Optimal anticipatory control of movement as a theory of motor preparation: a thalamo-cortical circuit model

Ta-Chu Kao[1], Mahdieh S. Sadabadi[1,2], and Guillaume Hennequin[@1]

[1]Computational and Biological Learning Lab, Department of Engineering, University of Cambridge, Cambridge, U.K.
[2]Department of Automatic Control and Systems Engineering, University of Sheffield, Sheffield, U.K.

@ Corresponding author (g.hennequin@eng.cam.ac.uk)

SUPPLEMENTARY MATERIAL

Contents

S1   Overview of this document

This document provides details for the theory presented at a high level in the main text, as well as details of the methods used for data analysis. In Section S2, we begin by describing our model of movement generation, including our model of M1 dynamics and that of the two-link arm. In Section S3, we present our formalization of optimal preparatory control and how it relates to the "optimal subspace hypothesis" of Shenoy et al. (2013). In Section S4, we then apply optimal feedback control theory to this problem, and present the steps taken to build a circuit implementation taking into account relevant neural constraints. In Section S5, we give an in-depth description of the methods used for analyzing the monkey data.

The parameters used in our simulations are all listed in Table S1.

S2   A model for movement generation by cortical dynamics

S2.1   Network dynamics

We model M1 as a network with two separate populations of $N_{\mathrm{E}} = 160$ excitatory (E) neurons and $N_{\mathrm{I}} = 40$ inhibitory (I) neurons, operating in the inhibition-stabilized regime (Tsodyks et al., 1997; Ozeki et al., 2009; Hennequin et al., 2014). We constructed its synaptic architecture exactly as we have shown previously in Hennequin et al. (2014). We describe the dynamics of these $N = N_{\mathrm{E}} + N_{\mathrm{I}}$ neurons by a standard nonlinear rate equation. Specifically, the vector $\mathbf{x}(t) = \left(\mathbf{x}_{\mathrm{E}}(t)^T, \mathbf{x}_{\mathrm{I}}(t)^T\right)^T$ of internal neuronal "activations" obeys:

$$\tau \frac{d\mathbf{x}}{dt} = -\mathbf{x}(t) + \mathbf{W}\phi\left[\mathbf{x}(t)\right] + \overline{\mathbf{h}} + \mathbf{h}(t) + \mathbf{u}(t) \tag{S1}$$

where τ is the single-neuron time constant, $\mathbf{W}$ is the synaptic connectivity matrix, and $\phi(x) = \max(x, 0)$ is a static, rectified-linear nonlinearity – applied elementwise to $\mathbf{x}$ – that converts internal activation into momentary firing rates. The input consist of three terms: an input $\overline{\mathbf{h}} = \mathbf{x}_{\mathrm{sp}} - \mathbf{W}\phi\left[\mathbf{x}_{\mathrm{sp}}\right]$ held constant throughout all phases of the task to instate a heterogeneous set of spontaneous firing rates $\mathbf{x}_{\mathrm{sp}}$ (elements drawn i.i.d. from $\mathcal{N}(20, 9)$); a transient, movement-condition-independent and spatially uniform α-shaped input bump

$$\mathbf{h}(t) = (1, \ldots, 1)^T \times \begin{cases} \text{if } t > t_{\mathrm{move}}: & A\left[\exp\left(-\frac{t - t_{\mathrm{move}}}{\tau_{\mathrm{decay}}}\right) - \exp\left(-\frac{t - t_{\mathrm{move}}}{\tau_{\mathrm{rise}}}\right)\right] \\ \text{otherwise:} & 0 \end{cases} \tag{S2}$$

kicking in at movement onset (Kaufman et al., 2016); and a preparatory control input $\mathbf{u}(t)$ (further specified below) whose role is to drive the circuit into a preparatory state appropriate for each movement.

We assume that the uncontrolled dynamics ($\mathbf{u} = \mathbf{0}$) of this network directly drives movement. A two-dimensional linear readout of the excitatory neurons,

$$\mathbf{m}(t) = \mathbf{C}\phi\left[\mathbf{x}_{\mathrm{E}}(t)\right] \tag{S3}$$

with $\mathbf{C} \in \mathbb{R}^{2 \times N_{\mathrm{E}}}$, is used as a set of torques to actuate the two-link arm model described in the next section. Although our simulations show that the muscle readouts $\mathbf{m}(t)$ are very small during preparation, they do cause drift in the hand prior to movement onset (and therefore wrong movements afterwards) as their are effectively integrated twice by the dynamics of the arm (see below). For this reason, we artificially set $\mathbf{m}$ to zero during movement preparation.

S2.2 Arm model

To simulate reaching movements, we used the planar two-link arm model previously described in Li and Todorov (2004), with parameters listed in Table S1. The upper arm and the lower arm are connected at the elbow (Figure S1). The two links have lengths L_1 and L_2, masses M_1 and M_2, and moments of inertia I_1 and I_2 respectively. The lower arm's center of mass is located a distance D_2 from the elbow. By considering the geometry of the upper and lower limb, we can write down the position of the hand as a vector $\mathbf{y}(t)$ given by

$$\mathbf{y} = \left(\begin{array}{c} L_1 \cos \theta_1 + L_2 \cos(\theta_1 + \theta_2) \\ L_1 \sin \theta_1 + L_2 \sin(\theta_1 + \theta_2) \end{array} \right) \tag{S4}$$

where the angles θ_1 and θ_2 are defined in Figure S1A. The joint angles $\boldsymbol{\theta} = (\theta_1; \theta_2)^T$ evolve dynamically according to the differential equation

$$\mathbf{m}(t) = \mathcal{M}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \mathcal{X}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) + \mathcal{B}\dot{\boldsymbol{\theta}}, \tag{S5}$$

where $\mathbf{m}(t)$ is the momentary torque vector (the output of the neural network, c.f. Equation S3), $\mathcal{M}$ is the matrix of inertia, $\mathcal{X}$ accounts for the centripetal and Coriolis forces, and $\mathcal{B}$ is a damping matrix representing joint friction. These parameters are given by

$$\mathcal{M}(\boldsymbol{\theta}) = \left(\begin{array}{cc} a_1 + 2a_2 \cos \theta_2 & a_3 + a_2 \cos \theta_2 \\ a_3 + a_2 \cos \theta_2 & a_3 \end{array} \right) \tag{S6}$$

$$\mathcal{X}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) = a_2 \sin \theta_2 \left(\begin{array}{c} -\dot{\theta}_2(2\dot{\theta}_1 + \dot{\theta}_2) \\ \dot{\theta}_1{}^2 \end{array} \right) \qquad \mathcal{B} = \left(\begin{array}{cc} 0.05 & 0.025 \\ 0.025 & 0.05 \end{array} \right) \tag{S7}$$

with $a_1 = I_1 + I_2 + M_2 L_1^2$, $a_2 = M_2 L_1 D_2$, and $a_3 = I_2$.

S2.3 Target hand trajectories

We generated a set of eight target hand trajectories, namely straight reaches of size $d = 20$ cm going from the origin into eight different directions, with a common bell-shaped scalar speed profile

$$v(t) = v_0 \left(\frac{t}{\tau_{\text{reach}}} \right)^2 \exp \left[-\frac{1}{2} \left(\frac{t}{\tau_{\text{reach}}} \right)^2 \right], \tag{S8}$$

where v_0 is chosen such that the hand reaches the target. Given these target hand trajectories, we solved for the required timecourse of the torque vector $\mathbf{m}(t)$ through optimization, by backpropagating through the equations of motion of the arm to minimize the squared difference between actual and desired hand trajectories. We forced the initial torques at $t = 0$ to be zero, and also included a roughness penalty in the form of average squared torque gradient.

Similarly, we then backpropagated through the equations of the recurrent neural network (Equations S1 and S3) to optimize a set of eight movement-specific initial conditions $\{\mathbf{x}_k^\star\}$, $k = 1, \ldots, 8$, as well as the readout matrix $\mathbf{C}$, so as to achieve the desired torques in the output. This was done by minimizing the squared difference between actual and desired torque trajectories, with a penalty on $\mathbf{C}$'s squared Frobenius norm.

We parameterized the readout matrix $\mathbf{C}$ in such a way that its nullspace automatically contains both the spontaneous activity vector $\mathbf{x}_{\text{sp}}$ and the movement-specific initial conditions $\{\mathbf{x}_k^\star\}$, $k = 1, \ldots, 8$. This is to ensure that (i) there is no muscle output during spontaneous activity and (ii) the network does not unduly generate muscle output at the end of preparation, before
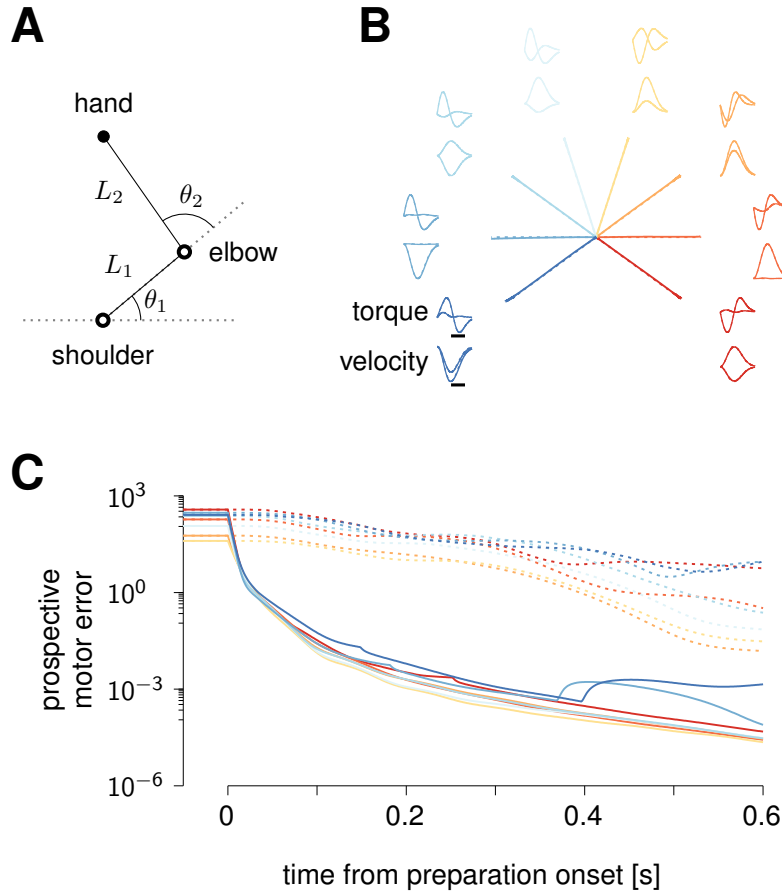
3

