10.1038/nn.3643>.
37. Churchland, M.M., Cunningham, J.P., Kaufman, M.T., Ryu, S.I., and Shenoy, K.V. (2010). Cortical Preparatory Activity: Representation of Movement or First Cog in a Dynamical Machine? *Neuron* 68, 387–400. <https://doi.org/10.1016/j.neuron.2010.09.015>.
38. Sabatini, D.A., and Kaufman, M.T. (2024). Reach-dependent reorientation of rotational dynamics in motor cortex. *Nat. Commun.* 15, 7007. <https://doi.org/10.1038/s41467-024-51308-7>.
39. Elsayed, G.F., Lara, A.H., Kaufman, M.T., Churchland, M.M., and Cunningham, J.P. (2016). Reorganization between preparatory and movement population responses in motor cortex. *Nat. Commun.* 7, 13239. <https://doi.org/10.1038/ncomms13239>.
40. Thibeault, V., Allard, A., and Desrosiers, P. (2024). The low-rank hypothesis of complex systems. *Nat. Phys.* 20, 294–302. <https://doi.org/10.1038/s41567-023-02303-0>.
41. Todorov, E. (2005). Stochastic Optimal Control and Estimation Methods Adapted to the Noise Characteristics of the Sensorimotor System. *Neural Comput.* 17, 1084–1108. <https://doi.org/10.1162/0899766053491887>.
42. Guigon, E. (2023). A computational theory for the production of limb movements. *Psychol. Rev.* 130, 23–51. <https://doi.org/10.1037/rev0000323>.

