Compensating for a sensorimotor delay requires a predictor that convolves over a memory buffer of efference copies

Eric Maris*

Donders Institute for Brain, Cognition, and Behaviour, Radboud University, P.O. Box 9104 HE Nijmegen

Corresponding Author:

Eric Maris

e-mail: eric.maris@donders.ru.nl

tel: +31243612651

The number of words of the abstract: 207

The number of figures: 6

The number of tables: 0

Abstract

Effective motor control requires sensory feedback but is seriously complicated by the sensorimotor delay (SMD), which is the time delay between the state of the body at feedback generation and the arrival in the body's muscles of the feedback-informed motor command. I describe and evaluate three SMD compensation mechanisms: gain scaling, the convolution predictor, and the Smith predictor. These mechanisms are implemented using control theory results for linear dynamical systems, which are well motivated for balance control. These mechanisms are investigated theoretically and by simulations of balance control, both free standing and while riding a bicycle. I demonstrate that compensating for a SMD requires a convolution predictor, which involves a convolution over a memory buffer of efference copies and an initial condition obtained from a state observer that is based on a delayed-input forward model. The performance of a convolution predictor does not crucially depend on its exact computational implementation because a similar performance is obtained with an approximate convolution using a boxcar kernel. I also demonstrate that gain scaling is an effective SMD compensation mechanism but is not sufficient to compensate for a neurobiological SMD. Finally, I demonstrate that the Smith predictor is an ineffective and neurobiologically implausible SMD compensation mechanism for an unstable mechanical system.

Keywords

Sensorimotor delay; motor control; prediction; state observer; efference copy; balance control; standing balance; bicycle balance; closed-loop feedback control; linear quadratic Gaussian; finite spectrum assignment; Smith predictor; underactuated.

Introduction

The central nervous system (CNS) controls the body via motor commands, and this usually requires sensory feedback about the state of the body (i.e., its position and velocity). Adjusting motor commands based on sensory feedback is complicated by the delay between the sensory feedback (informing us about the state of our body) and the effect of the feedback-based motor commands on the body. This is called the sensorimotor delay (SMD). For example, in balance control, the motor commands that keep the body upright lag the sensory feedback about the body's state relative to gravity. Motor control deteriorates as the SMD is artificially increased [1-3], but repeated exposure to such imposed delays (up to 430 ms.) results in improvements in motor control [4-6]. This suggests that the CNS implements computational mechanisms that compensate for the SMD.

In this paper, I describe and evaluate three SMD compensation mechanisms. I evaluate these mechanisms both theoretically and by simulating their performance in balancing two unstable mechanical systems, a freely standing person and a rider-bicycle combination. The SMD compensation mechanisms are implemented using control theory results for linear dynamical systems [7, 8]. Although mechanical systems involving the human body have nonlinear equations of motion (EoM), for balance control, a linear approximation of these nonlinear EoM is accurate. This is because a balancing body (standing person, rider, bicycle) most of the time stays close to the upright position, which is an unstable fixed point of the nonlinear mechanical system. The linear approximation at this unstable fixed will be accurate as long as the body stays close to it.

Control theory allows to formulate SMD compensation mechanisms in terms of a CNS-internal dynamical system (the computational system) that controls the mechanical system (the plant). This mechanical system is the human body plus the objects attached to it (e.g., a bicycle), and its dynamics depends on the forces that act on it, including gravity and the centripetal force. Control is specified in terms of an optimality criterion/objective. In the time domain, this criterion is usually expressed in terms of a loss function, and for balance control this loss function depends on the difference between the body's orientation and the earth-vertical axis. In the Laplace domain, the optimality criterion is expressed in terms of the poles of transfer functions. The best-known type of control is linear-quadratic-Gaussian (LQG) control [9, 10], which contains all the ingredients that are needed as a formal background for

this paper. Crucially, although I have used LQG in my simulations, the main concepts in this paper do not depend on the exact mathematical specification of the optimality criterion.

In the absence of a SMD, LQG-optimal control for a linear mechanical system is realized by a control action that obeys the separation principle. Under this principle, the optimal control action u is of the following linear type: $u = -K\hat{x}$, in which $-K$ is an output gain and $\hat{x}$ is an estimate of the mechanical system's state. The separation principle involves that the optimal control action u is obtained by two separate optimizations: (1) find the optimal (most accurate) state estimate $\hat{x}$ for a fixed output gain $-K$, and (2) find the optimal output gain $-K$ for a state estimate $\hat{x}$. Under the LQG assumptions, the optimal state estimate $\hat{x}$ is the output of a Kalman filter that takes as its input the sensory feedback y and the control action u, and the optimal output gain $-K$ is the Linear Quadratic Regulator (LQR).

I assume that, also in the presence of a nonzero SMD, the CNS obeys the separation principle: one CNS component is responsible for computing the optimal state estimate $\hat{x}$, and another component for computing the optimal control action $u = -K\hat{x}$. Corresponding to these two components, there are also two types of SMD compensation mechanisms: (1) state prediction, and (2) output gain scaling. State prediction involves that the state estimate $\hat{x}$ is replaced by a state prediction. Two types of state prediction have been proposed: (1) finite spectrum assignment (FSA) [11-17], and (2) the Smith predictor [11, 18-20]. Output gain scaling [21-23] involves downscaling of the output gain, and this is motivated quantitatively in the Laplace domain: by downscaling the output gain, the phase margin of the implied open-loop system is evaluated at a lower frequency, which has a smaller SMD-induced phase lag and therefore a larger phase margin.

All three mechanisms have their origin in engineering, but they generate useful ideas about the computations that are likely to be performed by the CNS. A very important idea is that the CNS uses internal models for motor control [24, 25]. Different internal models have been proposed, and here I focus on the forward and the sensory model: the forward model captures the dynamics of the mechanical system (including its dependence on the muscular input forces), and the sensory model captures the information in the sensory feedback about the mechanical system's state. Also an inverse model has been proposed [26-28], but its function is identical to that of the output gain $-K$, and I will therefore not use this term.

FSA and the Smith predictor differ with respect to the force input to their forward model: in the FSA forward model, the input force is delayed, whereas in the Smith predictor's forward model, it is non-delayed. These different forward models have implications for the CNS' computations: (1) FSA is a state predictor that depends on a memory buffer of efference copies, and (2) the Smith predictor is a forward state estimator with a correction term that depends on a memory buffer of state estimates.

Finite spectrum assignment (FSA) is a poor label for a technique that serves as an inspiration for CNS computations. The label FSA refers to the Laplace domain formulation of the technique, whereas the time domain formulation can be related much easier to CNS computations. In its time domain formulation, FSA is a state predictor produced by convolution. This differs from the Smith predictor, which is a state estimator combined with a term that is intended to correct for the SMD; it is the correction term that justifies the label predictor. In the following, instead of FSA, I will use the label convolution predictor.

The control theory literature on delay compensation is dominated by theoretical analyses and examples with sensory feedback that is a perfect copy of the state variables. Such feedback is called full-state and noise-free. Full-state noise-free feedback is unlikely to be observed in a neurobiological system, and therefore it is important to investigate whether SMD compensation mechanisms can control an unstable mechanical system using sensory feedback in which the states are only partially observable (denoted as incomplete feedback) and noisy. In this paper, I model muscle spindle proprioceptive feedback based on results from sensory neurophysiology: muscle spindle primary afferents scale with exafferent acceleration feedback [29-31], which is incomplete [10]. LQG-optimal control of a mechanical system using incomplete and noisy feedback requires an optimal state estimator (observer). This optimal observer is the Kalman filter [32, 33], which is a CNS-internal dynamical system that can be implemented as a recurrent neural network [10].

I will demonstrate that the convolution predictor is the only plausible SMD compensation mechanism. Although standing balance control does not require a compensation mechanism under a neurobiologically plausible SMD, bicycle balance control is only possible using the convolution predictor. The Smith predictor is an ineffective mechanism for an unstable

mechanical system, and gain scaling is not sufficiently effective for bicycle balance control. The convolution predictor requires a memory buffer of efference copies plus a state observer that is based on a delayed-input forward model. Effective SMD compensation using the convolution predictor does not require the computationally hard convolution; balance control performance is almost identical if a grossly simplified approximate convolution is used. None of the results are specific for the incomplete exafferent acceleration feedback because qualitatively similar results were obtained using full-state feedback.

Results

A model for a standing person's body: the double compound inverted pendulum

For a concrete description of the SMD compensation mechanisms, it is useful to start from a mechanical system that models the musculoskeletal properties of a standing person's body. For that purpose, I use the double compound inverted pendulum (DCIP), which is depicted in **Error! Reference source not found.**A. The DCIP has two parts, of which one represents the lower and the other the upper body. The DCIP has two joints, one at the ankle and one at the hip. The angular position of the lower and the upper body relative to gravity are denoted by, respectively, θ_1 and θ_2. The muscular forcing torque at the ankle and the hip joint are denoted by, respectively, z_1 and z_2. I will denote these angular positions and forcing torques as vectors $\boldsymbol{\theta} = [\theta_1, \theta_2]^t = [\theta_1; \theta_2]$ and $\boldsymbol{z} = [z_1, z_2]^t = [z_1; z_2]$. These variables are all functions of time, but I will suppress their argument t whenever the context allows.

The movements of the DCIP are fully specified by its EoM, which express the angular acceleration $\ddot{\boldsymbol{\theta}} = \left[\ddot{\theta}_1, \ddot{\theta}_2\right]^t$ as a function of angular position $\boldsymbol{\theta}$, angular velocity $\dot{\boldsymbol{\theta}} = \left[\dot{\theta}_1, \dot{\theta}_2\right]^t$, and forcing torque $\boldsymbol{z}$. EoM of multibody mechanical systems are usually derived using the method of Euler-Lagrange, which is straightforward but tedious. Here, I only give the result of this derivation, and I suppress a lot of detail:

$$M(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} = -C(\boldsymbol{\theta})\dot{\boldsymbol{\theta}} - K\boldsymbol{\theta} + F\sin(\boldsymbol{\theta}) + \boldsymbol{z} \qquad \text{Eq. 1}$$

The active dynamics of the DCIP depend on the input/forcing torque $\boldsymbol{z}$, and the passive dynamics depend on the matrices $M(\boldsymbol{\theta})$ (mass moment of inertia), $C(\boldsymbol{\theta})$ (damping matrix), K (stiffness matrix), and F (matrix specifying the gravitational torque). Eq. 1 is nonlinear because of $\sin(\boldsymbol{\theta})$, and because of the way the matrices $M(\boldsymbol{\theta})$ and $C(\boldsymbol{\theta})$ depend on $\boldsymbol{\theta}$ (via sines and cosines).