Figure S1: **(A)** Schematics of the arm model (see supplementary text). **(B)** Reaches produced by the model, along with associated torques at the two joints, and x-y velocities of the hand (solid lines). Scale bar: 200 ms. Dashed lines denote target trajectories. **(C)** Decay of the prospective motor cost during movement preparation, under optimal feedback control (solid lines, same color code for the movements as in B), and under a naive policy with temporally constant input that achieves a movement-specific fixed point ($\mathbf{x}_k^\star$) with zero motor error, asymptotically (dashed lines).

movement. More specifically, prior to movement preparation and long enough after movement execution, the cortical state is in spontaneous activity $\mathbf{x}_{\mathrm{sp}}$. By ensuring that $\mathbf{C}\mathbf{x}_{\mathrm{sp}} = \mathbf{0}$, we ensure that our model network does not elicit movement "spontaneously". Similarly, control inputs drive the cortical state $\mathbf{x}$ towards $\mathbf{x}_k^\star$, which it will eventually reach late in the preparation epoch. Therefore, if $\mathbf{x}_k^\star$ is not in the null-space of C, it would be difficult—if not impossible—for muscle readout to remain small or event silent during preparation.

S3   Formalization of anticipatory motor control

Here, we address the problem of controlling the state of the cortical network *in anticipation* of the movement phase, which is to occur in open-loop following the go cue or trigger. That is to say, we aim at driving the network into preparatory states from which the uncontrolled dynamics would generate the desired muscle output.

We formalise the notion of anticipatory control by asking: given an intended movement (indexed by k), and the current (preparatory) state $\mathbf{x}(t)$ of the network, how accurate would the movement

4

be if it were to begin *now*? We measure this prospective motor error as the squared difference $\mathcal{C}_k(\mathbf{x})$ between the subsequent timecourse of the target network output torques $\mathbf{m}_k^\star(t')$ $(t' > t)$, and that of the torques $\mathbf{m}(t')$ that the network would generate (Equations S1 and S3) *if left uncontrolled* from time t onwards, starting from initial condition $\mathbf{x}(t)$:

$$\mathcal{C}_k(\mathbf{x}) \triangleq \int_t^\infty \|\mathbf{m}(t') - \mathbf{m}_k^\star(t')\|^2 dt' \quad \text{with} \quad \mathbf{u}(t' \geq t) = 0 \tag{S9}$$

(we will often drop the explicit reference to the movement index k to remove clutter, as we did in the main text). Thus, any preparatory state $\mathbf{x}$ is associated with a prospective motor error $\mathcal{C}(\mathbf{x})$.

The prospective error $\mathcal{C}(\mathbf{x})$ changes dynamically during movement preparation, as $\mathbf{x}(t)$ evolves under the action of control inputs. The aim of the control inputs is to rapidly decrease this prospective error, until it drops below an acceptably small threshold, or until movement initiation is forced. We formalize this as the minimization of the following control cost:

$$\mathcal{J}[\mathbf{u}(t)] \triangleq \left\langle \int_0^\infty \left(\mathcal{C}(\mathbf{x}(t)) + \lambda \mathcal{R}(\mathbf{u}(t)) \right) dt \right\rangle_{p(\mathbf{x}(t=0))} \tag{S10}$$

where $\mathcal{R}(\mathbf{u})$ is a regularizer described below, and the average is over some distribution of states we expect the network to be found in at the time the controlled preparatory phase begins (we leave this unspecified for now as it turns out not to influence the optimal control strategy – see below). Thus, we want control inputs to rapidly steer the cortical network into states of low $\mathcal{C}(\mathbf{x})$ from which the movement can be readily executed. The infinite-horizon summation expresses uncertainty about how long movement preparation will last, and indeed encourages the network to be "ready" as soon as possible.

Mathematically, $\mathcal{J}[\cdot]$ is a functional of the spatio-temporal pattern of control input $\mathbf{u}(t)$ — indeed, $\mathbf{x}(t)$ depends on $\mathbf{u}(t)$ through Equation S1. The regularizer $\mathcal{R}(\mathbf{u})$, or "control effort", is specified further below. Without regularization, the problem is ill-posed, as arbitrarily large control inputs could be used to instantaneously force the network into the right preparatory state in theory, leading to physically infeasible control solutions in practice. Also note that Equation S10 is an "infinite-horizon" cost, i.e. the integral runs from the beginning of movement preparation when control inputs kick in, until infinity. This does *not* mean, however, that the preparation phase must be infinitely long. In fact, good control inputs should (and will!) bring the integrand close to zero very fast, such that the movement is ready to begin after only a short preparatory phase (see e.g. Figure 2A in the main text).

In order to derive the optimal control law, we further assume that the dynamics of the network remain approximately linear during both movement preparation and execution. This holds approximately true as long as only few neurons become silent in either phase (the saturation at zero firing rate is the only source of nonlinearity in our model, c.f. $\phi(\cdot)$ in Equation S1). In this case, the prospective motor error $\mathcal{C}(\mathbf{x})$ of Equation S9 affords a simpler, interpretable form, which we derive now. In the linear regime, Equation S1 becomes

$$\tau \frac{d\mathbf{x}}{dt} = \mathbf{A}\mathbf{x}(t) + \overline{\mathbf{h}} + \mathbf{h}(t) + \mathbf{u}(t) \tag{S11}$$

with an effective state transition matrix $\mathbf{A} \triangleq \mathbf{W} - \mathbf{I}$. The network output at time t, starting from state $\mathbf{x}$ at time $t = 0$ and with no control input thereafter, has an analytical form given by

$$\mathbf{m}(t) = \mathbf{C}\left[e^{(t/\tau)\mathbf{A}}(\mathbf{x} - \mathbf{x}_{\text{sp}}) + \mathbf{q}(t)\right] \tag{S12}$$

and similarly for $\mathbf{m}^\star(t)$ with $\mathbf{x}$ replaced by $\mathbf{x}^\star$. The final term $\mathbf{q}(t)$ is a contribution from the external input: it does not depend on the initial condition, and is therefore the same in both

cases. Thus, the prospective motor error (Equation S9) attached to preparatory state $\mathbf{x}$ is given by

$$\mathcal{C}(\mathbf{x}) = \int_0^\infty \|\mathbf{C}e^{(t/\tau)\mathbf{A}}(\mathbf{x} - \mathbf{x}^\star)\|^2 dt = (\mathbf{x} - \mathbf{x}^\star)^T \underbrace{\left[\int_0^\infty \left(e^{(t/\tau)\mathbf{A}^T}\mathbf{C}^T\mathbf{C}e^{(t/\tau)\mathbf{A}}\right) dt\right]}_{\mathbf{Q}} (\mathbf{x} - \mathbf{x}^\star).$$

(S13)

The matrix integral on the r.h.s. of Equation S13 is known as the "observability Gramian" $\mathbf{Q}$ of the pair $(\mathbf{A}, \mathbf{C})$ (Skogestad and Postlethwaite, 2007; Kao and Hennequin, 2019). It is found algebraically as the solution of the Lyapunov equation (Lemma 1 in Appendix A)

$$\mathbf{A}^T\mathbf{Q} + \mathbf{Q}\mathbf{A} + \tau\mathbf{C}^T\mathbf{C} = 0$$

(S14)

Thus, under linearity assumptions, the prospective motor error is a quadratic function of the difference between the momentary preparatory state and the optimal initial state $\mathbf{x}^\star$ known to elicit the right muscle outputs in open loop. The Gramian $\mathbf{Q}$, a symmetric, positive-definite matrix, determines how preparatory deviations away from $\mathbf{x}^\star$ give rise to subsequent motor errors. Deviations along the few eigenmodes of $\mathbf{Q}$ associated with large eigenvalues will lead to large errors in muscle outputs. The optimal control input $\mathbf{u}(t)$ will need to work hard to minimize this type of deviations – luckily, there are only few of them (Figure S3, top left; see also Figure 4 in the main text). In contrast, errors occurring along eigenmodes of $\mathbf{Q}$ with small eigenvalues – the vast majority – have almost no motor consequences. This large bottom subspace of $\mathbf{Q}$ provides a safe buffer in which preparatory activity is allowed to fluctuate without sacrificing control quality. It comprises both the "readout-null" and "dynamic-null" directions described in the main text (Figure 1D). Geometrically, we can therefore think of the optimal preparatory subspace as a high-dimensional ellipsoid centered on $\mathbf{x}^\star$, and whose small and (potentially inifinitely) large axes are given by the top and bottom eigenvectors of $\mathbf{Q}$, respectively (small axes, steep directions, large eigenvalues; long axes, flat directions, small eigenvalues).