43. Maciejowski, J.M. (2002). *Predictive Control: With Constraints* (Pearson Education).
44. Lara, A.H., Cunningham, J.P., and Churchland, M.M. (2018). Different population dynamics in the supplementary motor area and motor cortex during reaching. *Nat. Commun.* 9, 2754. <https://doi.org/10.1038/s41467-018-05146-z>.
45. Suresh, A.K., Goodman, J.M., Okorokova, E.V., Kaufman, M., Hatsopoulos, N.G., and Bensmaia, S.J. (2020). Neural population dynamics in motor cortex are different for reach and grasp. *eLife* 9, e58848. <https://doi.org/10.7554/eLife.58848>.
46. Kuzmina, E., Kriukov, D., and Lebedev, M. (2024). Neuronal travelling waves explain rotational dynamics in experimental datasets and modeling. *Sci. Rep.* 14, 3566. <https://doi.org/10.1038/s41598-024-53907-2>.
47. Russo, A.A., Bittner, S.R., Perkins, S.M., Seely, J.S., London, B.M., Lara, A.H., Miri, A., Marshall, N.J., Kohn, A., Jessell, T.M., et al. (2018). Motor Cortex Embeds Muscle-like Commands in an Untangled Population Response. *Neuron* 97, 953–966.e8. <https://doi.org/10.1016/j.neuron.2018.01.004>.
48. Thoroughman, K.A., and Shadmehr, R. (2000). Learning of Action Through Adaptive Combination of Motor Primitives. *Nature* 407, 742–747. <https://doi.org/10.1038/35037588>.
49. Sun, X., O'Shea, D.J., Golub, M.D., Trautmann, E.M., Vyas, S., Ryu, S.I., and Shenoy, K.V. (2022). Cortical preparatory activity indexes learned motor memories. *Nature* 602, 274–279. <https://doi.org/10.1038/s41586-021-04329-x>.
50. Vyas, S., Even-Chen, N., Stavisky, S.D., Ryu, S.I., Nuyujukian, P., and Shenoy, K.V. (2018). Neural Population Dynamics Underlying Motor Learning Transfer. *Neuron* 97, 1177–1186.e3. <https://doi.org/10.1016/j.neuron.2018.01.040>.
51. Nashed, J.Y., Crevecoeur, F., and Scott, S.H. (2012). Influence of the behavioral goal and environmental obstacles on rapid feedback responses. *J. Neurophysiol.* 108, 999–1009. <https://doi.org/10.1152/jn.01089.2011>.
52. Gallivan, J.P., Logan, L., Wolpert, D.M., and Flanagan, J.R. (2016). Parallel specification of competing sensorimotor control policies for alternative action options. *Nat. Neurosci.* 19, 320–326. <https://doi.org/10.1038/nn.4214>.
53. Česonis, J., and Franklin, D.W. (2022). Contextual cues are not unique for motor learning: Task-dependant switching of feedback controllers. *PLoS Comput. Biol.* 18, e1010192. <https://doi.org/10.1371/journal.pcbi.1010192>.
54. Kurtzer, I., Herter, T.M., and Scott, S.H. (2005). Random change in cortical load representation suggests distinct control of posture and movement. *Nat. Neurosci.* 8, 498–504. <https://doi.org/10.1038/nn1420>.
55. Franklin, D.W., and Wolpert, D.M. (2011). Computational Mechanisms of Sensorimotor Control. *Neuron* 72, 425–442. <https://doi.org/10.1016/j.neuron.2011.10.006>.
56. Shadmehr, R., and Krakauer, J.W. (2008). A computational neuroanatomy for motor control. *Exp. Brain Res.* 185, 359–381. <https://doi.org/10.1007/s00221-008-1280-5>.
57. Safaie, M., Chang, J.C., Park, J., Miller, L.E., Dudman, J.T., Perich, M.G., and Gallego, J.A. (2023). Preserved neural dynamics across animals performing similar behaviour. *Nature* 623, 765–771. <https://doi.org/10.1038/s41586-023-06714-0>.
58. Ahmadi-Pajouh, M.A., Towhidkha, F., and Shadmehr, R. (2012). Preparing to Reach: Selecting an Adaptive Long-Latency Feedback Controller. *J. Neurosci.* 32, 9537–9545. <https://doi.org/10.1523/JNEUROSCI.4275-11.2012>.
59. Kirk, E.A., Hope, K.T., Sober, S.J., and Sauerbrei, B.A. (2024). An output-null signature of inertial load in motor cortex. *Nat. Commun.* 15, 7309. <https://doi.org/10.1038/s41467-024-51750-7>.
60. Sauerbrei, B.A., Guo, J.-Z., Cohen, J.D., Mischiati, M., Guo, W., Kabra, M., Verma, N., Mensh, B., Branson, K., and Hantman, A.W. (2020). Cortical pattern generation during dexterous movement is input-driven. *Nature* 577, 386–391. <https://doi.org/10.1038/s41586-019-1869-9>.
61. Cherian, A., Fernandes, H.L., and Miller, L.E. (2013). Primary motor cortical discharge during force field adaptation reflects muscle-like dynamics. *J. Neurophysiol.* 110, 768–783. <https://doi.org/10.1152/jn.00109.2012>.
62. Scott, S.H., Sergio, L.E., and Kalaska, J.F. (1997). Reaching Movements With Similar Hand Paths but Different Arm Orientations. II. Activity of Individual Cells in Dorsal Premotor Cortex and Parietal Area 5. *J. Neurophysiol.* 78, 2413–2426. <https://doi.org/10.1152/jn.1997.78.5.2413>.
63. Pruszynski, J.A., Kurtzer, I., Nashed, J.Y., Omrani, M., Brouwer, B., and Scott, S.H. (2011). Primary motor cortex underlies multi-joint integration for fast feedback control. *Nature* 478, 387–390. <https://doi.org/10.1038/nature10436>.
64. Perich, M.G., Arlt, C., Soares, S., Young, M.E., Mosher, C.P., Minxha, J., Carter, E., Rutishauser, U., Rudebeck, P.H., Harvey, C.D., et al. (2021). Inferring brain-wide interactions using data-constrained recurrent neural network models. Preprint at bioRxiv. <https://doi.org/10.1101/2020.12.18.423348>.
65. Trainin, E., Meir, R., and Karniel, A. (2007). Explaining Patterns of Neural Activity in the Primary Motor Cortex Using Spinal Cord and Limb Biomechanics Models. *J. Neurophysiol.* 97, 3736–3750. <https://doi.org/10.1152/jn.01064.2006>.
66. Crevecoeur, F., Munoz, D.P., and Scott, S.H. (2016). Dynamic Multisensory Integration: Somatosensory Speed Trumps Visual Accuracy during Feedback Control. *J. Neurosci.* 36, 8598–8611. <https://doi.org/10.1523/JNEUROSCI.0184-16.2016>.
67. Takei, T., Lomber, S.G., Cook, D.J., and Scott, S.H. (2021). Transient deactivation of dorsal premotor cortex or parietal area 5 impairs feedback control of the limb in macaques. *Curr. Biol.* 31, 1476–1487.e5. <https://doi.org/10.1016/j.cub.2021.01.049>.
68. Zimnik, A.J., and Churchland, M.M. (2021). Independent generation of sequence elements by motor cortex. *Nat. Neurosci.* 24, 412–424. <https://doi.org/10.1038/s41593-021-00798-5>.
69. Michaels, J.A., Dann, B., and Scherberger, H. (2016). Neural Population Dynamics during Reaching Are Better Explained by a Dynamical System than Representational Tuning. *PLoS Comput. Biol.* 12, e1005175. <https://doi.org/10.1371/journal.pcbi.1005175>.
70. Feulner, B., Perich, M.G., Miller, L.E., Clopath, C., and Gallego, J.A. (2025). A neural implementation model of feedback-based motor learning. *Nat. Commun.* 16, 1805. <https://doi.org/10.1038/s41467-024-54738-5>.
71. Merel, J., Botvinick, M., and Wayne, G. (2019). Hierarchical motor control in mammals and machines. *Nat. Commun.* 10, 5489. <https://doi.org/10.1038/s41467-019-13239-6>.
72. Chiappa, A.S., Tano, P., Patel, N., Ingster, A., Pouget, A., and Mathis, A. (2024). Acquiring musculoskeletal skills with curriculum-based reinforcement learning. *Neuron* 112, 3969–3983.e5. <https://doi.org/10.1016/j.neuron.2024.09.002>.
73. Donchin, O., Gribova, A., Steinberg, O., Bergman, H., and Vaadia, E. (1998). Primary motor cortex is involved in bimanual coordination. *Nature* 395, 274–278.
74. Gallego, J.A., Perich, M.G., Naufel, S.N., Ethier, C., Solla, S.A., and Miller, L.E. (2018). Cortical population activity within a preserved neural manifold underlies multiple motor behaviors. *Nat. Commun.* 9, 4233. <https://doi.org/10.1038/s41467-018-06560-z>.
75. Heming, E.A., Cross, K.P., Takei, T., Cook, D.J., and Scott, S.H. (2019). Independent representations of ipsilateral and contralateral limbs in primary motor cortex. *eLife* 8, e48190.
76. Michaels, J.A., Kashefi, M., Zheng, J., Codol, O., Weiler, J., Kersten, R., Lau, J.C., Gribble, P.L., Diedrichsen, J., and Pruszynski, J.A. (2025).