The matrices $M(\boldsymbol{\theta})$, $C(\boldsymbol{\theta})$, K, and F depend on parameters that determine, respectively, the system's mass moment of inertia, the joints' damping and stiffness, and the gravitational torque. The values of these parameters are crucial for the dynamic properties of the DCIP. For instance, the DCIP is only unstable if the stiffness of the joints is less than a gravity-dependent constant (the critical stiffness) that determines the gravitational torque per unit of lean angle. Fortunately, the values of the crucial parameters have been measured. For

instance, several studies have demonstrated that the ankle joint stiffness is less than the critical stiffness, and that this is due to the compliance of the Achilles tendon [34-37].

To express $\ddot{\boldsymbol{\theta}}$ as a function of $\boldsymbol{\theta}$, $\dot{\boldsymbol{\theta}}$ and $\boldsymbol{z}$, the left and right side of Eq. 1 must be premultiplied by $M(\boldsymbol{\theta})^{-1}$. Usually, $\boldsymbol{\theta}$ and $\dot{\boldsymbol{\theta}}$ are combined in the vector $\boldsymbol{x} = [\boldsymbol{\theta}; \dot{\boldsymbol{\theta}}]$, which is called the state of the mechanical system. The EoM can then be written as a first order nonlinear differential equation for the state $\boldsymbol{x}$, which I will denote as $\dot{\boldsymbol{x}} = \Phi(\boldsymbol{x}, \boldsymbol{z})$; note that $\dot{\boldsymbol{x}} = [\dot{\boldsymbol{\theta}}; \ddot{\boldsymbol{\theta}}]$. This differential equation is usually called the EoM in state space form and is described in more detail in the Methods section.

In the neighborhood of $\boldsymbol{x} = \boldsymbol{0}$, the nonlinear differential equation $\dot{\boldsymbol{x}} = \Phi(\boldsymbol{x}, \boldsymbol{z})$ can be approximated by a linear differential equation of which the coefficient matrices are obtained as the Jacobian of $\Phi(\boldsymbol{x}, \boldsymbol{z})$ and evaluated at $\boldsymbol{x} = \boldsymbol{0}$ and $\boldsymbol{z} = \boldsymbol{0}$. Note that $\ddot{\boldsymbol{\theta}}$ depends linearly on $\boldsymbol{z}$. This is also called a Taylor approximation of $\Phi(\boldsymbol{x}, \boldsymbol{z})$. This linear approximation is accurate for a standing person that is almost stationary (i.e., $\dot{\boldsymbol{\theta}} \approx \boldsymbol{0}$) and stays close to the upright position (i.e., $\boldsymbol{\theta} \approx \boldsymbol{0}$). It can be written as follows:

$$\dot{\boldsymbol{x}} = \Phi(\boldsymbol{x}, \boldsymbol{z}) \approx A\boldsymbol{x} + B\boldsymbol{z} \qquad \text{Eq. 2}$$

in which A and B are the Jacobian matrices of $\Phi(\boldsymbol{x}, \boldsymbol{z})$ with respect to, respectively, $\boldsymbol{x}$ and $\boldsymbol{z}$, and evaluated at $\boldsymbol{x} = \boldsymbol{0}$ and $\boldsymbol{z} = \boldsymbol{0}$.

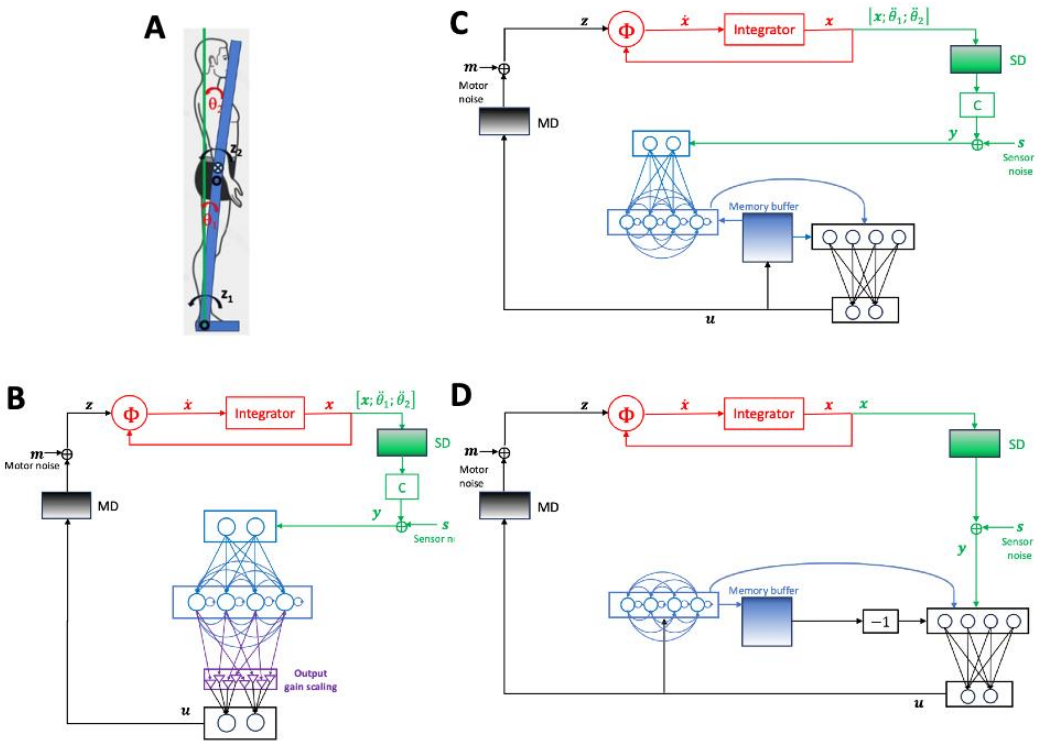

Fig. 1: Three SMD compensation mechanisms applied to standing balance control. (A) A standing body with a superimposed double compound inverted pendulum rotating about the ankle and the hip, used to model standing balance control. See text for an explanation of the symbols. (B) Block diagram of closed-loop feedback control with SMD compensation by means of gain scaling (see text). (C) Block diagram of closed-loop feedback control with SMD compensation by means of the convolution predictor (see text). (D) Block diagram of closed-loop feedback control with SMD compensation by means of the Smith predictor (see text).

Overview of the SMD compensation mechanisms

The three SMD compensation mechanisms are depicted in Fig. 1B-D. Fig. 1B shows a block diagram of closed-loop feedback control with SMD compensation by means of gain scaling. The diagram involves a mechanical (in red), a sensory (in green), a computational (blue, black and purple elements; see below), and a motor output system (in black) whose input is modulated by gain scaling (in purple). The computational system models the CNS as a neural network that (1) receives the sensory feedback $\boldsymbol{y}$ in its input level (green rectangle), (2) has an intermediate level with recurrent connections (blue rectangle), and (3) produces output $\boldsymbol{u}$ via two feedforward levels (purple and black rectangle). The black-and-white box depicts the

delay between output $\boldsymbol{u}$ and the mechanical system state $\boldsymbol{x}$ (motor delay, MD), and the green-and-white box depicts the delay between $\boldsymbol{x}$ and the sensory feedback $\boldsymbol{y}$ (sensory delay, SD). The SMD is equal to MD plus SD.

Fig. 1C shows a block diagram of closed-loop feedback control with SMD compensation by means of the convolution predictor. The main difference with the gain scaling in Fig. 1B is the role of the memory buffer of efference copies, which is depicted as a blue-and-white box. This memory buffer is used to store input for the computational and the motor output system.

Fig. 1D shows a block diagram of closed-loop feedback control with SMD compensation by means of the Smith predictor. The blue-and-white box now depicts a memory buffer of state estimates (produced by the recurrently connected level in blue) that stores input for the motor output system.

In Fig. 1B and 1C, the sensory system (in green) maps the vector $[\boldsymbol{x}; \ddot{\boldsymbol{\theta}}]$ onto sensory variables, adds noise $\boldsymbol{s}$ and feeds the resulting feedback $\boldsymbol{y}$ into the computational system. By including the angular acceleration $\ddot{\boldsymbol{\theta}}$ in the sensory feedback, I deviate from most of the existing literature on closed-loop feedback control, in which the sensory feedback only pertains to the state variable $\boldsymbol{x}$. This is because I use a sensory model for muscle spindle proprioceptive feedback, and sensory neurophysiology studies are well in line with the assumption that primary spindle afferents predominantly encode acceleration [38-41]. Angular acceleration sensory feedback can be modeled as incomplete state feedback and therefore requires a state observer [10].

The role of the sensory feedback is different for the Smith predictor and the two other mechanisms: the Smith predictor uses the sensory feedback to correct the state estimate such that it turns into a state prediction, whereas the other mechanisms use the sensory feedback to update the state estimate. To correct the state estimate in the Smith predictor, the sensory feedback must be full-state and noise-free (i.e., $\boldsymbol{y}(t) = \boldsymbol{x}(t - SD)$) and therefore Fig. 1D does not contain the matrix C that maps the state variables on the sensory feedback (or, equivalently, C is the identity matrix).

A computational system for the scenario without SMD

My interest is in how the computational system can control the mechanical system in the presence of a SMD. It is useful to first describe this computational system for the scenario without SMD. This computational system can be described both as a neural network and algebraically.

The neural network and the algebraic representation

The two representations are shown side-by-side in Fig. 2. From a neurobiological perspective, the computational system is a neural network with an input level, an intermediate level with recurrent connections, and an output level (see Fig. 2A). These recurrent connections agree with the fact that, in the CNS, neuronal groups are connected both in a feedforward (away from the sensory input) and a feedback (towards the sensory input) direction. As is common in neuroscience, I assume that the CNS learns optimal weights for the neural network's connections [28, 42, 43]. Calculating these optimal weights is a computational challenge and is often considered a part of machine learning. Here, I follow a different approach: I establish a correspondence between the neural network and an equivalent algebraic approach for which an optimal solution is known. I use an LQG optimality criterion that quantifies the objective to stay upright using control signals that stay within the torque limits of the mechanical system [9]. The LQG-optimal solution computes (1) an optimal internal state estimate $\hat{\boldsymbol{x}}$ by integrating a linear differential equation (the Kalman filter) that takes as input an efference copy $\boldsymbol{u}$ and sensory feedback $\boldsymbol{y}$, and (2) an optimal control action $\boldsymbol{u}$ by multiplying the state estimate $\hat{\boldsymbol{x}}$ by the LQR gain $-K$ (see Fig. 2B).