To quantify these geometric insights, we define a measure of motor potency for a subspace $\mathbf{S}$ spanned by orthonormal column vectors $(\mathbf{d}_1, \mathbf{d}_2, \cdots, \mathbf{d}_K)$ as

$$\text{motor potency}(\mathbf{S}) = \frac{1}{K}\sum_{i=1}^K \mathbf{d}_i^T\mathbf{Q}\mathbf{d}_i.$$

(S15)

This quantifies the amount of prospective motor error induced on average when the state of the network deviates from $\mathbf{x}^\star$ in $\mathbf{S}$. This is what we showed in Figure 4.

Finally, we note that the optimal control input $\mathbf{u}(t)$ must keep the infinite-horizon integral in Equation S10 finite. This requires $\mathbf{x}(t)$ to reach a fixed point equal to $\mathbf{x}^\star$, which in turn requires the control input to eventually settle to a steady-state value equal to

$$\mathbf{u}^\star = -\mathbf{A}\mathbf{x}^\star - \mathbf{h}$$

(S16)

Thus, defining $\delta\mathbf{u}(t) \triangleq \mathbf{u}(t) - \mathbf{u}^\star$ and $\delta\mathbf{x}(t) \triangleq \mathbf{x}(t) - \mathbf{x}^\star$, a relevant regularizer for our control problem is

$$\mathcal{R}(\mathbf{u}(t)) \triangleq \|\delta\mathbf{u}(t)\|^2$$

(S17)

and our control cost functional becomes

$$\mathcal{J}[\mathbf{u}(t)] = \left\langle \int_0^\infty \left[\delta\mathbf{x}(t)^T\mathbf{Q}\delta\mathbf{x}(t) + \lambda\|\delta\mathbf{u}(t)\|^2\right] dt \right\rangle_{p(\mathbf{x}(t=0))}.$$

(S18)

In our simulations, we perform a simple scalar normalization of $\mathbf{Q}$ so that trace$(\mathbf{Q}) = N$. This makes the first term of the cost more easily comparable to the energy penalty $\lambda\|\delta\mathbf{u}\|^2$, which also scales with N. In the next section, we show that the quadratic formulation of $\mathcal{C}(\mathbf{x})$ in Equation S17 leads to analytically tractable optimization of our cost functional $\mathcal{J}$ in Equation S10. We will continue to assume linear network dynamics in order to derive optimal control laws, but we will always implement these solutions in the fully nonlinear circuit.
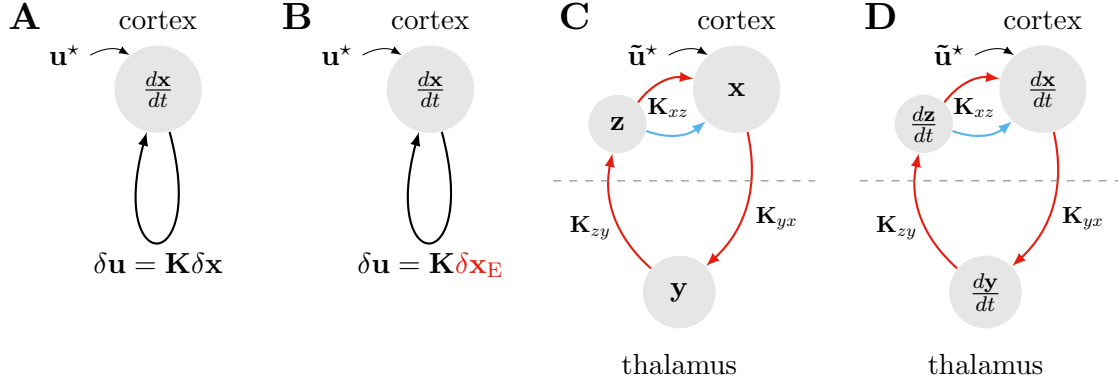
6

Figure S2: Four steps to arrive at a biologically plausible implementation of optimal anticipatory motor control. **(A)** The classical LQR solution prescribes instantaneous state feedback, with reentrant control inputs of the form of $\delta\mathbf{u}(t) = \mathbf{K}\delta\mathbf{x}(t)$ and a constant external input $\mathbf{u}^\star$. Section S4.2 shows how to obtain the optimal feedback matrix $\mathbf{K}$ in this case. **(B)** It is possible to constrain feedback to be of the form $\delta\mathbf{u}(t) = \mathbf{K}\left[\mathbf{I}_{N_\mathrm{E}} \ \mathbf{0}_{N_i}\right]\delta\mathbf{x}(t) = \mathbf{K}\delta\mathbf{x}_\mathrm{E}(t)$ instead. Section S4.5 shows how to obtain the optimal feedback matrix $\mathbf{K}$ in this case. **(C)** For flexibility, we propose that feedback be relayed by the motor thalamus, which is under the gating control of the basal ganglia. In Section S4.6, we show that the optimal feedback gain $\mathbf{K}$ obtained in (B) can be decomposed into sign-constrained matrices implementing E connections from M1 to thalamus ($\mathbf{K}_{yx}$), from thalamus to M1-layer 4 ($\mathbf{K}_{zy}$), and Dale-structured E/I connections from layer 4 back into the main recurrent M1 circuit ($\mathbf{K}_{xz}$). **(D)** Finally, first-order dynamics can be introduced in our model thalamus and M1-layer 4 neurons. We show in Section S4.7 how the lag introduced by such dynamics can be taken into account, to obtain a set of connections that achieve optimal anticipatory control of movement under these biological constraints.

S4   Optimal control solutions and associated circuits

S4.1   Naive solution

A straightforward solution exists for ensuring that, after enough preparation time, $\mathbf{x}(t)$ converges exponentially to $\mathbf{x}^\star$ – thus *eventually* leading to the correct movement. This "naive" solution consists in setting $\mathbf{u}(t)$ to the constant vector $\mathbf{u}^\star$ in Equation S16 (thus $\delta\mathbf{u}(t) = 0$ throughout preparation). Note that for the full nonlinear model, $\mathbf{u}^\star = \mathbf{x}^\star - \mathbf{W}\phi\left[\mathbf{x}^\star\right] - \overline{\mathbf{h}}$. This constant input is provided during movement preparation and removed at the desired time of movement onset.

S4.2   Classical LQR solution

When no specific constraints on $\mathbf{u}(t)$ are imposed, the minimization of Equation S18 is given by the celebrated linear quadratic regulator (LQR). Specifically, the optimal control input $\mathbf{u}_\mathrm{opt}(t) = \mathbf{u}^\star + \delta\mathbf{u}_\mathrm{opt}(t)$ takes the form of (instantaneous) linear state feedback (Figure S2A):

$$\delta\mathbf{u}^\mathrm{opt}(t) = \mathbf{K}\,\delta\mathbf{x}(t) \qquad \text{with} \qquad \mathbf{K} = -\lambda^{-1}\mathbf{P} \tag{S19}$$

where $\mathbf{P}$ is a symmetric, positive definite matrix, obtained as the solution to the following Riccati equation:

$$\mathbf{A}^T\mathbf{P} + \mathbf{P}\mathbf{A} - \lambda^{-1}\mathbf{P}\mathbf{P} + \mathbf{Q} = 0 \tag{S20}$$

(we will recover this optimal feedback law in Section S4.5 as part of a more general mathematical derivation; for now, we refer to standard texts, e.g. Skogestad and Postlethwaite, 2007). Thus,

to achieve optimal anticipatory control of fast movements, the best strategy for the preparatory phase is to feed back into the circuit a linearly weighted version of the momentary error signal $\delta\mathbf{x}(t)$. The optimal feedback matrix $\mathbf{K}$ turns out to not depend on the choice of distribution $p(\mathbf{x}(t=0))$. For a linear model, this also implies that $\mathbf{K}$ does not depend on the specific movement to be performed, i.e. on the specific state $\mathbf{x}^\star$ to be approached during preparation. Only the steady-state control input $\mathbf{u}^\star$ (in Equation S16) is movement-specific.

S4.3   Control geometry of the ISN model of M1 dynamics

While the optimal LQR strategy described above is difficult to map directly onto a realistic circuit architecture (see below), it can be used to expose the challenges associated with controlling the inhibition-stabilized model of M1 that we use here. Indeed, network activity may be more easily controlled (or "steered") along some directions than along others, and having analytical access to the optimal control inputs (Equations S19 and S20) allows us to quantify this "control geometry". Specifically, we quantify control performance as $\mathcal{E} = \int_0^\infty (\delta\mathbf{x}^T \mathbf{Q} \delta\mathbf{x}) dt$, i.e. our original cost functional $\mathcal{J}$ in Equation S10 without the input energy penalty. We can then ask: what is the smallest such cost $\mathcal{E}_{\min}$ that can be achieved with a fixed input energy budget $\int_0^\infty \|\delta\mathbf{u}\|^2 dt$? We know that $\mathcal{E}_{\min}$ is achieved by the LQR solution $\delta\mathbf{u} = \lambda^{-1} \mathbf{P}_\lambda$ (we use the $\cdot_\lambda$ subscript to make the dependence of $\mathbf{P}$ on λ explicit). It can be shown that the input energy induced by this optimal feedback law is a decreasing function of λ. Thus, all we need to do is find the λ that gives us the desired value of $\int_0^\infty \|\delta\mathbf{u}\|^2 dt$, and evaluate $\mathcal{E}_{\min}$ for this particular λ. Importantly, the result will depend on the state of the cortical network at the beginning of the controlled preparatory phase, relative to the target $\mathbf{x}^\star$.