- Sensory expectations shape neural population dynamics in motor circuits. *Nature* 648, 668–677. <https://doi.org/10.1038/s41586-025-09690-9>.
77. Miri, A., Warriner, C.L., Seely, J.S., Elsayed, G.F., Cunningham, J.P., Churchland, M.M., and Jessell, T.M. (2017). Behaviorally Selective Engagement of Short-Latency Effector Pathways by Motor Cortex. *Neuron* 95, 683–696.e11. <https://doi.org/10.1016/j.neuron.2017.06.042>.
78. Mizes, K.G.C., Lindsey, J., Escola, G.S., and Ölveczky, B.P. (2024). The role of motor cortex in motor sequence execution depends on demands for flexibility. *Nat. Neurosci.* 27, 2466–2475. <https://doi.org/10.1038/s41593-024-01792-3>.
79. Rodriguez, A.C., Perich, M.G., Miller, L.E., and Humphries, M.D. (2024). Motor Cortex Latent Dynamics Encode Spatial and Temporal Arm Movement Parameters Independently. *J. Neurosci.* 44. <https://doi.org/10.1523/JNEUROSCI.1777-23.2024>.
80. Perich, M.G., Gallego, J.A., and Miller, L.E. (2018). A Neural Population Mechanism For Rapid Learning. *Neuron* 100, 964–976.e7. <https://doi.org/10.1101/138743>.
81. Comite, A.D., Crevecoeur, F., and Lefèvre, P. (2022). Continuous Tracking of Task Parameters Tunes Reaching Control Online. *eNeuro* 9. <https://doi.org/10.1523/ENEURO.0055-22.2022>.
82. Crevecoeur, F., Thonnard, J.-L., and Lefèvre, P. (2020). A Very Fast Time Scale of Human Motor Adaptation: Within Movement Adjustments of Internal Representations during Reaching. *eNeuro* 7, 0149. <https://doi.org/10.1523/ENEURO.0149-19.2019>.
83. Kalidindi, H.T., and Crevecoeur, F. (2024). Task-dependent coarticulation of movement sequences. *eLife* 13, RP96854. <https://doi.org/10.7554/eLife.96854>.
84. Kobak, D., and Mehring, C. (2012). Adaptation Paths to Novel Motor Tasks Are Shaped by Prior Structure Learning. *J. Neurosci.* 32, 9898–9908. <https://doi.org/10.1523/JNEUROSCI.0958-12.2012>.
85. Sadtlir, P.T., Quick, K.M., Golub, M.D., Chase, S.M., Ryu, S.I., Tyler-Kabara, E.C., Yu, B.M., and Batista, A.P. (2014). Neural constraints on learning. *Nature* 512, 423–426. <https://doi.org/10.1038/nature13665>.
86. Ostry, D.J., and Gribble, P.L. (2016). Sensory Plasticity in Human Motor Learning. *Trends Neurosci.* 39, 114–123. <https://doi.org/10.1016/j.tins.2015.12.006>.
87. Singh, K., and Scott, S.H. (2003). A motor learning strategy reflects neural circuitry for limb control. *Nat. Neurosci.* 6, 399–403.
88. Pemberton, J., Chadderton, P., and Costa, R.P. (2024). Cerebellar-driven cortical dynamics can enable task acquisition, switching and consolidation. *Nat. Commun.* 15, 10913. <https://doi.org/10.1038/s41467-024-55315-6>.
89. Israely, S., Ninou, H., Rajchert, O., Elmaleh, L., Harel, R., Mawase, F., Kadmon, J., and Prut, Y. (2025). Cerebellar output shapes cortical preparatory activity during motor adaptation. *Nat. Commun.* 16, 2574. <https://doi.org/10.1038/s41467-025-57832-4>.
90. Omrani, M., Murnaghan, C.D., Pruszynski, J.A., and Scott, S.H. (2016). Distributed task-specific processing of somatosensory feedback for voluntary motor control. *eLife* 5, e13141. <https://doi.org/10.7554/elife.13141>.
91. O'Shea, D.J., Duncker, L., Goo, W., Sun, X., Vyas, S., Trautmann, E.M., Diester, I., Ramakrishnan, C., Deisseroth, K., Sahani, M., et al. (2022). Direct neural perturbations reveal a dynamical mechanism for robust computation. Preprint at bioRxiv. <https://doi.org/10.1101/2022.12.16.520768>.
92. Park, J., Polidoro, P., Fortunato, C., Arnold, J., Mensh, B., Gallego, J.A., and Dudman, J.T. (2025). Conjoint specification of action by neocortex and striatum. *Neuron* 113, 620–636.e6. <https://doi.org/10.1016/j.neuron.2024.12.024>.
93. Vahidi, P., Sani, O.G., and Shanechi, M.M. (2024). Modeling and dissociation of intrinsic and input-driven neural population dynamics underlying behavior. *Proc. Natl. Acad. Sci. USA* 121, e2212887121. <https://doi.org/10.1073/pnas.2212887121>.
94. Scott, S.H. (2016). A Functional Taxonomy of Bottom-Up Sensory Feedback Processing for Motor Actions. *Trends Neurosci.* 39, 512–526. <https://doi.org/10.1016/j.tins.2016.06.001>.
95. Becker, M.I., and Person, A.L. (2019). Cerebellar Control of Reach Kinematics for Endpoint Precision. *Neuron* 103, 335–348.e5. <https://doi.org/10.1016/j.neuron.2019.05.007>.
96. Chowdhury, R.H., Glaser, J.I., and Miller, L.E. (2020). Area 2 of primary somatosensory cortex encodes kinematics of the whole arm. *eLife* 9, e48198. <https://doi.org/10.7554/eLife.48198>.
97. Guo, H., Kuang, S., and Gail, A. (2025). Sensorimotor environment but not task rule reconfigures population dynamics in rhesus monkey posterior parietal cortex. *Nat. Commun.* 16, 1116. <https://doi.org/10.1038/s41467-025-56360-5>.
98. Hennequin, G., Vogels, T.P., and Gerstner, W. (2014). Optimal Control of Transient Dynamics in Balanced Networks Supports Generation of Complex Movements. *Neuron* 82, 1394–1406. <https://doi.org/10.1016/j.neuron.2014.04.045>.
99. Heald, J.B., Lengyel, M., and Wolpert, D.M. (2021). Contextual inference underlies the learning of sensorimotor repertoires. *Nature* 600, 489–493. <https://doi.org/10.1038/s41586-021-04129-3>.
100. Elsayed, G.F., and Cunningham, J.P. (2017). Structure in neural population recordings: an expected byproduct of simpler phenomena? *Nat. Neurosci.* 20, 1310–1318. <https://doi.org/10.1038/nn.4617>.