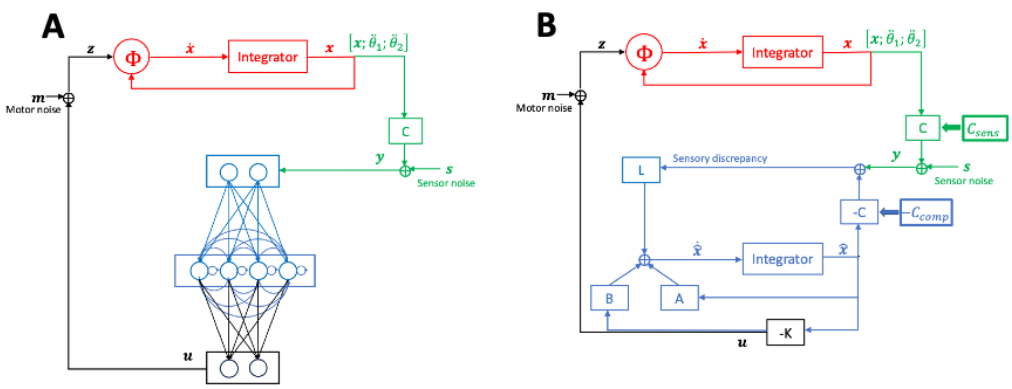

Fig. 2: Closed-loop feedback control model without SMD. (A) Neural network representation (see text). (B) Corresponding algebraic representation (see text).

The computational system is based on (1) a linear forward model $\dot{\boldsymbol{x}} = A\boldsymbol{x} + B\boldsymbol{z}$ that approximates the nonlinear mechanical system $\dot{\boldsymbol{x}} = \Phi(\boldsymbol{x}, \boldsymbol{z})$, and (2) a linear sensory model $\boldsymbol{y} = C_{Comp}\boldsymbol{x} + \boldsymbol{s}$ of the input to the computational system. I assume the matrices A, B and C_{Comp} to be known/learned and therefore can compute the LQG-optimal values for L (the Kalman gain) and -K (the LQR gain). It follows that the optimal recurrent weights are $A - BK - LC_{Comp}$. To prevent possible misunderstandings, the equivalence between the neural network in Fig. 2A and the algebraic formalism in Fig. 2B is only used to demonstrate its neural plausibility; no neural network implementation with nonlinear properties has been built.

The computational system receives input from a sensory system, and it implements an internal model for this system. In Fig. 2B, the sensory system is characterized by the matrix C_{Sens} and its internal model by the matrix C_{Comp}. I consider a sensory system for muscle spindle proprioceptive feedback, which is characterized by the following properties: (1) the firing rates of the muscle spindle primary afferents scale with intrafusal fiber acceleration [38-41], and (2) this intrafusal fiber acceleration is controlled by fusimotor neurons that cancel the reafferent acceleration component [10, 29-31] resulting in exafferent acceleration feedback. To avoid clutter, the fusimotor control of the sensory system is not included in Fig. 1 (panels B and C) and Fig. 2, but it is included in the corresponding Fig. 1 in [10]. The matrix C_{Sens} linearly combines total acceleration feedback (exafferent plus reafferent) and fusimotor input, which cancels the reafferent acceleration component, resulting in exafferent acceleration feedback [10].

The exafferent acceleration feedback of the DCIP is a nonlinear function of the state variables, but it can be approximated linearly at the unstable fixed point. The coefficient matrix of this linear approximation is the lower half of the matrix A in Eq. 2 and is denoted by C_{Comp}. The resulting linear sensory internal model is not full rank, and therefore the state can only be estimated using a state observer (see Eq. 1). This model is described in more detail in [10] and in the Methods (see *A mechanical model for standing balance constrains the relation between exafferent joint acceleration and the state variables*).

The LQG solution

The LQG solution for a computational system without SMD compensation is the following linear dynamical system:

$$\dot{\hat{\boldsymbol{x}}}(t) = A\hat{\boldsymbol{x}}(t) + B\boldsymbol{u}(t) + L\left[\boldsymbol{y}(t) - C_{Comp}\hat{\boldsymbol{x}}(t)\right] \qquad \text{Eq. 3}$$

$$\boldsymbol{u}(t) = -K[\hat{\boldsymbol{x}}(t) - \boldsymbol{r}] \qquad \text{Eq. 4}$$

Eq. 3 is a state equation and is also called a state observer; if L is the Kalman gain, it is called the Kalman filter. The state observer specifies the derivative of the state estimate as the sum of two components: (1) an expected state derivative $A\hat{\boldsymbol{x}}(t) + B\boldsymbol{u}(t)$, and (2) a feedback-based correction $L\left[\boldsymbol{y}(t) - C_{Comp}\hat{\boldsymbol{x}}(t)\right]$. The state observer is a mathematical specification of the powerful idea that perception is the result of a process that combines what the CNS already believes (the current state estimate $\hat{\boldsymbol{x}}(t)$) with sensory information $\boldsymbol{y}(t)$ and a memory of its control signal $\boldsymbol{u}(t)$.

Eq. 4 is an output equation, and it computes a control signal $\boldsymbol{u}(t)$ by scaling the difference vector $[\hat{\boldsymbol{x}}(t) - \boldsymbol{r}]$, in which $\boldsymbol{r}$ is the target state that specifies the movement objective. For balance control, $\boldsymbol{r}$ is the angle relative to gravity of the line that joins the center of gravity (CoG) with the area of support. People prefer a slightly leaned forward position of approximately 0.0156 radians [4], and therefore $\boldsymbol{r} = [0.0156, 0.0156, 0, 0]^t$ is a plausible choice. For simplicity, the target state $\boldsymbol{r}$ was omitted from Fig. 1 and Fig. 2. In engineering terms, the combined state and output equation are a filter of the sensory feedback $\boldsymbol{y}(t)$ and the target state $\boldsymbol{r}$ that produces the control signal $\boldsymbol{u}(t)$ as its output.

The state observer in Eq. 3 is not needed if the sensory feedback is full-state and noise-free (i.e., $\boldsymbol{y}(t) = \boldsymbol{x}(t)$). In that case, in the output equation Eq. 4, $\hat{\boldsymbol{x}}$ is replaced by $\boldsymbol{x}$. If the sensory feedback is full-state but noisy, the state observer (with C_{Comp} equal to the identity matrix) is used to optimally combine the noisy feedback with the forward model.

Expected versus predicted states

At this point, it is useful to make a sidestep to explain the difference between expected and predicted states. I do this because the expected state derivative $A\hat{\boldsymbol{x}}(t) + B\boldsymbol{u}(t)$ corresponds to a term in the discrete time state observer that has also been called a prediction [44-46], whereas in this paper, I use prediction in a different meaning. The discrete time state observer involves the following two-step operation:

$$\begin{aligned} \hat{\boldsymbol{x}}_{k+1}^{exp} &= A\hat{\boldsymbol{x}}_k + B\boldsymbol{u}_k \\ \hat{\boldsymbol{x}}_{k+1} &= \hat{\boldsymbol{x}}_{k+1}^{exp} + L\left(\boldsymbol{y}_k - C\hat{\boldsymbol{x}}_{k+1}^{exp}\right) \end{aligned} \qquad \text{Eq. 5}$$

The time between consecutive integer-valued time points k and $k+1$ is δ, and the relation between the continuous and the discrete time axis is given by $t = k\delta$. I use $\hat{\boldsymbol{x}}_k$, $\boldsymbol{u}_k$ and $\boldsymbol{y}_k$ as a shorthand for $\hat{\boldsymbol{x}}(k\delta)$, $\boldsymbol{u}(k\delta)$ and $\boldsymbol{y}(k\delta)$. For notational simplicity, I have reused the symbols A, B, L and C of the continuous time state observer in Eq. 3, but the first three refer to different matrices than the corresponding ones for the continuous time observer. I will return to this when describing Eq. 15.

The relevant term in Eq. 5 is $\hat{\boldsymbol{x}}_{k+1}^{exp}$, which I will denote as the expected state, but in other papers it is also denoted as the predicted state or the a priori state estimate. The expected state $\hat{\boldsymbol{x}}_{k+1}^{exp}$ predicts the state estimate at time $(k+1)\delta$ from the state estimate at time $k\delta$ (i.e., $\hat{\boldsymbol{x}}_k$) and the efference copy $\boldsymbol{u}_k$. Crucially, this is not a prediction over a neurobiologically meaningful interval, but over a discretization interval of length δ that is chosen small enough to study the properties of the continuous state estimate $\hat{\boldsymbol{x}}(t)$ in discrete time. For prediction over a neurobiologically meaningful interval, I use $d = SMD/\delta$ to denote the SMD length as an integer number of intervals of length δ. In a later section (*The convolution predictor*), I will motivate a prediction of the state at time $(k+d)\delta = k + SMD$ from a state estimate at time $k\delta$ and the efference copies $\boldsymbol{u}_k, \boldsymbol{u}_{k+1}, \ldots, \boldsymbol{u}_{k+d}$. This is very different from the state observer, which combines sensory feedback and the most recent efference copy, but does not meaningfully predict beyond the time of the sensory feedback; its discrete time version only predicts over an arbitrary discretization interval.

Gain scaling

Gain scaling is depicted in Fig. 1B. Gain scaling modifies the output equation Eq. 4. The output of this equation scales with the gain $-K$, and the larger its magnitude $|K|$, the more aggressive the corrections that are intended to bring the mechanical state to its target. With increasing SMD, it becomes more likely that corrections destabilize the mechanical system. This is because the stability of a closed loop system depends on the gain of the open loop system (the mechanical and the sensory system, followed by the observer and the controller $-K$) at a critical frequency, the frequency for which the open loop transfer function flips the phase by 180 degrees. To achieve stability, at this critical frequency, the amplitude of the open loop output gain must be less than 1. Crucially, this critical frequency decreases with the SMD and brings it in a range where the open loop output gain typically is higher. Now, by downscaling $|K|$, the open loop output gain decreases, ideally below 1 at the critical frequency.