A simple derivation based on Lemma 1 (Appendix A) shows that starting the control phase from some initial condition $\mathbf{x}^\star + \delta\mathbf{x}_0$ yields a total control cost equal to $\mathcal{J} = \delta\mathbf{x}_0^T \mathbf{P}_\lambda \delta\mathbf{x}_0$. Moreover, the corresponding energy cost is given by $\delta\mathbf{x}_0^T \mathbf{Y} \delta\mathbf{x}_0$ where $\mathbf{Y}$ is the solution to

$$\mathbf{A}_{\mathrm{cl}}^T \mathbf{Y} + \mathbf{Y}\mathbf{A}_{\mathrm{cl}} + \lambda^{-2} \mathbf{P}_\lambda \mathbf{P}_\lambda = 0, \tag{S21}$$

and

$$\mathbf{A}_{\mathrm{cl}} \triangleq \mathbf{A} + \mathbf{K} = \mathbf{A} - \lambda^{-1} \mathbf{P}_\lambda \tag{S22}$$

is the effective state matrix governing the dynamics of the closed control loop. For a given $\delta\mathbf{x}_0$, we use a simple root-finding method (bisection with initial interval bracketting) to find the λ that achieves the set, desired energy cost (our fixed "energy budget"). For this λ, we then calculate the associated control cost $\mathcal{E} = \delta\mathbf{x}_0^T (\mathbf{P} - \lambda\mathbf{Y}) \delta\mathbf{x}_0$. This is plotted in Figure S3, for initial deviations of $\mathbf{x}$ from $\mathbf{x}^\star$ chosen to be the top 20 eigenvectors of $\mathbf{Q}$, ranked by their respective eigenvalues ν_i (Equation S13).

Figure S3 (right) shows that there is "no free lunch": preparatory deviations from $\mathbf{x}^\star$ that induce the worst motor errors (the top eigenvectors of $\mathbf{Q}$, with the largest eigenvalues ν_i) are also those that are the most difficult to control, i.e. for which the minimal control cost $\mathcal{E}_{\min}$ will be largest for a fixed input energy budget.

From the point of view of dynamical systems, this result is rather intuitive. The optimal initial conditions $\{\mathbf{x}_k^\star\}$ (found via optimization to achieve the required torques; Section S2) are positioned in state space where the flow induced by the recurrent connectivity is strong – strong enough to elicit rich transients that can be decoded into torques patterns that grow transiently before decaying. To steer M1 towards (and maintain it at) these states, the input $\delta\mathbf{u}(t)$ (and the steady input $\mathbf{u}^\star$) must work against the strong local flow of the recurrent dynamics. This requires large input energy. From a physiological standpoint, this is also intuitive. The optimal initial states $\{\mathbf{x}_k^\star\}$ are shown to be states in which the E/I balance is momentarily broken (Hen-
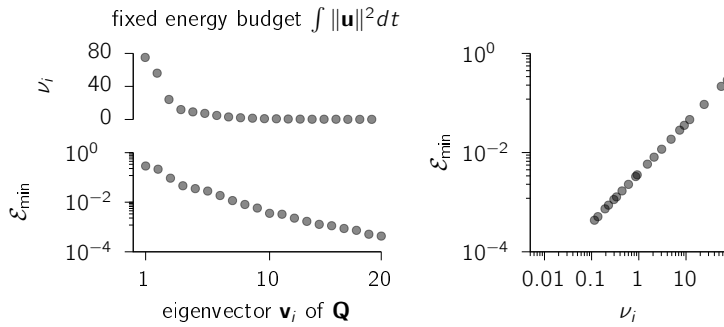
Figure S3: **Left:** eigenvalues ν_i (top) and minimum control cost $\mathcal{E}_{\min}$ achievable given a fixed energy budget (see text), for the top 20 eigenvectors of the observability Gramian $\mathbf{Q}$ defined in Equation S13. Note that ν_i is also the motor error $\mathcal{C}$ experienced if the cortical network remains displaced by $\mathbf{v}_i$ away from the optimal preparatory state $\mathbf{x}^\star$ when movement is initiated. **Right:** same ν_i and $\mathcal{E}_{\min}$ as shown on the left, plotted against each other.

nequin et al., 2014). Much input energy must be spent to sustain an E/I imbalance in a network whose connectivity strives to maintain balance.

S4.4 Optimality under neural constraints

The linear quadratic regulator presented in Section S4.2 brings the fundamental insight that control can (and in fact, should) be achieved via a feedback loop (Figure S2A). Such a loop could technically be embedded directly as a modification of the recurrent connectivity within M1, as all that matters for the control cost is the effective closed-loop state matrix $\mathbf{A} + \mathbf{K}$. However, this would make it very difficult to switch the loop ON when movement preparation must begin, and OFF again when the movement is triggered. A more flexibly strategy would be to have the loop pass through another brain area, and gain-modulate this area (e.g. via inhibitory drive) to close or open the loop when appropriate.

A natural candidate structure for mediating such cortico-cortical feedback is the motor thalamus, which has been shown to be causally involved in movement preparation (Guo et al., 2017). Importantly, basic anatomy and physiology pose constraints on the type of connectivity and dynamics around the control loop, such that we will have to adapt the classical LQR theory to derive plausible circuit mechanisms. In particular, the thalamus is not innervated by the local inhibitory interneurons of M1, so feedback will have to computed based on the activity of (some of) the excitatory cells only, precluding full-state feedback. Moreover, the optimal LQR gain matrix $\mathbf{K}$ (given by Equations S19 and S20) contains both positive and negative elements with no structure; this violates Dale's law, i.e. that neurons can be either excitatory or inhibitory but are never of a mixed type. Finally, the classical LQR solution prescribes *instantaneous* state feedback, whereas thalamic neurons will have to integrate their inputs on finite timescales, thereby introducing some "inertia", or lag, in the feedback loop. In the rest of this section, we flesh out these biological constraints in more detail, and show that all of these limitations can be addressed mathematically, to eventually yield optimal control via a realistic thalamocortical feedback loop (see Figure S2B-D for a graphical overview).

S4.5 Feedback based on excitatory neurons only

Here, we incorporate the key biological constraints that feedback from the cortex onto itself via the thalamus will have to originate from the excitatory cells only. Thus, instead of $\delta\mathbf{u}(t) =$

243 $\mathbf{K}\delta\mathbf{x}(t)$, we look for a feedback matrix of the form (Figure S2B)

$$\mathbf{K} = \mathbf{Z}\boldsymbol{\Gamma} \tag{S23}$$

244 where $\boldsymbol{\Gamma} \triangleq [\mathbf{I}_{N_\mathrm{E}}\ \mathbf{0}_{N_\mathrm{E} \times N_\mathrm{I}}]$ singles out the activity of the E neurons when computing the control
245 input $\mathbf{K}\delta\mathbf{x}$, and $\mathbf{Z}$ is an $N \times N_\mathrm{E}$ matrix of free parameters. To gain generality (which we will
246 need later), we also assume that the control input enters the network through a matrix $\mathbf{B}$, i.e.
247 the closed-loop state matrix (Equation S22) becomes $\mathbf{A}_\mathrm{cl} = \mathbf{A} + \mathbf{BZ}\boldsymbol{\Gamma}$. We now derive algebraic
248 conditions of optimality for $\mathbf{Z}$, along with a gradient-based method to find the optimal $\mathbf{Z}$ that
249 fulfills them.

250 First, we use Lemma 1 in Appendix A to rewrite the cost function $\mathcal{J}$ in Equation S18 as:

$$\mathcal{J}(\mathbf{Z}) = \mathrm{trace}(\mathbf{P}) \tag{S24}$$

251 where $\mathbf{P}$ satisfies

$$0 = \mathbf{G}(\mathbf{P}, \mathbf{Z}) \triangleq \mathbf{A}_\mathrm{cl}^T\mathbf{P} + \mathbf{P}\mathbf{A}_\mathrm{cl} + \mathbf{Q} + \lambda\boldsymbol{\Gamma}^T\mathbf{Z}^T\mathbf{Z}\boldsymbol{\Gamma}. \tag{S25}$$

252 Note that $\mathcal{J}$ in Equation S24 is now a function of the feedback matrix $\mathbf{K}$, and therefore of the
253 parameter matrix $\mathbf{Z}$. To minimize $\mathcal{J}$ w.r.t. $\mathbf{Z}$ subject to the constraint in Equation S25, we
254 introduce the Lagrangian:

$$\mathcal{L}(\mathbf{P}, \mathbf{Z}, \mathbf{S}) \triangleq \mathrm{trace}(\mathbf{P}) + \mathrm{trace}(\mathbf{G}(\mathbf{P}, \mathbf{Z})\,\mathbf{S}) \tag{S26}$$

255 where $\mathbf{S}$ is a symmetric matrix of Lagrange multipliers (the matrix equality in Equation S25 is
256 symmetric, thus effectively providing $N(N+1)/2$ constraints). After some matrix calculus, we
257 obtain the following coupled optimality conditions:

$$
\begin{aligned}
0 &= \partial\mathcal{L}/\partial\mathbf{P} &=& \quad \mathbf{A}_\mathrm{cl}\mathbf{S} + \mathbf{S}\mathbf{A}_\mathrm{cl}^T + \mathbf{I} & \tag{S27}\\
0 &= \partial\mathcal{L}/\partial\mathbf{S} &=& \quad \mathbf{G}(\mathbf{P}, \mathbf{Z}) & \tag{S28}\\
0 &= \partial\mathcal{L}/\partial\mathbf{Z} &=& \quad 2\mathbf{B}^T\left(\mathbf{P} + \lambda\mathbf{BZ}\boldsymbol{\Gamma}\right)\mathbf{S}\boldsymbol{\Gamma}^T. & \tag{S29}
\end{aligned}
$$

258 When the two Lyapunov equations Equations S27 and S28 are satisfied, the second term
259 (trace($\mathbf{GS}$)) in $\mathcal{L}$ vanishes, such that $\partial\mathcal{L}/\partial\mathbf{Z}$ of Equation S29 is in fact the gradient of trace($\mathbf{P}$)
260 w.r.t. $\mathbf{Z}$ subject to the algebraic constraint of Equation S25. We use this gradient equation,
261 together with the L-BFGS optimizer (Byrd et al., 1995) to find the optimal parameter matrix
262 $\mathbf{Z}$. We then recover the optimal feedback gain matrix $\mathbf{K}$ according to Equation S23. We start
263 each optimization by setting $\mathbf{Z} = \overline{\mathbf{K}}\boldsymbol{\Gamma}^T(\boldsymbol{\Gamma}\boldsymbol{\Gamma}^T)^{-1}$, where $\overline{\mathbf{K}}$ is the classical LQR solution to the
264 same problem, such that $\mathbf{Z}\boldsymbol{\Gamma} = \overline{\mathbf{K}}$.