STAR★METHODS

KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Deposited data		
Datasets generated from the simulations	This study	Figshare: https://doi.org/10.6084/m9.figshare.29149901
Experimental models: Organisms/strains		
Rhesus macaque (Macaca mulatta)	Open source	The datasets from two monkeys (A and B) used in this study were originally reported in prior publications. The dataset from Monkey A was reported as “exampleData” by Churchland and colleagues, ⁸ while the dataset for Monkey B was reported as monkey number 4 by Suresh and colleagues. ⁴⁵ The data are publicly available from the following repositories, Dropbox: https://www.dropbox.com/scl/fo/duf5zbcibcsux467c6oc9/AIN-ZiFsy2Huyh8h7VMdL7g?rlkey=3o5axmq5hirel4cij7g64jc0r&e=1&st=obza8dvy&dl=0 (Monkey A) and https://datadryad.org/dataset/doi:10.5061/dryad.xsj3tx9cm (Monkey B). Preprocessing of Monkey B dataset was performed using open-source code provided by Kuzmina and colleagues ⁴⁶ available at Github: https://github.com/NevVerVer/neural-dynamics-gyration
Software and algorithms		
Simulation code for feedback-controlled networks	This study	Github: https://github.com/neurohari/simulateLQRNetworks with the following DOI Zenodo: https://doi.org/10.5281/zenodo.18163693
Python 3	Open source	N/A
Anaconda (for scientific libraries in python)	Open source	https://anaconda.org/
Gyration analysis	Open source	GitHub: https://github.com/NevVerVer/neural-dynamics-gyration
jPCA analysis	Open source	https://churchland.zuckermaninstitute.columbia.edu/content/code
Tensor Maximum Entropy method (TME)	Open source	GitHub: https://github.com/gamaleldin/TME
MATLAB R2024b	The Mathworks Inc.	N/A

METHOD DETAILS

We used the following convention throughout the paper: scalar quantities appear in italicized symbols, vectors in bold and italicized font, matrices in capital symbols, and sets of parameters in bold and capital symbols. The dependency on time is sometimes omitted for clarity.

Network dynamics

We generated random networks comprising $n = 100$ units. The network activity was represented by the rate vector $\mathbf{r}(t)$ denoting the firing rate of all network units. The dynamics of this network was governed by the rate equation:

$$\tau_{net} \dot{\mathbf{r}} = -\mathbf{r}(t) + W\mathbf{r}(t) + \boldsymbol{\varepsilon} + \mathbf{u}(t) \quad (\text{Equation 7})$$

Here $\tau_{net} = 20ms$ is the time constant of a single network unit, and W is the network adjacency matrix. For compactness, we describe the network dynamics by combining the first two terms into $J = \frac{1}{\tau_{net}}(W - I)$, where I is the identity matrix. Throughout this article, we refer to J as the network state transition matrix. The input to the network consists of two terms: $\boldsymbol{\varepsilon} \in \mathbb{R}^n$ is a constant vector drawn from a zero-mean normal distribution $\mathcal{N}(0, 5)$ generating an offset representing spontaneous activity. This constant offset was treated with system augmentation considering $\dot{\boldsymbol{\varepsilon}} = 0$. The vector $\mathbf{u}(t)$ is the control signal that must be calculated based on the cost function representing task demands. In our model, we assumed that all units receive direct control inputs. We observed that restricting direct control inputs to a subset of network units produced similar results. The network adjacency matrix W was initialized such that all eigenvalues of the resulting state transition matrix J were negative. We studied three classes of random networks described below.

The first class of networks consisted of *Inhibition Stabilized Networks (ISN's)*, comprising separate inhibitory and excitatory populations, with $n_E = 80$ excitatory units and $n_I = 20$ inhibitory units. The combined network activity was represented by the rate vector $\mathbf{r}(t) = \begin{bmatrix} \mathbf{r}_E(t) \\ \mathbf{r}_I(t) \end{bmatrix}$, where $\mathbf{r}_E(t)$ and $\mathbf{r}_I(t)$ denote the firing rates of the excitatory and inhibitory populations, respectively. To simulate inhibitory stabilized regime, the network adjacency matrix was constructed using the algorithm described by Hennequin and colleagues in.^{26,98}

The second class of networks consisted of *Fully Connected Recurrent Networks*. Here, the network adjacency matrix was drawn from a normal distribution $\mathcal{N}(0, g^2/n)$, where $g = 0.8$. This configuration ensured stable recurrent dynamics, as described previously.^{9,11}

The third class of networks corresponded to *Erdős-Rényi* random networks, in which connections between unit pairs were determined probabilistically. Specifically, each connection was assigned with a given probability parameter, with $p = 0.4$ indicating that only 40% of total number of neuron pairs were connected (other parameter values yielded comparable results). Excitatory and inhibitory weights were assigned by randomly setting nonzero connections to +0.5 or −0.5 with equal probability. To ensure network stability, we first set the diagonal terms in the adjacency matrix to zeros. This was followed by incremental decrease in the diagonal terms until all eigen values became negative.

Overall, the given network drives limb movement by actuating muscles through an output vector $\mathbf{a}(t)$. The relationship between network activity and limb muscle actuation was defined as:

$$\mathbf{a}(t) = W_{out}\mathbf{r}(t) \quad (\text{Equation 8})$$

Here $W_{out} \in \mathbb{R}^{2 \times n}$ is the readout matrix whose entries are drawn from a normal distribution, $(0, 0.5)$, allowing multiple network units to contribute at different amounts to limb actuation. Notably, each random network corresponded to different random samples of the adjacency matrix, while maintaining a fixed readout matrix.

Limb dynamics

For the point mass approximation of limb dynamics, the limb state vector was defined as $\mathbf{q} = [x \ \dot{x} \ f_x \ y \ \dot{y} \ f_y]^T$ where the coordinates x and y represent the displacement of point mass in the lateral and forward directions respectively, the dot represent their time derivatives, and f_x and f_y are the forces applied by the controlled actuator to the mass. These forces respond to the network's output $\mathbf{a}(t)$, with a_x and a_y representing the network's actuation in the x and y directions, respectively. Overall, the system of differential equations for the limb and the actuation dynamics was:

$$m\ddot{x} = -k\dot{x} + f_x \quad (\text{Equation 9})$$

$$m\ddot{y} = -k\dot{y} + f_y \quad (\text{Equation 10})$$

$$\tau_{act}\dot{f}_x = a_x - f_x \quad (\text{Equation 11})$$

$$\tau_{act}\dot{f}_y = a_y - f_y \quad (\text{Equation 12})$$

Here k is the dissipative constant (0.1 Nsm^{-1}), and Equations 11, 12 represent the first order muscle dynamics with the actuation time-constant $\tau_{act} = 0.1\text{s}$.