For a mechanical system, the output gain $-K$ is a matrix and, in principle, every matrix element could be scaled separately. This is also the scaling scenario depicted in Fig. 1B, and optimal scaling in this scenario would thus be equivalent to knowing the optimal output gain for a nonzero SMD. Because there is no theory about optimal output gains for a nonzero SMD, this type of optimal scaling is computationally infeasible. In this paper, I only consider optimal global downscaling of the output gain: replace $-K$ (the LQR gain for SMD=0) by $-\alpha K$, in which α optimizes the LQR objective function.

The convolution predictor

The convolution predictor computes a control signal by applying an output gain to a prediction of the future state $\boldsymbol{x}(t + SMD)$. In its simplest form, the convolution predictor is obtained as the solution of a delayed-input forward model, and this is also how it was introduced under the label finite spectrum assignment (FSA) [11-17]. Here, I take the same starting point but deviate from the existing literature by estimating the initial condition of the solution by a state observer that is based on the delayed-input forward model. This

generalizes the method to situations with noisy and incomplete sensory feedback, and it does so optimally if the state observer uses the Kalman gain.

Solving the delayed-input forward model

The delayed-input forward model describes the dynamics of the mechanical system as follows:

$$\dot{\boldsymbol{x}}(t) = A\boldsymbol{x}(t) + B\boldsymbol{z}(t - SMD) \qquad \text{Eq. 6}$$

In this forward model, there is a delay of SMD seconds between the noisy control signal $\boldsymbol{z}$ ($= \boldsymbol{u} + \boldsymbol{m}$) and the time at which it arrives in the mechanical system. This delay reflects the fact that the control signal is based on sensory feedback that is generated SMD seconds before its arrival in the mechanical system.

The future state $\boldsymbol{x}(t + SMD)$ is obtained by evaluating the solution of Eq. 6 at time $t + SMD$. This solution can be found in the specialized literature [13, 17] or derived from more general results in a textbook [47]:

$$\boldsymbol{x}(t + SMD) = e^{A \times SMD}\boldsymbol{x}(t) + \int_{-SMD}^{0} e^{-A\tau} B\boldsymbol{z}(t + \tau)d\tau \qquad \text{Eq. 7}$$

in which the multiplication sign ($\times$) is only added for readability. The ingredients for this solution are the initial condition $\boldsymbol{x}(t)$ and the noisy control signals $\boldsymbol{z}(t + \tau)$ for $-SMD < \tau < 0$. From Eq. 7, one can obtain an estimate of the future state $\boldsymbol{x}(t + SMD)$ by replacing the initial condition and the noisy control signals by estimates. This is straightforward for the noisy control signals, which can be replaced by the efference copies $\boldsymbol{u}(t + \tau)$. Crucially, this operation involves the assumption that the CNS has access to a memory buffer that stores the efference copies for the most recent interval of SMD seconds.

The initial condition can be estimated from the sensory feedback $\boldsymbol{y}(t)$, and this will be described in the next section, *The delayed-input state observer.* There, it will be described how, from the sensory feedback $\boldsymbol{y}(t)$ one can obtain an estimate of $\boldsymbol{x}(t - SD)$, the state of the mechanical system at the time the sensory feedback was generated. With the estimate $\hat{\boldsymbol{x}}(t - SD)$ as the initial condition, application of Eq. 7 results in an estimate of $\boldsymbol{x}(t + MD)$, the state of the mechanical system SMD seconds after the sensory feedback was generated. I will denote this estimate by $\hat{\boldsymbol{x}}^{SMD}(t)$:

$$\hat{\boldsymbol{x}}^{SMD}(t) = e^{A \times SMD}\hat{\boldsymbol{x}}(t - SD) + \int_{-SMD}^{0} e^{-A\tau} B\boldsymbol{u}(t + \tau)d\tau \qquad \text{Eq. 8}$$

A potentially confusing aspect of this equation is that it combines quantities that are expressed on different times axes: $\hat{\boldsymbol{x}}(t - SD)$ estimates a state at time $t - SD$ on the time axis of the mechanical system, but this is computed at time t on the time axis of the computational system (CNS), which is also used to express the efference copies $\boldsymbol{u}(t + \tau)$. In other words, at time t on the time axis of the computational system, a prediction $\hat{\boldsymbol{x}}^{SMD}(t)$ is computed that estimates $\boldsymbol{x}(t + MD)$ on the time axis of the mechanical system, and for that it uses efference copies $\boldsymbol{u}(t + \tau)$ on the computational system's time axis.

Note that the convolution predictor does not have to know the separate sensory and motor delay (SD, resp., MD), but only their sum SMD. This is because it uses an initial condition $\hat{\boldsymbol{x}}(t - SD)$ of which it doesn't have to know the implied time axis; it only must know the total time span of the memory buffer of efference copies.

The delayed-input state observer

The delayed-input state observer computes the initial condition $\hat{\boldsymbol{x}}(t - SD)$ that is used in the convolution predictor in Eq. 8. This state observer is based on the delayed-input forward model in Eq. 6, and is written as follows:

$$\dot{\hat{\boldsymbol{x}}}(t) = A\hat{\boldsymbol{x}}(t) + B\boldsymbol{u}(t - SMD) + L[\boldsymbol{y}(t) - C\hat{\boldsymbol{x}}(t)] \qquad \text{Eq. 9}$$

The difference with the state observer in Eq. 3 is the expected state derivative $A\hat{\boldsymbol{x}}(t) + B\boldsymbol{u}(t - SMD)$, which is taken from the delayed-input forward model instead of the forward model with non-delayed input. If the matrix L in Eq. 9 is the Kalman gain, then the delayed-input state observer produces an optimal estimate of $\boldsymbol{x}(t - SD)$, namely the estimate with the smallest mean-squared error. This follows from the following facts: (1) the sensory input $\boldsymbol{y}(t)$ is a noisy version of $C\boldsymbol{x}(t - SD)$, and (2) at the time the sensory input is generated, the systematic (noise-free) part of the mechanical system input is $\boldsymbol{u}(t - SMD)$. This may be confusing because $\boldsymbol{x}(t - SD)$ and $\boldsymbol{u}(t - SMD)$ are expressed on different times axes: $\boldsymbol{x}(t - SD)$ is expressed on the time axis of the mechanical system and $\boldsymbol{u}(t - SMD)$ on the time axis of the computational system, where control signals are generated MD seconds before they arrive in the mechanical system.

The delayed-input state observer requires that the CNS has access to a memory buffer that stores efference copies $\boldsymbol{u}(t)$ of the control signals that were produced in the most recent SMD interval (i.e., between $t -$ SMD and t seconds). It uses the oldest efference copy $\boldsymbol{u}(t - SMD)$ from this buffer to evaluate the state observer in Eq. 9.

Approximate convolution

As will be explained in the following, it is likely that the convolution operation in Eq. 7 is a computational challenge for the CNS. Therefore, it is useful to investigate whether approximate convolution could be as effective for SMD compensation as the full-blown convolution operation. To introduce different approximate convolution schemes, it is useful to start from the discrete time version of Eq. 8. In the derivation of this discrete time version, I assume that $\boldsymbol{u}(t)$ is a piecewise constant function, with piece length equal to δ sec., the defining time step for the discrete time version. The discrete time quantities that correspond to the predictions $\widehat{\boldsymbol{x}}^{SMD}(t)$, the state estimates $\widehat{\boldsymbol{x}}(t - SD)$, and the efference copies $\boldsymbol{u}(t)$ are denoted by, resp., $\widehat{\boldsymbol{x}}_k^{SMD}$, $\widehat{\boldsymbol{x}}_k$ and $\boldsymbol{u}_k$, with index $k = t/\delta$. (The index k under the state estimate $\widehat{\boldsymbol{x}}_k$ denotes the time at which the estimate is computed, and not the time at which the corresponding sensory feedback was generated.) Under the assumption of a piecewise constant $\boldsymbol{u}(t)$, the discrete time approximation of Eq. 8 can be written as follows[1]:

$$\widehat{\boldsymbol{x}}_k^{SMD} = A^{SMD}\,\widehat{\boldsymbol{x}}_k + \sum_{i=1}^{d} B^{i,\delta}\boldsymbol{u}_{k-i} \qquad \text{Eq. 10}$$

In this equation, A^{SMD} denotes the matrix exponential $e^{A\times SMD}$, d equals SMD/δ, and the efference copy weight matrices $B^{i,\delta}$ denote an integral that has an analytical solution:

$$B^{i,\delta} = \left[\int_{-i\delta}^{-(i-1)\delta} e^{-A\tau}\,d\tau\right]B = A^{-1}\left(e^{Ai\delta} - e^{A(i-1)\delta}\right)B \qquad \text{Eq. 11}$$

The convolution in the second term on the right of Eq. 10 is computationally heavy because it does not allow to reuse parts of previous computations. It is unknown whether the CNS has the required neural circuitry to successfully complete this job. In addition, this term involves

[1] The stability of this discrete time approximation in the context of closed loop control has been a topic of debate, and it has been shown that a low-pass filter of the predictions $\widehat{\boldsymbol{x}}_k^{SMD}$ effectively deals with stability problems [14,17]. This debate was in the context of full-state feedback and therefore did not involve a state observer. In my simulations, I did not encounter stability problems and therefore do not consider low-pass filtering.

a set of efference copy weight matrices $B^{i,\delta}$ that must be learned from experience with the mechanical system. Importantly, for an unstable mechanical system, the oldest efference copies are the most important ones, because the weights $A^{-1}\left(e^{Ai\delta} - e^{A(i-1)\delta}\right)B$ increase exponentially with the buffer time index i. This exponential increase follows from the fact that an unstable system has a system matrix A with at least one positive eigenvalue. Thus, the largest weights are the most difficult to learn because they depend on the oldest memories, and these are likely to be the least accurate.