S4.6 Dale's law

266 The previous subsection showed how to obtain a gain matrix $\mathbf{K}$ of size $N \times N_\mathrm{E}$ that implements
267 optimal, instantaneous cortico-cortical feedback originating from the excitatory cells. However,
268 this optimal matrix typically has a mix of positive and negative elements that are not specifi-
269 cally structured. To implement the more realistic feedback architecture shown in Figure S2C,
270 implicating the motor thalamus and M1 layer 4 (M1-L4), we seek a decomposition of the form

$$\mathbf{K} \approx \underbrace{\mathbf{K}_{xz}}_{(+|-)} \underbrace{\mathbf{K}_{zy}}_{(+)} \underbrace{\mathbf{K}_{yx}}_{(+)} \tag{S30}$$

271 where $\mathbf{K}_{yx}$ (M1 to thalamus) is an $N_\mathrm{E} \times N$ matrix of non-negative elements, $\mathbf{K}_{zy}$ (thalamus
272 to M1-L4) is an $M \times N_\mathrm{E}$ matrix of non-negative elements, and $\mathbf{K}_{xz}$ (M1-L4 to the recurrent
273 M1 network) is an $N \times M$ matrix composed of M_E non-negative columns and M_I non-positive

columns (thus $M = M_\mathrm{E} + M_\mathrm{I}$). Such a sign-structured decomposition will allow optimal control to be performed through the more realistic feedback architecture shown in Figure S2C, with corresponding dynamics of the form:

$$\mathrm{M1} \qquad \tau\frac{d\mathbf{x}}{dt} = -\mathbf{x}(t) + \mathbf{W}\phi\left[\mathbf{x}(t)\right] + \overline{\mathbf{h}} + \mathbf{h}(t) + \tilde{\mathbf{u}}^\star + \mathbf{K}_{xz}\phi[\mathbf{z}(t)] \qquad (\mathrm{S31})$$

$$\mathrm{M1\text{-}layer\ 4} \qquad \mathbf{z}(t) = \mathbf{K}_{zy}\phi[\mathbf{y}(t)]$$

$$\mathrm{Thal.} \qquad \mathbf{y}(t) = \mathbf{K}_{yx}\phi[\mathbf{x}(t)]$$

where

$$\tilde{\mathbf{u}}^\star = \mathbf{x}^\star - (\mathbf{W} + \mathbf{K})\phi(\mathbf{x}^\star) - \overline{\mathbf{h}} \qquad (\mathrm{S32})$$

is a condition-dependent steady input given to the network during movement preparation so as to achieve the desired fixed point $\mathbf{x}^\star$.

To achieve this decomposition, we note that without loss of generality we can choose $\mathbf{K}_{yx} = [\mathbf{R}\ \mathbf{0}_{N_\mathrm{E}\times N_\mathrm{I}}]$ – where $\mathbf{R}$ is a random, element-wise positive $N_\mathrm{E} \times N_\mathrm{E}$ matrix – and apply the algorithm developed in Section S4.5 now with $\mathbf{\Gamma} = \mathbf{K}_{yx}$. This will return an optimal $N \times N_\mathrm{E}$ matrix $\mathbf{Z}$ describing feedback from thalamus back to M1, which – as long as $\mathbf{R}$ is invertible – will achieve the same minimum cost as if $\mathbf{R}$ had been set to $\mathbf{I}_{N_\mathrm{E}}$ (as in Section S4.5). Here, we simply draw each element of $\mathbf{R}$ from Bernoulli(p), i.e. random sparse projections (the magnitude of $\mathbf{R}$ does not matter at this stage, as only the product $\mathbf{Z}\mathbf{\Gamma}$ does; $\mathbf{R}$ will be renormalized later below). We now need to decompose this optimal feedback matrix as $\mathbf{Z} = \mathbf{K}_{xz}\mathbf{K}_{zy}$, with the same sign constraints as in Equation S30. We approach this via optimization, by minimizing the squared error implied by the decomposition, plus an 2-norm regularizer:

$$\frac{\|\mathbf{Z} - \mathbf{K}_{xz}\mathbf{K}_{zy}\|_\mathrm{F}^2}{\|\mathbf{Z}\|_\mathrm{F}^2} + \gamma\left(\|\mathbf{K}_{xz}\|_\mathrm{F}^2 + \|\mathbf{K}_{zy}\|_\mathrm{F}^2\right) \qquad (\mathrm{S33})$$

We parameterize each element of $\mathbf{K}_{xz}$ and $\mathbf{K}_{zy}$ as $\pm z^2$, where z is a free parameter to be optimized, and the $\pm$ sign enforces the sign structure written in Equation S30. Minimization is achieved using BFGS and typically converges in a few tens of iterations. We note that the product $\mathbf{K}_{xz}\mathbf{K}_{zy}\mathbf{K}_{yx}$ is invariant to any set of rescalings of the individual matrices as long as they cancel out to 1. Thus, after optimization, we re-balance the three matrices such that they have identical Frobenius norms. This is mathematically optional, but ensures that firing rates in M1, thalamus and M1-L4 have approximately the same dynamic range.

Importantly, we find that as long as the number of M1-L4 neurons (M) is chosen sufficiently large, the decomposition of $\mathbf{Z}$ that we obtain is almost exact, which implies that the dynamics of Equation S31 still achieves optimal anticipatory control of movement under the architectural constraint of Equation S23.

S4.7   Taking into account integration dynamics in thalamus and M1-layer 4

The optimal control solution that we arrived at in Equation S31 still relies on instantaneous feedback from cortex back onto itself. However, neurons in the thalamus and in M1's input layer have their own integration dynamics – this will introduce lag around the loop, which must be taken into account when designing the optimal feedback. We therefore include these dynamics:

$$\mathrm{M1} \qquad \tau\frac{d\mathbf{x}}{dt} = -\mathbf{x}(t) + \mathbf{W}\phi\left[\mathbf{x}(t)\right] + \overline{\mathbf{h}} + \mathbf{h}(t) + \tilde{\mathbf{u}}^\star + \mathbf{K}_{xz}\phi[\mathbf{z}(t)] \qquad (\mathrm{S34})$$

$$\mathrm{M1\text{-}layer\ 4} \qquad \tau_z\frac{d\mathbf{z}}{dt} = -\mathbf{z} + \mathbf{K}_{zy}\phi[\mathbf{y}(t)]$$

$$\mathrm{Thal.} \qquad \tau_y\frac{d\mathbf{y}}{dt} = -\mathbf{y} + \mathbf{K}_{yx}\phi[\mathbf{x}(t)]$$

where the steady input $\tilde{\mathbf{u}}^\star$ is again given by Equation S32, and $\{\tau_y, \tau_z\}$ are the single-neuron time constants in the thalamus and the cortical input layer. We then seek the optimal connectivity matrices $\{\mathbf{K}_{xz}, \mathbf{K}_{zy}, \mathbf{K}_{yx}\}$ to fulfill the same optimal-control principles as before, namely the minimization of the cost functional in Equation S18. In order to do that, we note that the dynamics of $\mathbf{x}$ (M1 activity) in the linear regime do not change if the system of differential equations in Equation S34 is simplified as

$$\text{M1} \qquad \tau\frac{d\mathbf{x}}{dt} = (\mathbf{W} - \mathbf{I})\mathbf{x}(t) + \overline{\mathbf{h}} + h(t) + \tilde{\mathbf{u}}^\star + \mathbf{K}\mathbf{z}(t) \tag{S35}$$

$$\text{M1-layer 4} \qquad \tau_z\frac{d\mathbf{z}}{dt} = -\mathbf{z} + \mathbf{y}(t)$$

$$\text{Thal.} \qquad \tau_y\frac{d\mathbf{y}}{dt} = -\mathbf{y} + \mathbf{x}(t)$$

where $\mathbf{K} = \mathbf{K}_{xz}\mathbf{K}_{zy}\mathbf{K}_{yx}$ summarizes the three connectivity matrices around the loop into one effective feedback gain matrix. This formulation allows us to combine the steps developed in Sections S4.5 and S4.6 to find the optimal connectivity matrices.