The limb state evolves according to the differential equation $\dot{\mathbf{q}} = M\mathbf{q}(t) + F\mathbf{a}(t)$, with

$$M = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -\frac{k}{m} & \frac{1}{m} & 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{\tau_{act}} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & -\frac{k}{m} & \frac{1}{m} \\ 0 & 0 & 0 & 0 & 0 & -\frac{1}{\tau_{act}} \end{bmatrix} \quad (\text{Equation 13})$$

$$F = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{1}{\tau_{act}} & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & \frac{1}{\tau_{act}} \end{bmatrix} \quad (\text{Equation 14})$$

Additionally, we have simulated a linear approximation of a two-dimensional planar arm actuated at the shoulder and elbow joints. Details on the methods and simulations with this shoulder-elbow arm are provided in the Supplementary Material.

For simplicity, we can define a joint variable $\mathbf{z} = \begin{bmatrix} \mathbf{r} \\ \mathbf{q} \\ \mathbf{e} \end{bmatrix}$ to jointly describe the limb and network states, including the constant offset term for spontaneous activity, expressed as $\dot{\mathbf{e}} = 0$. The dynamics of this joint system can be expressed compactly as $\dot{\mathbf{z}} = \mathbf{A}\mathbf{z} + \mathbf{B}\mathbf{u}$, where:

$$\dot{\mathbf{z}} = \begin{bmatrix} \dot{\mathbf{r}} \\ \dot{\mathbf{q}} \\ \dot{\mathbf{e}} \end{bmatrix}, \mathbf{A} = \begin{bmatrix} \mathbf{J} & 0 & 1 \\ \mathbf{C} & \mathbf{M} & 0 \\ 0 & 0 & 0 \end{bmatrix} \text{ and } \mathbf{B} = \begin{bmatrix} \mathbf{I} \\ 0 \\ 0 \end{bmatrix} \mathbf{u}. \quad (\text{Equation 15})$$

Here 0 and 1 represents a block matrix filled with 0's and 1's respectively, and $\mathbf{I}$ represents an identity matrix with size equal to the number of network units. Here, $\mathbf{C} = \mathbf{F}\mathbf{W}_{out}$ represents the transformation ($\mathbf{F}\mathbf{W}_{out}$) from network activity ($\mathbf{r}$) to the forces acting on the limb ($\mathbf{f}$). Note the matrix $\mathbf{A}$ here indicates the default state-transition matrix of the network-body system. Applying closed-loop control, as follows, involves modifications in this state-transition matrix with feedback components.

Receding horizon control in a Linear Quadratic Regulator (LQR) form

We solved the problem of finding optimal network control inputs by repeatedly minimizing a cost on limb states and inputs within a moving time horizon

$$\mathbf{u} := \underset{\mathbf{u}}{\operatorname{argmin}} \mathcal{J}(\mathbf{z}, \mathbf{u}) \quad (\text{Equation 16})$$

Given the quadratic form of the cost function (Equation 2), and the linear system dynamics (Equation 1), the optimal control reflects the well-known LQR problem,^{35,41} whose solutions takes the following state-feedback form

$$\mathbf{u}(t) = -\mathbf{L}(\mathbf{A}, \mathbf{B}, \mathbf{Q}, \mathbf{R}, t)\mathbf{z}(t) \quad (\text{Equation 17})$$

Here, $\mathbf{L}$ is a time-varying control gain matrix that linearly maps the available state measurements ($\mathbf{z}$) or their estimates onto control inputs.

To compute the control gains, we transformed the control problem into a discrete time system based on Euler integration with 10ms time steps. Then the control gains were computed recursively backward in time following standard techniques.^{35,41} This procedure yields control gains that depend on the system dynamics ($\mathbf{A}$ and $\mathbf{B}$) and on the task objectives ($\mathbf{Q}$ and $\mathbf{R}$), as represented in Equation 17. Calling $\Theta^{(i)} = \{\mathbf{A}^{(i)}, \mathbf{B}^{(i)}, \mathbf{Q}^{(i)}, \mathbf{R}^{(i)}\}$ the set of task parameters known to the controller for a given task (i), we can rewrite the control law as:

$$\mathbf{u}(t) = -\mathbf{L}(\Theta^{(i)}, t)\mathbf{z}(t) = \begin{bmatrix} -\mathbf{L}_{net}(\Theta^{(i)}, t) & -\mathbf{L}_{limb}(\Theta^{(i)}, t) \end{bmatrix} \begin{bmatrix} \mathbf{r}(t) \\ \mathbf{q}(t) \end{bmatrix} \quad (\text{Equation 18})$$

This formulation expresses that the state-feedback control policy is specific to a given task parameter set Θ . Notice that in results section, we denoted the task parameter set with individual block components ($\mathbf{J}, \mathbf{C}, \mathbf{M}$) of the system matrix ($\mathbf{A}$), but they are essentially the same. Notably, the control law can be decomposed to separate the contribution of network or limb related feedback.

Overall, the Receding Horizon Control scheme⁴² iteratively recalculates the optimal control gains at each step. This process involves.

1. Collecting the task parameters $\Theta^{(i)}$ corresponding to the current task (i) at each time step (t)
2. Calculating the control gains over the prediction horizon $\{\mathbf{L}(\Theta^{(i)}, t), \dots, \mathbf{L}(\Theta^{(i)}, t+\Delta h)\}$, for the current time window (from t to $t+\Delta h$).

3. Applying the first control gain of the sequence, $\mathbf{u}(t) = L(\Theta^{(i)}, t)\mathbf{z}(t)$
4. Updating the time to $(t+1)$ and repeating from step 1.