To investigate the consequences of different computational limitations of the CNS, I consider four predictors of different levels of complexity: (1) from passive dynamics only (i.e., using only the first term on the right side of Eq. 11), (2) under the assumption that the efference copy remains constant at the most recent value (i.e., using $\boldsymbol{u}_{k-1}$ with a single weight $\sum_{i=1}^{d} B^{i,\delta}$), (3) using the same average weight $\sum_{i=1}^{d} B^{i,\delta}/d$ for all efference copies, and (4) using the full-blown mechanically motivated Eq. 10. The first two predictions do not make use of a memory buffer and therefore can be considered as extrapolations based on, respectively, only the current state estimate or the current state estimate plus the current control signal. The third predictor is a simple boxcar filter but requires a full memory buffer.

The Smith predictor

Just like the convolution predictor, the Smith predictor computes a control signal by applying the output gain to a prediction of a future state. The Smith predictor assumes that the sensory feedback $\boldsymbol{y}(t)$ is full-state and noise-free: $\boldsymbol{y}(t) = \boldsymbol{x}(t - SMD)$. Now, if the SMD were 0, then $\boldsymbol{y}(t) = \boldsymbol{x}(t)$ and the control signal would be obtained by multiplying $\boldsymbol{y}(t)$ by the output gain: $\boldsymbol{u}(t) = -K\boldsymbol{y}(t)$. (For simplicity, I ignore the target state $\boldsymbol{r}$; see Eq. 4.) An easy way to introduce the Smith predictor is as a correction of the sensory feedback $\boldsymbol{y}(t)$ for the unwanted component that is due to a nonzero SMD. This unwanted component is $\boldsymbol{x}(t - SMD) - \boldsymbol{x}(t)$, the actual minus the desired sensory feedback.

The Smith predictor estimates the unwanted component using the forward model with non-delayed and noise-free input:

$$\dot{\boldsymbol{x}}(t) = A\boldsymbol{x}(t) + B\boldsymbol{u}(t) \qquad \text{Eq. 12}$$

This estimate is the difference $\hat{\boldsymbol{x}}(t - SMD) - \hat{\boldsymbol{x}}(t)$, in which $\hat{\boldsymbol{x}}$ denotes a state estimate that is obtained from the solution of the forward model in Eq. 12. Subtracting this correction term from $\boldsymbol{y}(t)$, the Smith predictor becomes $[\boldsymbol{y}(t) + \hat{\boldsymbol{x}}(t) - \hat{\boldsymbol{x}}(t - SMD)]$, which is then multiplied by the output gain to obtain the control signal. An alternative way to introduce the Smith predictor is to consider $\hat{\boldsymbol{x}}(t)$ as an estimate of the current state and $\hat{\boldsymbol{x}}(t - SMD)$ as a correction term for $\boldsymbol{y}(t)$.

The solution of the forward model with non-delayed input (see Eq. 12) has a similar structure as the solution of the forward model with delayed input that was used for the convolution predictor (see Eq. 7):

$$\hat{\boldsymbol{x}}(v) = e^{Av}\hat{\boldsymbol{x}}(0) + \int_0^v e^{A(v-\tau)} B\boldsymbol{u}(\tau)d\tau \qquad \text{Eq. 13}$$

in which v can be t or $t - SMD$. There are several important differences with the convolution predictor, and these are best highlighted using the discrete-time version of Eq. 13. As for the convolution predictor, I assume that $\boldsymbol{u}(t)$ is a piecewise constant function, with piece length equal to δ sec. The discrete time version of Eq. 13 can then be written as follows:

$$\hat{\boldsymbol{x}}_h = e^{Ah\delta}\hat{\boldsymbol{x}}_0 + \sum_{i=0}^{h-1} B^{(h-i),\delta}\boldsymbol{u}_i \qquad \text{Eq. 14}$$

in which h can be k or $k - d$, and $B^{(h-i),\delta}$ is defined in Eq. 11. The first difference with the convolution predictor in Eq. 10 is that the initial condition $\hat{\boldsymbol{x}}_0$ in Eq. 14 must be set once at the start of the time series and independently of the sensory feedback. The second difference is that the summation in Eq. 14 is not a convolution but an ordinary sum with number of terms increasing with h. This allows for a computationally efficient implementation based on this recursion:

$$\hat{\boldsymbol{x}}_h = e^{A\delta}\hat{\boldsymbol{x}}_{h-1} + B^{1,\delta}\boldsymbol{u}_{h-1} \qquad \text{Eq. 15}$$

This recursion can be derived by inserting the definition of $B^{(h-i),\delta}$ (see Eq. 11) in Eq. 14. Eq. 15 is the discrete time forward model with non-delayed input and is identical to the first step in Eq. 5, but now with the formally correct system matrix $e^{A\delta}$ and input matrix $B^{1,\delta}$; for simplicity, these matrices were denoted by A and B in Eq. 5. And the third difference is that a computationally efficient implementation of the Smith predictor requires a memory buffer of state estimates $\hat{\boldsymbol{x}}_h$ instead of efference copies $\boldsymbol{u}_h$. To compute the correction term in the discrete time Smith predictor $[\boldsymbol{y}_k + \hat{\boldsymbol{x}}_k - \hat{\boldsymbol{x}}_{k-d}]$, this memory buffer must contain d elements, just like the memory buffer for the convolution predictor.

In sum, the Smith predictor estimates a future state by correcting full-state delayed sensory feedback. This correction is based on the solution of a forward model with non-delayed feedback, and it can be implemented in a computationally efficient way using a memory buffer of state estimates.

The Smith predictor as a model for CNS computations

In motor control, the Smith predictor is far more popular than the convolution predictor, which is mostly discussed in the control theory literature [for an exception, see 16]. A probable reason for this is that the Smith predictor has played such an important role in the popularization of the concept of a forward model [19, 24, 25, 48]. However, the Smith predictor is not a useful model for CNS computations and this is for several reasons: (1) it cannot control an unstable mechanical system, (2) it ignores the fact that a plausible forward model must have a delayed input, whose learning and use requires a memory buffer of efference copies, and (3) it cannot be used in combination with a state observer.

First, the control theory literature has demonstrated that, under conditions that make it an interesting model for the CNS, the Smith predictor cannot control an unstable mechanical system [11, 20, 49]. The conditions that make the Smith predictor an interesting model for the CNS pertain to how well the forward model approximates the mechanical system. Specifically, to be an interesting model for the CNS, the Smith predictor's performance (1) must be robust to some degree of approximation, and (2) must improve as the approximation becomes better. Although the control theory literature is clear in this respect, its results must be formulated carefully: there are examples of an unstable system that is controlled by the Smith predictor, but these do not involve a forward model that approximates the mechanical system. Instead, they involve a stable forward model that is totally unrelated to the unstable mechanical system [20].

Second, the Smith predictor ignores the fact that a plausible forward model must have a delayed input. The CNS can learn a forward and a sensory model only by associating motor signals (the input to the mechanical system) with sensory feedback (the signal that informs CNS about the mechanical system's output). Because of the SMD, this learning-by-

association is only possible if the CNS learns a forward model with delayed input, and this in turn requires that the CNS has access to a memory buffer of efference copies.

Third, when used as a model for CNS computations, the Smith predictor cannot be used in combination with a state observer. From an engineering point of view, it is straightforward to replace the sensory feedback $\boldsymbol{y}(t)$ by the output of the delayed-input state observer in Eq. 9, but from the point of view of its possible neurobiological implementation, such a hybrid mechanism would be implausible. This is for two reasons. First, it would require that the CNS implements both the forward model with delayed input (for the delayed-input state observer) and the forward model with non-delayed input (for the correction term). Second, it would require that the CNS implements two memory buffers: one memory buffer that stores efference copies (for the delayed-input state observer), and another one that stores state estimates (for the correction term).

Evaluating the performance of the SMD compensation mechanisms using simulation

I now give a general description of a set of the simulation studies with which I have evaluated the performance of the different SMD compensation mechanisms; the details are given in the Methods. The parameters of the two mechanical systems (a standing person and a rider-bicycle combination) were assigned realistic values, most of which were based on measurements (see Methods).

The model for the standing person (the DCIP) is nonlinear, and the one for the rider-bicycle combination is linear. Because the SMD compensation mechanisms depend on linear internal models, control of the nonlinear standing person system requires internal models that only approximate this mechanical system. In the simulations, these internal models were obtained as linear approximations of the mechanical system's nonlinear EoM. In the control of the linear rider-bicycle system the internal models were identical to this system's linear EoM.

I ran simulations using two sensory systems: (1) a neurophysiologically plausible sensory system that produces noisy exafferent acceleration feedback, and (2) a neurophysiologically implausible sensory system that produces noisy full-state feedback. In my report, I focus on the simulations with noisy exafferent acceleration feedback. If these simulations reveal an

interesting pattern, I investigate their generality by checking whether this pattern is also present in simulations with noisy full-state feedback.

The time-varying input to the closed-loop control system involves sensor, motor and fusimotor noise. The noise amplitudes were set such that, without SMD, the root-mean-square deviation (RMSD) of the simulated CoG lean angle was equal to the RMSD of empirically observed CoG lean angles. The empirically observed RMSD for standing balance control (0.25 degrees) was taken from Figure 6 in [50], and the one for bicycle balance control (1.3321 degrees) was computed from a pilot dataset obtained from a study using a bicycle simulator [51]. The amplitude of the sensor noise was set equal to the amplitude of the combined sensory consequences of the motor and fusimotor noise. The motor and the fusimotor noise had identical amplitudes and their shared variance was 50%. These noise amplitudes were set separately for the exafferent acceleration and the full-state feedback; they were more than an order of magnitude larger for the full-state feedback.

The LQR gain $-K$ minimizes an objective function that quantifies a double objective with a precision and an energetic component: stay close to the target position (precision) with as little effort as possible (energetic). I computed $-K$ using parameters that weight the contribution of the precision and the energetic component equally, as quantified by realistic maximum values for the states and the joint torques. The Kalman gain L depends on the motor, fusimotor and sensor noise amplitudes, and I used the same values as for the noise signals that were fed into the mechanical and the sensory system.

Gain scaling starts from the LQR-optimal gain matrix $-K$ for a non-delayed system. For every SMD, I computed the scalar scaling parameter α that minimizes the LQR objective function for a delayed system controlled using the output gain $-\alpha K$. Depending on the SMD, it could happen that no α parameter value could be found that controls the mechanical system (i.e., the simulated standing person or rider-bicycle combination fell on the ground). In that case, it was concluded that gain scaling could not control the mechanical system for this SMD and larger ones.