Specifically, we apply the algorithm of Section S4.5 to an augmented system with state matrix

$$\mathbf{A}' \triangleq \begin{bmatrix} \mathbf{A} & \mathbf{0}_{N \times N_{\mathrm{E}}} & \mathbf{0}_{N \times N_{\mathrm{E}}} \\ (\tau/\tau_y)[\mathbf{R}\ \mathbf{0}] & -(\tau/\tau_y)\mathbf{I}_{N_{\mathrm{E}}} & \mathbf{0}_{N_{\mathrm{E}}} \\ \mathbf{0} & (\tau/\tau_z)\mathbf{I}_{N_{\mathrm{E}}} & -(\tau/\tau_z)\mathbf{I}_{N_{\mathrm{E}}} \end{bmatrix}, \tag{S36}$$

input matrix

$$\mathbf{B}' \triangleq \begin{bmatrix} \mathbf{I}_N \\ \mathbf{0}_{N_{\mathrm{E}} \times N} \\ \mathbf{0}_{N_{\mathrm{E}} \times N} \end{bmatrix}, \tag{S37}$$

quadratic cost weighting matrix

$$\mathbf{Q}' \triangleq \begin{bmatrix} \mathbf{Q} & \mathbf{0}_{N \times N_{\mathrm{E}}} & \mathbf{0}_{N \times N_{\mathrm{E}}} \\ \mathbf{0}_{N_{\mathrm{E}} \times N} & \mathbf{0}_{N_{\mathrm{E}} \times N_{\mathrm{E}}} & \mathbf{0}_{N_{\mathrm{E}} \times N_{\mathrm{E}}} \\ \mathbf{0}_{N_{\mathrm{E}} \times N} & \mathbf{0}_{N_{\mathrm{E}} \times N_{\mathrm{E}}} & \mathbf{0}_{N_{\mathrm{E}} \times N_{\mathrm{E}}} \end{bmatrix} \tag{S38}$$

and feedback input parameterized as

$$\mathbf{u} = \mathbf{K}\mathbf{x} \quad \text{with} \quad \mathbf{K} = \mathbf{Z}\boldsymbol{\Gamma} = \mathbf{Z}\begin{bmatrix} \mathbf{0}_{N_{\mathrm{E}} \times N} & \mathbf{0}_{N_{\mathrm{E}} \times N_{\mathrm{E}}} & \mathbf{I}_{N_{\mathrm{E}}} \end{bmatrix} \tag{S39}$$

In Equation S36, the matrix $\mathbf{R}$ is again a random matrix of sparse positive connections from M1 to thalamus (cf. Section S4.6 above). The optimal $\mathbf{Z}$ (Section S4.5) corresponds to the product $\mathbf{K}_{xz}\mathbf{K}_{zy}$, which we can further decompose under sign constraints to recover the individual connectivity matrices $\mathbf{K}_{xz}$ and $\mathbf{K}_{zy}$.

S4.8 Disinhibitory action of the basal ganglia

We model the disinhibitory action of the basal ganglia (BG) on thalamic neurons as an ON-OFF switch: to trigger movement, BG become active (BG neurons not explicitly modelled here) and the thalamic neurons are silenced instantly (i.e. $\mathbf{y}$ is set to $\mathbf{0}$). When this happens, thalamic inputs to M1-L4 vanish and M1-L4 neural activity decays to zero on a time-scale τ_z (see Equation S36). As the activity of L4 neurons decays, these neurons continue to exert an influence on M1 activity through the connectivity matrix $\mathbf{K}_{zx}$. This lead to changes in movement-related M1 dynamics, resulting in small movement errors, which we correct post-hoc by ever-so-slightly re-optimizing the desired initial state $\mathbf{x}^\star$ for each movement. From these new desired states, network dynamics evolves—with the additional inputs from M1-L4 neurons after movement onset—to

produce accurate hand trajectories. Crucially, unlike what we described in Section S2.3, we do not re-optimize the readout matrix $\mathbf{C}$ here. This is because the observability Gramian $\mathbf{Q}$ and thus the closed-loop controller $\mathbf{K}$ depend on $\mathbf{C}$: changing the readout matrix $\mathbf{C}$ at this stage would cause the $\mathbf{K}$ we found to no longer be optimal with respect to $\mathcal{J}$. However, because the closed-loop solution does not depend on the desired fixed points $\mathbf{x}^\star$, we can re-optimize $\mathbf{x}^\star$ and still be guaranteed that the $\mathbf{K}$ that we found remains optimal.

S4.9   Modelling the effect of photoinhibition

To model photoinhibition in our full circuit (whose dynamics are described by Equation S31), we simply add a constant positive input h_{ph} to a subset of cortical inhibitory neurons chosen randomly (see parameters in Table S1), for a duration $T_{\mathrm{ph}} = 400$ ms. This results in an overall decrease in population activity across both excitatory and inhibitory neurons, consistent with the well-known paradoxical effects of adding positive inputs to I cells in inhibition-stabilized networks (Tsodyks et al., 1997; Ozeki et al., 2009; Sanzeni et al., 2019).

We closely followed the analyses described in Li et al. (2016) to uncover how activity recovers along different state-space directions after perturbation. We focused on two reaches, a 0-degree (right) reach and a 180-degree (left) reach. We calculated the coding direction (CD) as the difference between the average firing rates of left and right reaches in unperturbed trials in a 400 ms time window ending at the end of movement preparation (i.e. 400 ms before the control inputs $\mathbf{u}(t)$ are removed). Independent of the CD, we identified the persistent mode (PM) as the direction that maximally separates average firing rates between perturbed and unperturbed trials in the same time window, averaged across the two reach conditions. We found that CD and PM are orthogonal to each other even though if we did not constrain them to be. We found a third remaining mode (RM), constrained to be orthogonal to CD and PM, which captures most of average firing rate activity variance across the two reaches in perturbed and unperturbed trials. The three modes (CD, PM and RM) together capture approximately 98% of the average firing rate variance during that time window. We projected perturbed and unperturbed activity onto these three modes and calculated root-mean-square deviation between perturbed and unperturbed projections over 300 independent perturbation experiments.

S4.10   Variability quenching

We modelled firing rate variability by adding a noisy input term $\boldsymbol{\xi}(t)$ to $\overline{\mathbf{h}}(t)$ in Equation S1, where $\boldsymbol{\xi}(t)$ is modelled as an independent Orstein-Uhlenbeck process for each neuron, with a time constant $\tau_\xi = 20$ ms and standard deviation $\sigma_\xi = 2$ Hz.

In Figure 6, we considered the effect of artificially increasing the dimensionality of the linear readout on the strength of variability quenching. To increase the dimensionality of the readout from 2 to $2+L$, we constructed an augmented linear readout $(\mathbf{C}^T, \tilde{\mathbf{C}}^T)^T$, where $\tilde{\mathbf{C}}$ has dimensions $L \times N_{\mathrm{E}}$. The rows of $\tilde{\mathbf{C}}$ are a set of orthogonal vectors that are also orthogonal to the rows of $\mathbf{C}$. The norm of the row vectors in $\tilde{\mathbf{C}}$ are chosen to be the mean squared singular values of $\mathbf{C}$, such that the corresponding outputs would have roughly the same norms.

To understand the mechanisms underlying variability suppression in the model, we examined the eigenvalues of the effective connectivity matrix in closed-loop (preparation) and in open loop (spontaneous fluctuations). In closed loop, most eigenvalues are more negative than their open-loop counterparts (Figure S4A). Moreover, the eigenvalues that are most shifted to the left in closed loop are associated with eigenvectors with high motor potency (compare Figure S4B and C). Our previous analysis of a similar type of model in Hennequin et al. (2018) showed that
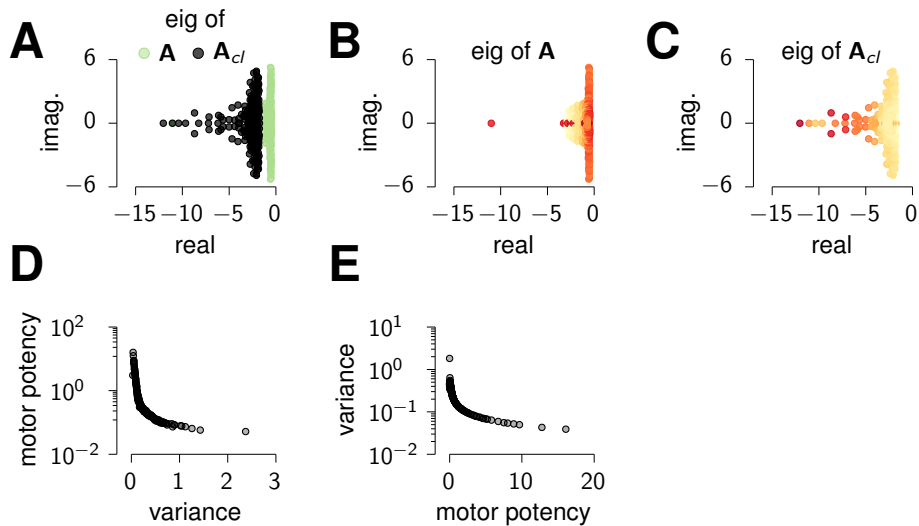
13

Figure S4: **Mechanisms of variability suppression in the model. (A)** Eigenvalues of $\mathbf{A} = \mathbf{W} - \mathbf{I}$ (open loop dynamics of the cortical network, green) and $\mathbf{A}_{\text{cl}} = \mathbf{A} + \mathbf{K}$ (LQR closed loop, black). **(B and C)** Same as in (A), with each eigenvalue colored by the motor potency of the corresponding eigenvector/eigenplanes, using the same color scheme as in Figure 4 of the main text. **(D)** Motor potency of each principal component of the closed-loop activity covariance in the stochastic model, as a function of the amount of total variance it captures. **(E)** Total closed-loop activity variance along the eigenmodes of $\mathbf{Q}$, plotted as a function of their motor potencies.

the more negative an eigenvalue, the less variability in the corresponding eigensubspace. Thus, variability quenching should be stronger along directions associated with more negative eigenvalues, and because of the selective shift of "potent" eigenvalues, our model predicts stronger variability suppression in directions of high motor potency. Indeed, we found an inverse relationship between the motor potency of the principal components of the closed-loop fluctuations, and the amount of variance they capture (Figure S4D). Similarly, the amount of variance captured by the eigenvectors of $\mathbf{Q}$ (those used in Figure 4) is inversely related to their motor potency (Figure S4E).