In this work, we assumed that the knowledge of task parameters is available and expect that adding a module to perform inference about these parameters when they are not known exactly may link our developments to more general contexts.⁹⁹

Prepare-to-reach task

This task involved holding posture at a given starting location $(x_{start}^*, y_{start}^*)$ for an unspecified duration, then executing the movement toward the target $(x_{target}^*, y_{target}^*)$ upon receiving a GO signal. This paradigm effectively induces a transition between postural control (prepare) and reaching (move) control following a signal which occurs at a non-predictable time. To capture the fact that the occurrence of the GO signal was unpredictable, we employed a two-phases receding horizon cost-function with three terms as follows:

$$\mathcal{J}(\mathbf{q}, \mathbf{u}, t) = \underbrace{\int_t^{t+\Delta h_{prep}} (\mathbf{q}(s) - \mathbf{q}_{start}^*)^T Q_{prep}(s) (\mathbf{q}(s) - \mathbf{q}_{start}^*) ds}_{prep\ cost} + \underbrace{\int_{t+\Delta h_{prep}}^{t+\Delta h_{prep}+\Delta h_{move}} (\mathbf{q}(s) - \mathbf{q}_{target}^*)^T Q_{move}(s) (\mathbf{q}(s) - \mathbf{q}_{target}^*) ds}_{move\ cost} + \underbrace{\int_t^{t+\Delta h_{prep}+\Delta h_{move}} \mathbf{u}(s)^T R(s) \mathbf{u}(s) ds}_{input\ cost} \quad (19)$$

(Equation 19)

Here, $\mathbf{q}_{start}^* = [x_{start}^* \ 0 \ 0 \ y_{start}^* \ 0 \ 0]^T$ and $\mathbf{q}_{target}^* = [x_{target}^* \ 0 \ 0 \ y_{target}^* \ 0 \ 0]^T$ represent the overall desired start and terminal target hand states. Note that the velocity and force terms of the desired states are set to 0, meaning any deviations from the non-zero velocities and forces incurs a cost. Here, $Q_{prep}(\cdot)$ and $Q_{move}(\cdot)$ penalize state deviations during the preparatory and movement epochs, respectively. Before the GO signal, the first term penalizes deviations from the starting position over a predicted preparatory horizon $\Delta h_{prep} = 50ms$. This horizon maybe much shorter than the actual preparatory period (typically $>500ms$), to encourage rapid preparation for the uncertain GO signal. The second term penalizes deviations from the target over a fixed movement horizon $\Delta h_{move} = 500ms$, starting at $t+\Delta h_{prep}$. The third term penalizes control inputs over both horizons. This cost was recomputed at each time step during the preparatory period until the GO signal occurs.

Once the GO signal was delivered, the predicted preparatory horizon was replaced by the measured time of the GO signal (t_{GO}). The cost function for the movement period became:

$$\mathcal{J}(\mathbf{q}, \mathbf{u}, t) = \int_t^{t_{GO}+\Delta h_{move}} (\mathbf{q}(s) - \mathbf{q}_{target}^*)^T Q_{move}(s) (\mathbf{q}(s) - \mathbf{q}_{target}^*) ds + \int_t^{t_{GO}+\Delta h_{move}} \mathbf{u}(s)^T R(s) \mathbf{u}(s) ds \quad (Equation\ 20)$$

The first term penalizes the deviations from the target state, while the second term penalizes control inputs over the fixed movement horizon. By expressing the distinct objectives of preparatory and movement periods into separate cost terms, this optimization naturally yields two distinct sets of feedback control gains: one for maintaining posture while anticipating a movement during preparation, and another for executing the reach once the GO signal arrives.

We use the following parameter values, $Q_{prep} = \text{diag}([100, 100 \ 0, 0.100, 100, 0])$ for the cost on hand deviation from the desired start location, and $Q_{move} = \text{diag}([10, 0.1, 0, 10, 0.1, 0])$ for the cost on hand deviation from the desired terminal target location. The terminal target cost penalty Q_{move} was further scaled by the factor $\left(\frac{t}{\Delta h_{move}}\right)^6$ to ensure low initial cost when the elapsed time (t) is small relative to the movement horizon (Δh_{move}), and a smooth build up until the terminal time step. The penalty on the network control inputs was set at $R = 10^{-10}$ ensuring the solutions prioritize reaching the targets with relatively smaller cost on the squared magnitude of control inputs.

Changes in limb mechanics

We investigated two specific scenarios: a velocity-dependent force field and an inertial loading task. In the Velocity-Dependent Force Field Task (FF), the limb was subjected to a velocity-dependent force field where the force f_x was proportional to the effector's velocity $\dot{y}$, expressed as $f_x = \alpha \dot{y}$, and the force f_y was proportional to the effector's velocity $\dot{x}$, expressed as $f_y = \alpha \dot{x}$ as in standard motor

adaptation paradigms. This introduced a coupling between forward velocity-lateral force, and lateral velocity-forward force, effectively modifying the limb dynamics matrix M to $M(\alpha)$. The limb dynamics including the force fields can be represented as follows:

$$M = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -\frac{k}{m} & \frac{1}{m} & 0 & \frac{\alpha}{m} & 0 \\ 0 & 0 & -\frac{1}{\tau_{act}} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & \frac{\alpha}{m} & 0 & 0 & -\frac{k}{m} & \frac{1}{m} \\ 0 & 0 & 0 & 0 & 0 & -\frac{1}{\tau_{act}} \end{bmatrix} \quad (\text{Equation 21})$$

In the Inertial Loading Task, we simulated the addition of inertial loads (m_{load}), which altered the limb dynamics. These additional loads influence the limb dynamics matrix M , resulting in a modified matrix $M(m_{load})$. This allowed us to examine how the control scheme adapted to changes in limb inertia. The limb dynamics with the additional inertial loads was formulated as follows:

$$M = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -\frac{k}{m+m_{load}} & \frac{1}{m+m_{load}} & 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{\tau_{act}} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & -\frac{k}{m+m_{load}} & \frac{1}{m+m_{load}} \\ 0 & 0 & 0 & 0 & 0 & -\frac{1}{\tau_{act}} \end{bmatrix} \quad (\text{Equation 22})$$

We simulated the prepare-to-reach task using the receding horizon control scheme across five different levels for each parameter: $\alpha \in \{-5, -2.5, 0, 2.5, 5\} \text{ Nsm}^{-1}$ and $m_{load} \in \{0.2, 0.4, 0.6, 0.8, 1.0\} \text{ KG}$. Notably, the cost parameters were the same across the three simulated reaching tasks: center-out reaching, force field and inertial loading.