The convolution predictor $\hat{\boldsymbol{x}}_k^{SMD}$ (see Eq. 10), was implemented in four different versions: (1) extrapolation based on the state estimate (passive dynamics only; without the second term on

the right of Eq. 10), (2) extrapolation based on the state estimate plus the most recent efference copy ($\boldsymbol{u}_{k-i} = \boldsymbol{u}_{k-1}$), (3) using the average efference copy weight $\sum_{i=1}^{d} B^{i,\delta}/d$ instead of $B^{i,\delta}$ (i.e., a boxcar filter), and (4) exactly according to Eq. 10.

The evaluation of the SMD compensation mechanisms is based on the largest SMD for which the RMSD CoG lean angle stays below an empirically determined maximum. For standing balance control, this maximum (1.5 degrees) was determined as the maximum RMSD that people tolerate and was taken from Fig. 6 (upper right panel) in [50]. For bicycle balance control, this maximum (15.11 degrees) corresponds to a turn radius that most people are not willing to take at the simulated speed (15.5 km/h) [10].

Standing balance control

Fig. 3 shows the standing balance simulation results for four of the six SMD compensation mechanisms. At large SMDs, the simulated angular position may exceed 90 degrees, indicating that the person falls on the floor. Simulated falls happened for all SMD compensation mechanisms but, depending on the mechanism, this started at a different SMD. If a simulated fall happened, no results are shown for the corresponding SMDs in Fig. 3.

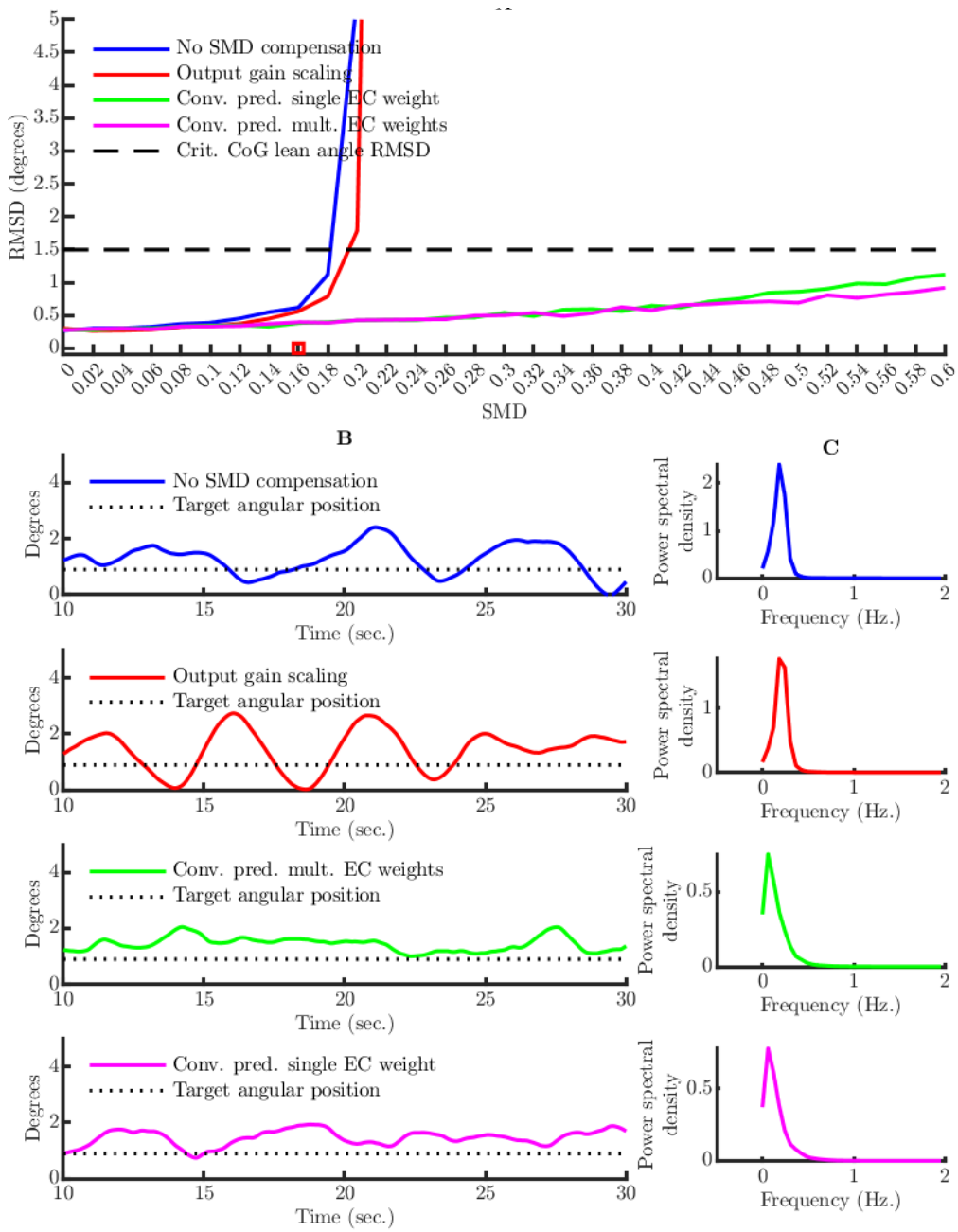

Fig. 3: Standing balance simulation study results. (A) Average RMSD CoG lean angle for four SMD compensation mechanisms (including no SMD compensation), as a function of SMD. The horizontal dashed line indicates the maximum angular position RMSD that people tolerate and is based on empirical work [50]. (B) Example angular position time series for the SMD that is indicated by a red box in panel A (SMD=0.16 sec.). (C) Power spectral density of the angular position for the same SMD as in B.

Without a SMD compensation mechanism, the critical CoG lean angle RMSD was exceeded for SMD>0.181 sec. (see Fig. 3A). This SMD is slightly above the range of the

neurobiological SMDs that have been reported in the literature: 0.1-0.16 sec. in [4, 52] and 0.14 sec. in [23]. As expected, output gain scaling produced smaller CoG lean angle RMSDs for all SMDs, and the critical RMSD was exceeded for a slightly larger SMD of 0.195 sec. (see Fig. 3A). The acceptable performance under no SMD compensation and output gain scaling depends on joint stiffness and damping (both at the ankle and the hip): lowering stiffness and damping below the measured values lowered the SMD that could be tolerated without exceeding the critical RMSD. When ankle joint stiffness and damping were set to zero, the maximum SMD that could be tolerated was only 0.05 sec.

The performance of the convolution predictor strongly depends on whether a full memory buffer of efference copies was used. The performance of extrapolation was so poor that the lines are not even visible in Fig. 3A: for passive dynamics only, the critical RMSD was exceeded for SMD>0.022 sec., and if also the most recent efference copy was used, the critical RMSD was exceeded for SMD>0.013 sec.

The versions of the convolution predictor that make use of a full memory buffer of efference copies have an excellent performance: both versions (with single and multiple efference copy weights) exceed the critical CoG lean angle RMSD only for a SMD larger than 0.6 sec. (see Fig. 3A). The precise maximum values are not shown in Fig. 3A; they are equal to 0.71 and 0.91 for, respectively, the single and the multiple weights version. It is useful to compare these values with the results of an empirical study in which the SMD was artificially increased by means of a robotic device [4]. In this study, humans could successfully compensate for an artificial SMD of 0.56 sec., well below the performance limit of a convolution predictor that makes use of a memory buffer of efference copies.

The performance of the convolution predictor depends on joint stiffness and damping: lowering stiffness and damping below the measured values lowered the SMD that could be tolerated without exceeding the critical RMSD. However, this effect was much less extreme than for the scenarios without SMD compensation and with output gain scaling. When ankle joint stiffness and damping were set to zero, the maximum SMDs that could be tolerated were 0.52 and 0.65 sec. for, respectively, the version with a single and multiple efference copy weights.

Two important results are (1) that the convolution predictor is only effective if it makes use of a full memory buffer of efference copies to predict a future state, and (2) that there is little difference between a computationally hard convolution using multiple efference copy weights and a computationally easy convolution using only a single weight. These results (necessity of a full memory buffer and sufficiency of a single weight) are not specific for noisy incomplete exafferent acceleration feedback; they were also obtained with noisy full-state feedback. Using the latter feedback, for extrapolation based on passive dynamics only and for extrapolation using the most recent efference copy, the critical RMSD was exceeded for, respectively, SMD>0.035 sec. and SMD>0.012 sec. And for the single and the multiple weights version of the convolution predictor, the critical RMSD was exceeded for, respectively, SMD>0.61 sec. and SMD>0.80 sec. Keep in mind that the noise level was set higher for full-state than for exafferent acceleration feedback to achieve an identical RMSD of 0.25 degrees at zero SMD for both feedback types.

I also confirmed the mathematical results for control performance of the Smith predictor when applied to an unstable mechanical system: even for the smallest SMD (0.001 sec.), the Smith predictor produced control signals that resulted in diverging CoG lean angles.

To give a more concrete picture of the simulated kinematics, in Fig. 3B, I show examples of CoG lean angle time courses for the effective SMD compensation mechanisms. Note the rhythmic pattern around 0.2 Hz for the scenario's without SMD compensation and with output gain scaling. Also note that the average CoG lean angle exceeds its target value, which is due to the positive gravitational torque that acts on the body when it is in a slightly leaned forward position.

I computed the power spectral densities of the CoG lean angle time courses after removing the mean CoG lean angle. These power spectral densities all show the drop-off between 0 and 2 Hz that is observed in empirical CoG lean angle power spectral densities (see Fig. 3C) [53].

Bicycle balance control

A mechanical model for bicycle balance control

There are two important differences between the physics of standing and bicycle balance control: (1) a bicycle's area of support (AoS) is a line instead of a surface, and (2) balance control of a moving bicycle involves not only the gravitational but also the centrifugal force. A stationary bicycle is balanced when the combined CoG of rider and bicycle is above the line that connects the contact points of the two wheels with the road surface, the so-called line of support (LoS). Because of disturbances, this CoG cannot be exactly above this one-dimensional LoS for some time. Therefore, a bicycle is considered balanced if the CoG fluctuates around the LoS within a limited range, small enough to prevent the bicycle from skidding and/or touching the road surface.