S5 Data analysis and comparison with model

S5.1 Task and neural recordings

We analyzed neural recordings of a monkey J performing a delayed reaching task (data courtesy of Mark Churchland, Matt Kaufman and Krishna Shenoy). Both the task and dataset have been described in detail previously (Churchland et al., 2010b). Briefly, monkey J performed center-out reaches on a fronto-parallel screen. At the beginning of each trial, monkey J fixated on the centre of the screen for some time, after which a target appeared on the screen. A variable delay period (0–1000 ms) ensued, followed by a go cue instructing the monkey to reach towards the target. In this paper, we analyzed only eight movement conditions, corresponding to the straight reaches that were most similar to the ones we modelled (Figure S5A). Moreover, we restricted our analysis to the trials with delay periods longer than 400 ms, though opening up to shorter delays did not substantially affect our results.

Recordings were made in the dorsal premotor and primary motor areas. We preprocessed spike trains of 123 neurons, following the same procedure outlined in Churchland et al. (2012). Briefly,
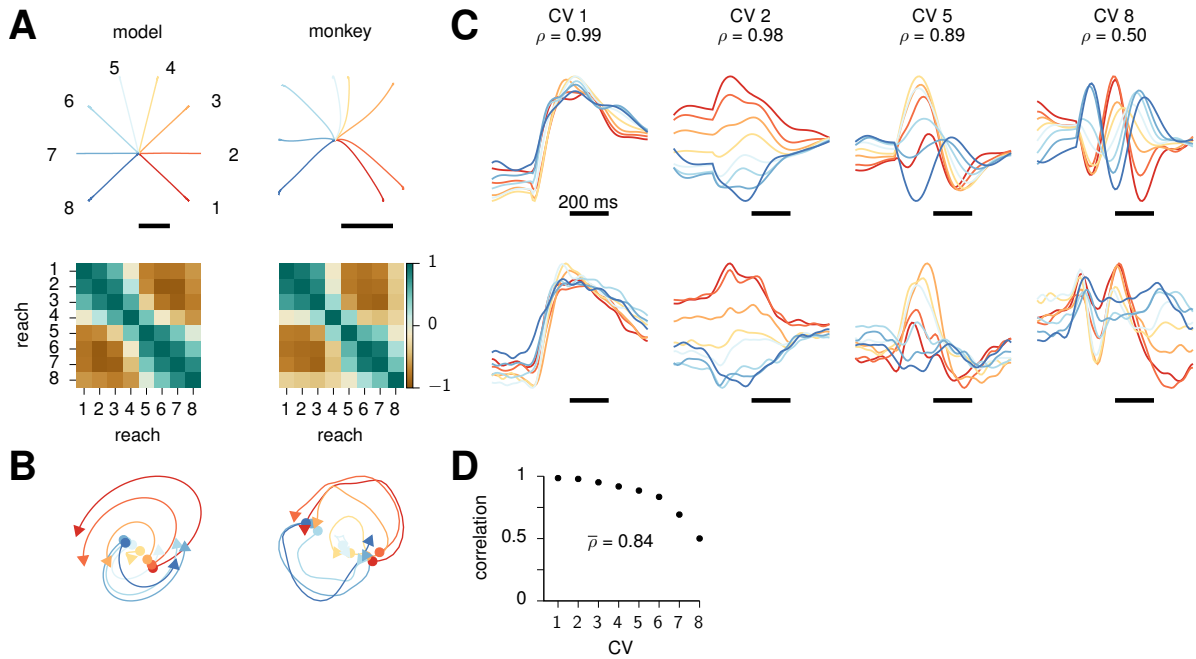
Figure S5: **Model and monkey data comparison.** **(A)** Top: model (left) and monkey (right) hand trajectories for eight straight reaches (color-coded). Black bars denote 10 cm. Monkey hand trajectories are averaged across trials with delays longer than 400 ms. Bottom: Overlap between the preparatory end-states for model (left) and monkey (right) activity. Reach numbers correspond to those indicated near the model hand trajectories (top left). **(B)** Neural activity in the model (left) and monkey (right) around movement onset, projected into the top jPC plane (see text). **(C)** Timecourse of the 1st, 2nd, 5th and 8th canonical variables in the model (top) and monkey (bottom), for each condition (color-coded). Black scale bars indicate 200 ms from movement onset (note that "movement onset" in the model is re-defined to account for the latency between the go cue and actual movement onset in the monkey; see text). **(D)** Full spectrum of canonical correlations.

we computed the average firing rates for each movement condition (8 straight reaches), further smoothed using a 20 ms Gaussian filter. Firing rates were computed separately for the delay and movement periods, time-locked to target and movement onset respectively; this is necessary because of variable delay periods and reaction times.

S5.2 Overlap between preparatory end-states

We calculated the pairwise overlaps (normalized dot-products) between the preparatory end-states in both model and monkey data for all reaches. Preparatory end states are defined as the activity states reached at the end of movement preparation (monkey activity aligned to go cue). In both model and monkey data, preparatory end-states are similar (Figure S5A, bottom) for reaches with similar hand trajectories (hand trajectories Figure S5A, top), but negatively related for more distant movements.

S5.3 jPCA

We used the method described in Churchland et al. (2012) to identify state-space directions in which activity trajectories rotate most strongly. Briefly, we used numerical optimization to fit skew-symmetric linear dynamical systems of the form $\dot{\mathbf{x}} = \mathbf{S}\mathbf{x}$ that best captured model and

15

monkey data, in a 400 ms window starting 200 ms before movement onset for the monkey data, and starting at the go cue (i.e. the moment the thalamic neurons are silenced) in the model. We projected model and monkey activity trajectories in this window onto a plane spanned by the top two eigenvectors of $\mathbf{S}$ (Figure S5B).

S5.4 Alignment index

To calculate the alignment index, we closely followed the methods described in Elsayed et al. (2016). The alignment index $\mathcal{A}$ is defined as the percentage of across-condition variance during movement captured by the top K principal components (PCs) of the delay-period activity:

$$\mathcal{A} = \text{Tr}\left(\frac{\mathbf{D}_{\text{prep}}^T \mathbf{C}_{\text{move}} \mathbf{D}_{\text{prep}}}{\sum_{i=1}^{K} \sigma_{i,\text{prep}}^2} \right). \tag{S40}$$

where the K columns of $\mathbf{D}_{\text{prep}}$ are the top K principal components of prep. activity ("prep-PCs"), $\mathbf{C}_{\text{move}}$ is the covariance matrix of move. activity, and $\sigma_{i,\text{prep}}^2$ is the prep. activity variance captured by the ith prep-PC. We choose K such that $\mathbf{D}_{\text{prep}}$ captures 85% of the variabce in prep. activity ($K = 12$ for monkey data and $K = 4$ for the circuit model). Here, we define prep. activity as the delay-period activity during a 300 ms window starting 150 ms after target onset; the activity is calculated time-locked to target onset. Similarly, move. activity is defined as activity during a 300 ms window starting 100 ms prior to movement onset; the activity is calculated using firing rates time-locked to movement onset.

Methods for calculating the control of the alignment index are described in detail in the Supplementary Material of Elsayed et al. (2016) and is not reproduced here. The model alignment index is calculated in the same way as that of the neural data. However, there is a mismatch between the time of movement onset in the model and that in neural data. In the model, movement starts immediately after the rapid change in neural activity (i.e., when control inputs are removed). In the monkey data, however, movement begins roughly 200 ms after activity starts rapidly changing. To roughly align the temporal profile of neural activity in the model and data, we defined the time of "movement onset" in the model to be 200 ms after the "go cue", attributing the delay in movement to delays in downstream motor processes not considered in this model.

S5.5 Canonical-correlation analysis

To compare model and monkey activity, we performed canonical-correlation analysis (CCA) on activity in a time window starting 400 ms before and ending 400 ms after movement onset (see alignment index discussion above, for nuance in defining the time of movement onset in model activity). To avoid overfitting to noise in CCA (Sussillo et al., 2015; Raghu et al., 2017), we first reduced the dimensionality of the two data sets, by projecting activity onto the top 13 (monkey) and 8 (model) principal components; the number of principal components are chosen to capture 90% of the across-condition activity variance in the two datasets. We then calculated the canonical correlations between the two reduced data sets (see Press, 2011, for a numerically stable implementation of CCA). We found that monkey and model activity are similar across time and reaching movements, with a high average canonical correlation $\overline{\rho} = 0.84$ (Figure S5 C and D). We obtained similar results when we varied the number of principal components kept in the two data sets (which in turn varied the number of canonical variables).

S5.6 Trial-by-trial variability

To quantify trial-by-trial fluctuations in the monkey data, we calculated the total spike count $c_{imk}(t)$ for neuron i in condition m and trial k in a 150 ms time window centered at time t. We focused on the 8 straight-reaching conditions shown in Figure S5A and trials with a delay period longer than 400 ms. We calculated the normalized spike count residuals

$$\tilde{c}_{imk}(t) = \frac{c_{imk}(t)}{\sqrt{\mu_{im}(t)}} - \sqrt{\mu_{im}(t)}, \tag{S41}$$

where $\mu_{im}(t)$ is the average of c_{imk} across trials. The Fano factor of neuron i in condition m is the variance of $\tilde{c}_{imk}(t)$ across trials. In Section S5.6B, we reproduced the results previously presented in Churchland et al. (2010a): the population- and condition-averaged Fano factor drops at target onset.

To dissect how variability is quenched along different state-space directions, we generalized this notion of Fano factors and defined the projected spike count variability along some state space direction $\mathbf{d}$ to be

$$\mathcal{V}(\mathbf{d}, t) = \left\langle (\mathbf{d}^T \tilde{\mathbf{c}}_{\bullet mk}(t))^2 \right\rangle_{mk}, \tag{S42}$$

where $\tilde{\mathbf{c}}_{\bullet mk} = (\tilde{c}_{1mk}, \tilde{c}_{2mk}, \cdots, \tilde{c}_{Nmk})^T$.