Changes in task objectives

To investigate the influence of task objectives on the network-body dynamics, we modulated matrix Q that assigns cost penalty on each state component.

In the spatial target redundancy task, we focused on the lateral deviation to simulate reaching to a target with different geometry (as the dot-bar task⁵¹). We systematically varied the parameter governing lateral deviation in the cost matrix Q_{move} from Equations 19 and 20 by multiplying the default value with a scalar λ_{bar} across five levels: $\{0.001, 0.01, 0.1, 1, 10\}$. This scalar multiplication essentially transforms the matrix penalizing state deviation as $Q_{move} = \text{diag}([100 \times \lambda_{bar}, 0.1, 0.100, 0.1, 0])$. A low parameter value (e.g., when multiplied with 0.001) imposed minimal penalty on lateral (x) deviation, allowing for larger deviation in endpoint location. Conversely, higher parameter value (e.g., when multiplied with 10) imposed a strong penalty on the lateral deviation, necessitating corrective actions to achieve a low terminal displacement.

In the “shooting” task, we focused on the hand velocity. We varied the cost associated with the lateral and forward velocity parameters by multiplying the default value with a scalar λ_{shoot} across the same five levels: $\{0.001, 0.01, 0.1, 1, 10\}$. This scalar multiplication transforms the terms penalizing the hand velocity by transforming the matrix $Q_{move} = \text{diag}([100, 0.1 \times \lambda_{shoot}, 0.100, 0.1 \times \lambda_{shoot}, 0])$. At low parameter value (e.g., when multiplied with 0.001), stopping at the target was not required, resulting in non-zero terminal velocity – consistent with a “shooting” motion. At higher values (e.g., when multiplied with 10), terminal velocity was heavily penalized, necessitating deceleration or stopping of the hand as it reached the target.

In both bar and shooting tasks, the penalty on the network control inputs remained unchanged at $R = 10^{-10}$.

Both biomechanical adaptation (modifying matrix M) and flexibility in task objectives (adjusting matrix Q) involved changes in the task parameter set $\Theta^{(t)}$. As described in Equations 17 and 18, alterations in these parameters modified the control gains $L(t)$, leading to

task-dependent network-body trajectories. By systematically varying the parameters associated with limb dynamics (M) and the entries in the cost function (Q), we assessed how the control system adapted its policy to maintain optimal performance across different biomechanical conditions and task objectives.

QUANTIFICATION AND STATISTICAL ANALYSIS

PCA in preparatory and movement epochs

Following previous work by Elsayed and colleagues,³⁹ we grouped the network responses into two matrices corresponding to the preparatory and movement epochs. Preparatory epoch responses were represented by a rectangular matrix $P \in \mathbb{R}^{n \times cd}$, where n is the number of units, c is the number of conditions (e.g., number of targets, force-field values, bar/shoot target penalties), and d is the duration within the preparatory epoch. Similarly, movement execution period responses are represented by $E \in \mathbb{R}^{n \times cd}$. We first soft-normalized the activity by the activity range plus a small constant (0.005). We performed PCA on the covariance matrix of preparatory response epochs P , treating each row (i.e., network unit) as a variable, to obtain the top six components (prep-PCs) of the network trajectories in the n -dimensional space. Similarly, we obtained the top six components from the movement epochs (move-PCs) by performing PCA on the covariance matrix of movement response E .

We compared how the pairwise correlations changed between the preparatory and movement epochs by computing the correlation in firing rates between each possible pair of units during each epoch. We then computed an average of these correlations across eight target conditions. This yielded two correlations for each pair of units, one for each of the preparation and movement epoch (Figure 2C). We tested if the correlation coefficient for each pair of neurons in one epoch linearly correlated with that of the other epoch. Pearson correlation coefficient was used to quantify the pairwise correlations between the preparatory and movement epochs across all neuron pairs (Figure 2D).

Alignment index

We determined how aligned the top-six preparatory and movement related principal components were by the alignment index, following the procedure described previously in³⁹:

$$AI = \frac{\text{Tr}(X_{move}^T \text{Cov}_{prep} X_{move})}{\text{Tr}(X_{prep}^T \text{Cov}_{prep} X_{prep})} \quad (\text{Equation 23})$$

Here X_{prep} and $X_{move} \in \mathbb{R}^{6 \times cd}$ are matrices containing the top six preparatory and movement principal components, Cov_{prep} is the preparatory covariance matrix, and $\text{Tr}(\cdot)$ is the trace operator. Equation 23 described the ratio of how much variance the top six movement principal components could explain of the preparatory activity with how much top six preparatory principal components could explain of the preparatory activity (the maximum preparatory variance captured by any six linear projections). The alignment index ranges from 0, indicating the preparatory and movement activities are orthogonal, to 1, indicating the preparatory and movement activities are perfectly aligned.

In Figure 2, we compared the alignment indices of the model with two publicly available monkey M1 recordings from prepare to reach tasks.^{8,45} The public dataset from⁸ predominantly captures execution-related neural activity, with only a brief 50ms window preceding the onset of the phasic M1 response, that may reflect the states achieved at the end of preparatory epoch. Accordingly, we defined proxy of the preparatory response (X_{prep}) from the 0-50ms interval prior to the phasic response onset, and the execution response (X_{move}) from the 50-100ms interval following movement onset, ensuring symmetric 50ms windows for both responses. For the model and M1 recordings from,⁴⁵ which include more extensive preparatory epochs, we defined the preparatory response in the 200ms window spanning 50–250 ms before the GO cue. The execution response was defined in the 200ms window from 50 to 250ms following movement onset.

jPCA analysis

We performed jPCA analysis on the network activity using procedure developed in.⁸ First, we obtained the top six movement period principal components ($6 \times cd$ dimensions). We numerically calculated the derivative of X yielding $\dot{X}$, and fit a linear dynamical model which found a relationship between X and $\dot{X}$, expressed as $\dot{X} = MX$. Here, M was constrained to be skew-symmetric, which restricted the possible dynamical systems to systems with oscillatory or rotational dynamics. We assessed the model's fit by calculating the coefficient of determination or goodness-of-fit metric (R^2). jPCA analysis was also applied on publicly available M1 recordings from^{8,45} (see [key resources table](#)). Across datasets, jPCA analysis was applied for the first 300ms after movement onset.