On a moving bicycle, not only gravity, but also the centrifugal force acts on the CoG. Crucially, the centrifugal force is under the rider's control via the turn radius [54]. The balance of a moving bicycle depends on the resultant of the gravitational and the centrifugal force: a bicycle is balanced if the direction of this resultant force fluctuates around the LoS without diverging. This dependence on the resultant force is also observed in walking, running, and skiing. Besides the forces that act on the CoG, there are also forces that turn the bicycle's front frame, and some of these forces are independent of the rider's action [55]. These rider-independent forces are responsible for the bicycle's self-stability within a limited speed range [55].

For the present study, I use the bicycle model in Fig. 4. This model consists of three rigid bodies: front frame, rear frame (which includes the rider's lower body and will also be denoted as the lower body), and the rider's upper body. The positions of these three bodies are specified by three angular variables: steering (δ), lower body (θ_1), and upper body (θ_2) angular position. The lower and upper body angular positions are relative to gravity.

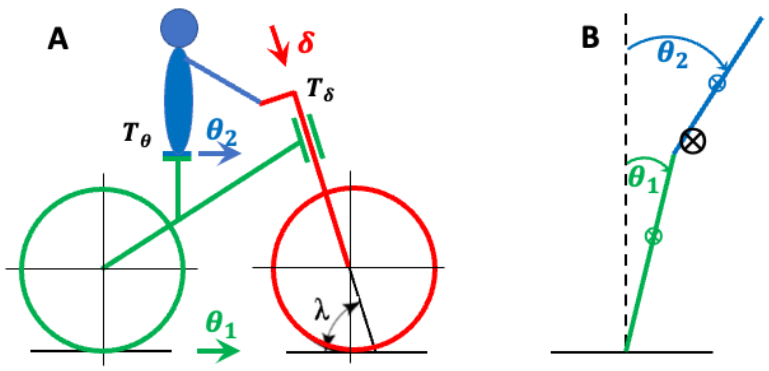

Fig 4: Kinematic variables of the bicycle model plus the rider-controlled forcing torques. (A) Side view. In green, the bicycle rear frame, characterized by its angular position θ_1 over the roll axis (green arrow). In red, the bicycle front frame, characterized by its angular position δ over the steering axis (red arrow). In blue, the rider's upper body, characterized by its angular position θ_2 over the roll axis (blue arrow). In black, (1) the steering torque T_δ and the hip torque T_θ, which are both applied by the rider, and (2) the steering axis angle λ. (B) Rear view. In green, the bicycle rear frame (including the lower body) angular position θ_1. In blue, the rider's upper body angular position θ_2. The symbol ⊗ denotes the CoG of the lower body (in green), the upper body (in blue), and the combined CoG (in black).

Cycling involves a double balance problem, and until now I have only described the first problem, keeping the combined CoG of rider and bicycle above the LoS. The second balance problem pertains to the rider's upper body only, and it involves keeping the upper body CoG above its own AoS, the saddle. I only consider balance over the roll axis (coinciding with the LoS), which corresponds to upper body movements to the left and the right. I thus ignore balance over the pitch axis (perpendicular to the LoS and gravity), which corresponds to upper body movements to the front and the back, typically caused by forward accelerations and braking.

For both balance problems (with respect to the combined and the upper body CoG), the relevant control action must produce a torque over the roll axis. Within the constraints of our kinematic model, there are two possible control actions: (1) turning the handlebars (using steering torque T_δ), and (2) leaning the upper body (using hip torque T_θ). At this point, it is convenient to make use of Fig. 4B, which shows the geometry of a DCIP but with different forces acting on it. By turning the handlebars, the contact point of the front tire (represented by the green rod) with the road surface moves to the left or the right, and this changes the

position of the combined CoG relative to the LoS. In the bicycle reference frame (in which the LoS is one of the axes) this corresponds to a centrifugal torque in the direction opposite to the turn (a tipping out torque). Steering in the direction of the lean produces a tipping out torque that brings the combined CoG over the LoS. This is called steering to the lean/fall.

The second control action is leaning the upper body, which can bring the upper body CoG above the saddle in a direct way. This deals with the second (upper body specific) balance problem. However, leaning the upper body cannot deal with the first balance problem (bringing the combined CoG above the LoS) in a direct way (i.e., without turning the handlebars). A crucial argument in favor of this claim is that a bicycle with a locked steer cannot be balanced; not a single case has been reported. As an aside, it must be noted that leaning the upper body can deal with the first balance problem (the one with respect to the combined CoG) in an indirect way via the front frame: leaning the upper body to one side will make the front and the rear frame lean to the other side (by conservation of angular momentum) and, depending on the geometrical properties of the bicycle, this lean may turn the front frame to the same side [55, 56].

Starting from the kinematic bicycle model in Fig. 4, I formulated EoM for the mechanical system, and these are derived in the Methods. These EoM are an extension of a linear 2-DoF benchmark bicycle model that has been extensively validated [55]. The extension involved replacing the rear frame of the benchmark model by a linearized DCIP of which the lower joint is actuated by steering instead of by joint-crossing muscles [57]. This model is called the Benchmark Double Pendulum (BDP, see Methods).

What sensory information informs the CNS about the lean angle?

One of the most challenging aspects of bicycle balance control pertains to the sensory feedback that informs the CNS about the lean angle (combined CoG angular position). A more detailed description of this problem is given elsewhere [10], and here I only summarize it. The problem is best understood by assuming that a bicycle is balanced using the same type of proprioceptive feedback as a standing person: ankle and hip joint accelerations over the same axis as the angular position the person wants to control. Because the joint between the rear frame and the road surface is not actuated by joint-crossing muscles, it also does not

provide proprioceptive feedback. Only the hip can be actuated by joint-crossing muscles and therefore only these muscles are a source of roll axis proprioceptive feedback. Not only does the CNS receive no proprioceptive feedback about the joint between the rear frame and the road surface, it also has no actuator to control this joint in a direct way. This joint can only be controlled indirectly by steering and rotating the hip joint. For this reason, the rider-bicycle combination belongs to the category of underactuated systems [58].

Bicycle balance simulation results

Fig. 5 shows simulation results for the four versions of the convolution predictor. It does not show the simulation results for the scenario without SMD compensation and the one with output gain scaling. The performance in these two scenarios is so poor that the lines would not be visible in the figure. For both scenarios, a 0.0025 sec. SMD resulted in lower body (bicycle rear frame) angular positions that were so large that the bicycle touched the road surface.

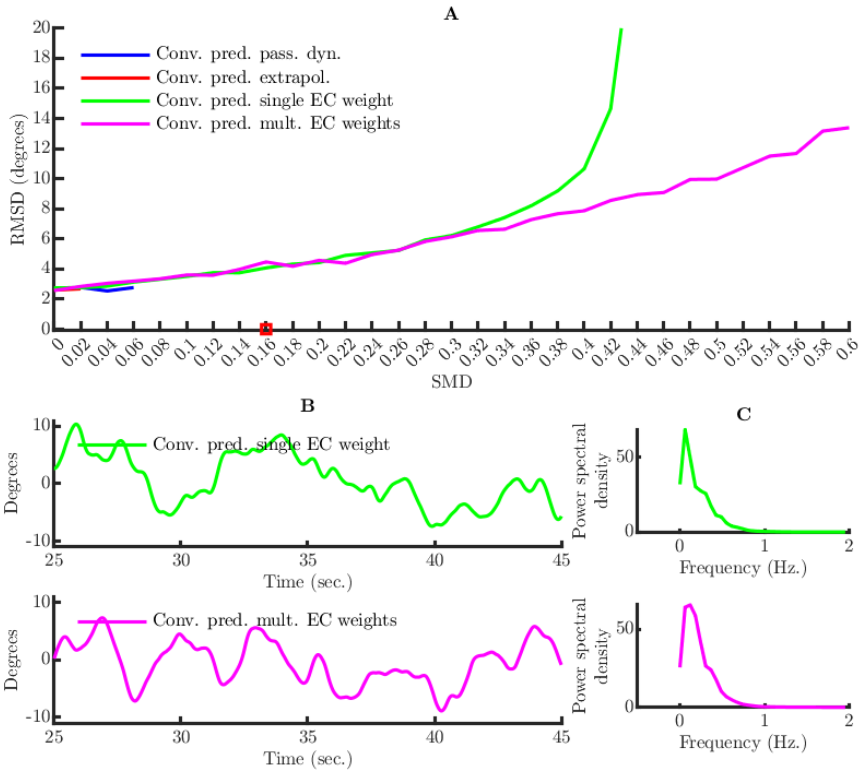

Fig. 5: Bicycle balance simulation study results. (A) Average RMSD CoG lean angle for four SMD compensation mechanisms as a function of SMD. (B) Example angular position time

series for the SMD that is indicated by a red box in panel A (SMD=0.16 sec.). (C) Power spectral density of the angular position for the same SMD as in B.

The performance of the convolution predictor strongly depends on whether a full memory buffer of efference copies was used. The performance of extrapolation was poor: for extrapolation based on passive dynamics only, the bicycle touched the road surface for SMD>0.07 sec., and for extrapolation using the most recent efference copy, it touched the road surface for SMD>0.035 sec. These critical SMDs are far below the neurobiological SMDs that are reported in the literature [4, 52, 59, 60].

A much better performance was obtained using the two versions of the convolution predictor that make use of a full memory buffer of efference copies (single and multiple efference copy weights). For the convolution predictor with a single efference copy weight, the bicycle started to touch the road surface for an SMD of approximately 0.41 sec. And for the convolution predictor with multiple efference copy weights this happened for an SMD of approximately 0.58 sec.

Different from standing balance control, for bicycle balance control, there is no experimentally determined critical lean angle RMSD beyond which cyclists feel too uncomfortable. I therefore used a rougher measure based on a plausible maximum for the combined CoG lean angle amplitudes. I calculated the lean angle amplitude that was required to take a 7 m. diameter corner at a speed of 15.5 km/h. (the speed used in the simulations) and found this to be 15 degrees inwards to the turn. Based on my own experience, this cornering task is feasible but challenging for experienced amateur mountain bikers. I assume that, without experience, this lean angle will feel uncomfortable, and I therefore calculated the SMD for which this critical lean angle amplitude was exceeded in more than one percent of the time. For the convolution predictor with a single efference copy weight, this happened for SMD>0.27 sec., and for the convolution predictor with multiple efference copy weights it happened for SMD>0.28 sec.