Our model predicts that variability should be quenched preferentially in directions that matter for movement. To test this prediction in monkey data, we extracted three subspaces from trial-averaged data. First, we defined a "coding subspace" (CS) as the subspace spanned by average neural activity in movements towards the end of movement preparation. We expect this subspace to be a potent movement subspace as it is spanned by directions that could move activity from a preparatory state corresponding to one reach into that corresponding to another reach. In practice, we considered the trial-averaged firing rates averaged in a 100 ms time window starting 300 ms after target onset. We removed the mean across condition for each neuron and time and constructed a data matrix $\mathbf{X} \in \mathbb{R}^{123 \times 8}$, where each column contains the population firing rate vector for a different movement condition. We only included trials with a delay period longer than 400 ms and smoothed the resulting time-averaged activity with a 30 ms Gaussian kernel. We then performed PCA and extracted the top $K = 4$ principal components, which captured 95% of the variance across conditions. We used these K principal components to define the CS. The average projected spike count in the CS at time t is a weighted average of the projected spike count variability along the K principal components $\mathbf{d}_i$:

$$\frac{1}{\sum_{i=1}^{K} \sigma_i^2} \sum_{i=1}^{K} \sigma_i^2 \, \mathcal{V}(\mathbf{d}_i, t), \tag{S43}$$

where σ_i^2 is the across-condition variance captured by component i. The motivation for this weighted sum is the following. Not only are directions that capture more of the across-condition variance likely more potent (higher prospective error), they are also more reliably estimated in the presence of noise (finite number of trials). Thus, they should be weighed more strongly.

The second subspace we considered is the "late-change subspace" (LCS), spanned by population activity fluctuations experienced towards the end of movement preparation. Since the monkey is able to produce an accurate reach after a delay much shorter than 400 ms (Lara et al., 2018), late-changes in preparatory activity are likely inconsequential. To estimate this subspace, we again considered the trial-and-time averaged firing rates in a 100 ms time window starting 200 ms after target onset. We then calculated the eight within-condition differences between the average population rate vector in this time window and that collected at the end of preparation (i.e. 300 ms after target onset). We further removed the mean across conditions for each neuron

17

and time, and assembled a matrix $\mathbf{Y} \in \mathbb{R}^{123 \times 8}$ as above for the CS. We orthogonalized this collection of vectors against the CS, and performed PCA as for the CS, retaining 95% of the variance across conditions ($K = 7$). The projected spike count variance in the LCS is then calculated as above for the CS.

Finally, we considered a third subspace as an independent estimate of likely potent directions. This subspace, called the "early-change subspace" (ECS), was defined exactly like the LCS except that we considered activity changes *early* during preparation. Specifically, we substracted activity collected at mid-preparation (100 ms window starting 200 ms after target onset) with prep. activity in the first 100 ms from target onset. The rest of the procedure is as described above for the LCS.

symbol	value	unit	description

<div align="center">

PARAMETERS OF THE M1 CIRCUIT MODEL

</div>

symbol	value	unit	description
N_{E}	160	-	number of E units
N_{I}	40	-	number of I units
τ	150	ms	time constant of M1 dynamics
τ_{rise}	50	ms	rise time constant of $\mathbf{h}(t)$
τ_{decay}	500	ms	decay time constant of $\mathbf{h}(t)$
A	implicit	-	set so that $\mathbf{h}(t)$ has a maximum of 5

<div align="center">

ARM MECHANICS AND HAND TRAJECTORIES

</div>

symbol	value	unit	description
ℓ_1	30	cm	length of upper arm link
ℓ_2	33	cm	length of lower arm link
M_1	1.4	kg	mass of upper arm link
M_2	1.0	kg	mass of lower arm link
D_2	16	cm	center of mass of lower link, away from elbow
I_1	0.025	kg m^{-2}	moment of inertia of upper link
I_2	0.045	kg m^{-2}	moment of inertia of lower link
θ_1^{init}	10.	deg.	value of θ_1 at rest
θ_2^{init}	143.54	deg.	value of θ_2 at rest
$\theta_{\mathrm{reach}}^{(i)}$	$36 \times (i-2)$	deg.	reach angles ($i = 1, \ldots, 8$)
d_{reach}	20	cm	reach distance
τ_{reach}	120	ms	time constant of reach velocity profile

<div align="center">

LQR SOLUTION

</div>

symbol	value	unit	description
λ	0.1	-	input energy penalty in Equation S18

<div align="center">

THALAMO-CORTICAL CIRCUIT MODEL

</div>

symbol	value	unit	description
λ	0.01	-	input energy penalty in Equation S18
p	0.2	-	density of random connections from M1 to thal.
M_{E}	100	-	number of E units in M1 L4
M_{I}	100	-	number of I units in M1 L4
τ_y	10	ms	neuronal time constant in thalamus
τ_z	10	ms	neuronal time constant in M1 L4

<div align="center">

PHOTOINHIBITION

</div>

symbol	value	unit	description
N_{ph}	100	-	number of M1 I units perturbed (60%)
h_{ph}	3	-	input to perturbed I units during photoinhibition
T_{ph}	400	ms	duration of photoinhibition

Table S1: Generic parameters used throughout all simulations

A   Core lemma

Lemma 1 *The matrix integral*

$$\mathbf{Q} \triangleq \int_0^\infty e^{t\mathbf{A}^T}\mathbf{C}^T\mathbf{C}e^{t\mathbf{A}}\ dt \tag{S44}$$

satisfies the continuous-time Lyapunov equation

$$\mathbf{A}^T\mathbf{Q} + \mathbf{Q}\mathbf{A} + \mathbf{C}^T\mathbf{C} = \mathbf{0} \tag{S45}$$

This lemma is central to the theory of linear quadratic control, where cost functions are often of the form of integrated squared functions of the state, output, or input, under linear dynamics (as are the costs used in this paper). It allows one to manipulate these integrals algebraically, and compute them numerically by solving a linear matrix equation (e.g. Bartels and Stewart, 1972).

References

Bartels, R. H. and Stewart, G. W. (1972). Solution of the matrix equation AX+XB=C. *Communications of the ACM*, 15:820–826.

Byrd, R. H., Lu, P., Nocedal, J., and Zhu, C. (1995). A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*, 16:1190–1208.

Churchland, M. M., Byron, M. Y., Cunningham, J. P., Sugrue, L. P., Cohen, M. R., Corrado, G. S., Newsome, W. T., Clark, A. M., Hosseini, P., Scott, B. B., et al. (2010a). Stimulus onset quenches neural variability: a widespread cortical phenomenon. *Nat Neurosci*, 13:369–378.

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(7405):51.

Churchland, M. M., Cunningham, J. P., Kaufman, M. T., Ryu, S. I., and Shenoy, K. V. (2010b). Cortical preparatory activity: representation of movement or first cog in a dynamical machine? *Neuron*, 68:387–400.

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.

Guo, Z. V., Inagaki, H. K., Daie, K., Druckmann, S., Gerfen, C. R., and Svoboda, K. (2017). Maintenance of persistent activity in a frontal thalamocortical loop. *Nature*, 545:181–186.

Hennequin, G., Ahmadian, Y., Rubin, D. B., Lengyel, M., and Miller, K. D. (2018). The dynamical regime of sensory cortex: stable dynamics around a single stimulus-tuned attractor account for patterns of noise variability. *Neuron*, 98:846–860.

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.

Kao, T.-C. and Hennequin, G. (2019). Neuroscience out of control: control-theoretic perspectives on neural circuit dynamics. *Curr Opin Neurobiol*, 58:122–129.

Kaufman, M. T., Seely, J. S., Sussillo, D., Ryu, S. I., Shenoy, K. V., and Churchland, M. M. (2016). The Largest Response Component in the Motor Cortex Reflects Movement Timing but Not Movement Type. *eNeuro*, 3(4):0085–16.2016.

Lara, A. H., Elsayed, G. F., Zimnik, A. J., Cunningham, J. P., and Churchland, M. M. (2018). Conservation of preparatory neural events in monkey motor cortex regardless of how movement is initiated. *eLife*, 7:e31826.

Li, N., Daie, K., Svoboda, K., and Druckmann, S. (2016). Robust neuronal dynamics in premotor cortex during motor planning. *Nature*, 532(7600):459–464.

Li, W. and Todorov, E. (2004). Iterative linear quadratic regulator design for nonlinear biological movement systems. *International Conference on Informatics in Control, Automation and Robotics*.

Ozeki, H., Finn, I. M., Schaffer, E. S., Miller, K. D., and Ferster, D. (2009). Inhibitory stabilization of the cortical network underlies visual surround suppression. *Neuron*, 62(4):578–592.

Press, W. H. (2011). Canonical correlation clarified by singular value decomposition.

Raghu, M., Gilmer, J., Yosinski, J., and Sohl-Dickstein, J. (2017). Svcca: singular vector canonical correlation analysis for deep learning dynamics and interpretability. In *NeurIPS*, pages 6078–6087.

Sanzeni, A., Akitake, B., Goldbach, H. C., Leedy, C. E., Brunel, N., and Histed, M. H. (2019). Inhibition stabilization is a widespread property of cortical networks. *bioRxiv*, page 656710.

Shenoy, K. V., Sahani, M., and Churchland, M. M. (2013). Cortical control of arm movements: a dynamical systems perspective. *Ann. Rev. Neurosci.*, 36:337–359.

Skogestad, S. and Postlethwaite, I. (2007). *Multivariable feedback control: analysis and design*, volume 2. Wiley New York.

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.

Tsodyks, M. V., Skaggs, W. E., Sejnowski, T. J., and McNaughton, B. L. (1997). Paradoxical effects of external modulation of inhibitory interneurons. *J Neurosci*, 17(11):4382–4388.