Quantifying rotations and decay in data

To quantify the rotational structure within the neural population activity, we implemented the gyration number analysis proposed by Kuzmina and colleagues.⁴⁶ This model-agnostic method measures the rotational and decay components present in time-series data

without requiring dimensionality reduction. The analysis was performed using the publicly available code from the original publication. The core steps were as follows:

First, neural activity data of movement execution epochs from all experimental conditions were concatenated into a single matrix $E \in \mathbb{R}^{n \times cd}$, where rows represent the number of nodes and the columns represent the activity across $c = 8$ reaching directions and d time duration (from 0 to 300 ms after the GO cue). To make the data formatting suitable with the libraries developed by Kuzmina and colleagues, we transposed the original matrix to $Y = E^T$ so that the rows of Y represent the time points across all reach directions (cd) and columns represent individual nodes (n).

Next, we computed spatial differential covariance matrix as $\dot{Y}Y^T$, and identified its i eigen values (λ_i). The coordinates for the gyration plane were derived from the eigen spectrum. The decay axis, which captures expansive or contractive dynamics, and the rotation axis, which captures rotational strength, were calculated using the first complex-conjugated eigenvalue pair as follows:

Rotation axis:

$$rotation = \frac{|Im(\lambda_1)| + |Im(\lambda_2)|}{\sum_{i=1}^n |\lambda_i|} \quad (\text{Equation 24})$$

Decay axis:

$$decay = \frac{|Re(\lambda_1)| + |Re(\lambda_2)|}{\sum_{i=1}^n |\lambda_i|} \quad (\text{Equation 25})$$

This provided a standardized, two-dimensional measure to compare the rotational dynamics across datasets.

Tensor maximum entropy (TME)

We tested our findings – specifically, low alignment indices, quantity of rotations, and preserved PCs across networks – against the hypothesis that these features are a byproduct of tuning and smoothness properties of neurons. We employed the TME method to generate surrogate datasets.¹⁰⁰ TME method generates surrogate datasets that retain up to the second-order marginal moments of the original network data (mean, covariances across neurons, conditions, and time) while randomizing the higher-order interaction terms. We generated surrogate datasets for each network across simulated tasks.

In the center-out reach simulated task, we computed the PCs of 1000 randomly sampled TME surrogate datasets, for each network activity dataset, during both the preparation and movement epochs, then calculated the alignment index and the fit to rotational dynamics. We compared these surrogate alignment indices and rotations with those obtained from the actual network data (Figures 2; S2; S3 and S4). Similarly, surrogate datasets were computed for M1 recordings (Figure 2), and the alignment indices and rotations were compared with those obtained from the actual data.

Furthermore, for each task, we computed the PCs of 1000 randomly sampled TME datasets for each network and task, and examined the correlations among these PCs across networks performing the same task (Figure S8).

Singular value decomposition analysis

We performed singular value decomposition to analyze the dimensionality of the network connectivity matrices and population activities (Figure 6). Recall that the network-body dynamics can be jointly expressed as a closed-loop system of the form presented in Equation 4. For simplicity, we can ignore the spontaneous activity offset ε , and the overall system can be represented as follows,

$$\begin{bmatrix} \dot{\mathbf{r}} \\ \dot{\mathbf{q}} \end{bmatrix} = \begin{bmatrix} \mathbf{J} - \mathbf{L}_{net}(\Theta) & -\mathbf{L}_{limb}(\Theta) \\ \mathbf{C} & \mathbf{M} \end{bmatrix} \begin{bmatrix} \mathbf{r} \\ \mathbf{q} \end{bmatrix} \quad (\text{Equation 26})$$

The above network-body dynamics jointly express closed-loop system of the form $\dot{\mathbf{z}} = \mathbf{A}_c \mathbf{z}$, where $\mathbf{A}_c$ has a block structure corresponding to network and limb dynamics as well as their connection. Note that $\mathbf{A}_c$ represents the closed-loop state-transition matrix that includes control gains, and it is different from the open-loop state-transition matrix $\mathbf{A}$ described in Equation 15 before applying the feedback control law. This matrix was then decomposed into two terms ($\mathbf{A}_c^{(1)}$ and $\mathbf{A}_c^{(2)}$) as described in Equation 5.

Singular value spectra have been found for each of the decomposed terms and of the composite matrix (Figures 6A and 6B). Additionally, singular value spectrum has also been computed for the network time-series activity matrix (i.e., network dynamics) in the movement epoch, $E \in \mathbb{R}^{n \times cd}$ (Figure 6C). Here, $n = 100$ are the number of network nodes; $c = 8$ describes the number of target directions in the center-out reaching task for both the via-point (Figure 6D) and biomechanical tasks (Figure 6F); d represents the time-window of the network activity (from 50 to 400ms after the movement onset).

Dimensionality

Given the singular value spectra, $\sigma_{i=1,2,\dots,n}$, of a matrix A , its dimensionality can be measured from the well-defined Roy and Vetterli's effective rank as described in⁴⁰:

$$erank(A) = \exp([H(p_1, \dots, p_n)]) \quad (\text{Equation 27})$$

Here $H(p_1, \dots, p_n) = -\sum_i p_i \log p_i$ is the Shannon entropy of the singular value spectrum, measured as a function of the singular value mass function

$$p_i = \frac{\sigma_i}{\|\mathbf{A}\|_*} = \frac{\sigma_i}{\sum_{j=1}^n \sigma_j}, \forall i \in \{1, \dots, n\} \quad (\text{Equation 28})$$

erank measures the uniformity of the singular value distribution, resulting in a low effective rank when the singular values decrease rapidly.