Thus, the convolution predictor is only effective if it makes use of a full memory buffer of efference copies to predict a future state, but there is little difference between a computationally hard convolution using multiple efference copy weights and a

computationally easy convolution using only a single weight. These results (necessity of a full memory buffer and sufficiency of a single weight) are not specific to incomplete exafferent acceleration feedback because they were also obtained with noisy full-state feedback. With this noisy full-state feedback, the only SMD compensation mechanisms that are effective for a neurobiologically plausible SMD are the two versions of the convolution predictor that make use of a full memory buffer of efference copies.

As for standing balance control, I confirmed the mathematical results for control performance of the Smith predictor: even for the smallest SMD (0.001 sec.), the Smith predictor produced diverging upper and lower body angular positions.

For the two successful combinations, Fig. 5B and 5C show example combined CoG angular position time courses and their corresponding average psds. These are added for completeness but it is not informative to compare them to the angular position time courses when cycling on the public roads. In fact, the latter task involves road boundaries, which requires control actions (steering to stay between the road boundaries) that are unrelated to balance control.

Discussion

I have described and evaluated three SMD compensation mechanisms: gain scaling, the convolution predictor, and the Smith predictor. These mechanisms are implemented using control theory results for linear dynamical systems [7, 8]. Dependence on results for linear dynamical systems can be motivated for balance control because the state of the body stays in a part of state space where a linear approximation of the EoM is accurate. I have evaluated these SMD compensation mechanisms both theoretically and by means of simulations. These are my conclusions:

1. The necessity of a SMD compensation mechanism depends on the mechanical system: no SMD compensation mechanism is required to balance a standing person with realistic joint stiffness and damping, but such a mechanism is required to balance a rider-bicycle combination. For conclusions 2-5, I focus on the rider-bicycle combination.
2. Compensating for a SMD requires the convolution predictor. This predictor involves a convolution over a memory buffer of efference copies.
3. The convolution predictor requires an initial condition obtained from a state observer that is based on a delayed-input forward model. Updating this state observer requires the oldest efference copy in the memory buffer of efference copies.
4. Compensating for a SMD does not require a computationally hard convolution with multiple efference copy weights; a similar performance is obtained with a computationally easy boxcar convolution that uses only a single weight.
5. Gain scaling is an effective SMD compensation mechanism but is not sufficient to compensate for a neurobiological SMD.
6. The Smith predictor is an ineffective and neurobiologically implausible SMD compensation mechanism for an unstable mechanical system.

Limitations of this study

I now discuss three of the study's main limitations: (1) the small set of mechanical systems and parameter configurations, (2) the absence of a systematic robustness evaluation, and (3) the confinement to a single task. First, it is difficult to rule out that the simulation results would have been different if different mechanical systems would have been used. Mechanical systems can differ in several aspects, such as geometry, mass distribution, joint stiffness and

damping, and the type of control input (torque, impedance and stiffness modulation, etc.). With respect to geometry, for the rider-bicycle mechanical system, I could not obtain the EoM for a more realistic form of cycling and therefore considered a restricted form of cycling in which the rider keeps his legs still, does not use them to carry weight, and relies on a motor for propulsion. However, most cyclists transfer a large part of their weight to the pedals and the handlebars; downhill mountain bikers (the balance artists in the cycling community) even do this for the full 100 percent. For more realistic simulations, we need EoM for a rider-bicycle mechanical system in which (1) the lower body is no longer a part of the rear frame, and (2) the AoS for the combined upper and lower body is formed by saddle, pedals, and handlebars. Such a mechanical system also implies a very different sensory system for the exafferent acceleration feedback because the forces (gravitational and centrifugal) now produce exafferent acceleration feedback in both the upper and the lower body. If the rider's legs carry weight (i.e., the saddle is at least partially unloaded) then the between-leg difference in load is informative about the resultant of the gravitational and the centrifugal force: because load is reflected in muscle tension, information about this resultant force can be obtained from leg muscle exafferent acceleration feedback.

Mechanical systems can also differ with respect to joint dynamics, and here I must distinguish between passive and active dynamics. Passive dynamics are all aspects of a mechanical system's dynamics that do not depend on the system's input, and it includes mass distribution, joint stiffness and damping. And active dynamics are all aspects of dynamics that depend on the input to the system, such as torque, impedance and stiffness. With respect to passive dynamics, I investigated the effect of mass distribution, joint stiffness, and joint damping on the simulation results, but not in a systematic way. In the Results, I reported on a few findings with respect to the stabilizing effect of joint stiffness and damping. With respect to active joint dynamics, the models only involved torque input, even though there are good arguments for the role of impedance and stiffness modulation [61, 62].

Second, this study is limited because it did not systematically investigate the robustness of the control performance to inaccuracies of the internal models. Robustness is relevant because the CNS must learn the internal models through experience with the mechanical system, and it is unrealistic to expect that these learned internal models exactly capture the mechanical system’s true dynamics (for the forward model) and the sensory feedback’s

relation with the state variables (for the sensory model). This issue is especially relevant for LQG control with incomplete feedback because there is an example of an LQG controller with arbitrarily small stability margins [63], which makes it highly susceptible to model deviations.

Although I did not systematically investigate robustness to internal model inaccuracies, the simulations of standing balance control using the nonlinear DCIP do provide information about the robustness of the linear convolution predictor. In fact, the linear convolution predictor could control the nonlinear DCIP for SMDs that far exceed neurobiologically plausible SMDs.

Third, this study is limited because of its focus on balance control. Balance control is a form of continuous motor control without a beginning or end. This differs from pointing, reaching and grasping, which pertain to episodes with an obvious beginning and end. Such tasks naturally lead to a distinction between feedforward and feedback control, and this distinction is closely related to the SMD. Between the beginning of an action and SMD seconds later, no sensory feedback has arrived in the CNS, and therefore only feedforward control is possible; when the first sensory feedback arrives, the CNS can switch to feedback control. During this period of feedforward control, for a convolution predictor that integrates over the full memory buffer of efference copies (no extrapolation), the prediction changes as more efference copies are added to the memory buffer. I did not investigate whether such a mixed feedforward-feedback convolution predictor can model the movement trajectories in a pointing, reaching or grasping task.

What is a plausible neurobiological implementation of the convolution predictor?

The Smith predictor has received a lot of attention in neuroscience because it has been linked to a neurobiological implementation with a central computational role for the cerebellum as a forward model with non-delayed input. This implementation involves two cortico-cerebellar loops [19]: one loop feeds a non-delayed state estimate $\hat{\boldsymbol{x}}(t)$ into the motor cortex, and the other loop estimates the delayed sensory feedback $\boldsymbol{y}(t)$ by the delayed state estimate $\hat{\boldsymbol{x}}(t - SMD)$.

Inspired by this neurobiological implementation of the Smith predictor, I now propose one for the convolution predictor (see Fig. 6). I hypothesize that the cerebellum computes convolutions over a memory buffer of efference copies. For the convolution over the full memory buffer (second term on the right side of Eq. 8), the convolution operator acts like a low-pass filter (see Fig. 6), and for the delayed-input state observer, the convolution operator selects the oldest efference copy from the memory buffer. The hypothesis that the cerebellum operates on a memory buffer of efference copies agrees with the fact that the cerebellum receives efference copies from the motor cortex, primarily through the cortico-ponto-cerebellar pathway. And the corollary that the cerebellum acts like a low-pass filter agrees with the fact that the most prominent symptom of cerebellar damage is a loss of smooth movements; cerebellar patients' movements are erratic, uncoordinated, and incorrectly timed. This pattern is denoted as dysmetria.

I hypothesize that the state observer is implemented in the parietal cortex. This agrees with the fact that the parietal cortex receives direct and short-range input from the sensory areas and possesses the recurrent connectivity necessary to implement a differential equation. Finally, I hypothesize that the motor cortex combines input from the parietal cortex (the state estimate) and the cerebellum (the outcome of the convolution over the memory buffer of efference copies) and computes a control signal using a feedforward neural network.

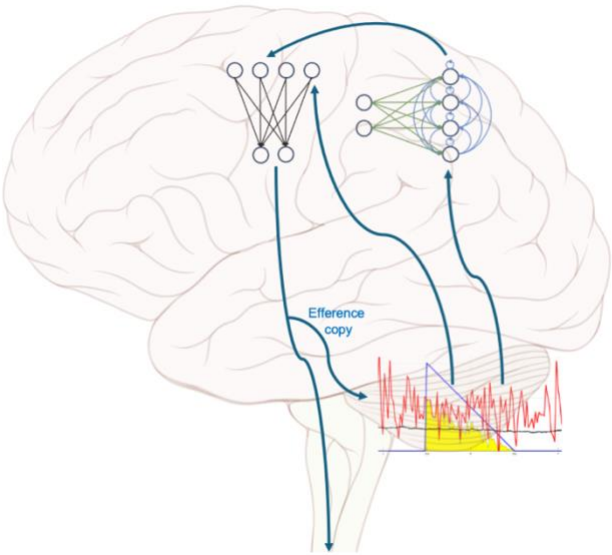

Fig. 6: A possible neurobiological implementation of the convolution predictor. The cerebellum computes convolutions over a memory buffer of efference copies and sends

output to the parietal and the motor cortex. A graphical representation of the convolution operation is superimposed on the cerebellum: the red signal is the efference copy input, the blue signal is the convolution kernel, and the black signal is the convolution output (the size of the yellow area under the thin grey line). The parietal cortex implements a state observer as a recurrent neural network and the motor cortex implements a feedforward neural network.

Spinal mechanisms for motor control

The less the number of synapses between sensory input and motor output, the smaller the SMD. Therefore, it may be that SMD compensation mechanisms are less important if control mechanisms can be transferred to the spinal cord. To be plausible, spinal neural circuity must allow for computations that can realize the flexible motor control that humans are able to; simple mono-synaptic reflex arcs are almost certainly insufficient [64].

An elegant mechanism has been proposed that adds flexibility to the spinal stretch reflex: stretch-reflex sensitization via gamma motor neuron activity. Gamma motor neuron activity increases the intrafusal fiber tension, making muscle spindles more sensitive to stretch: a stretch that would normally trigger few or no action potentials in the sensory afferents, would now trigger many more after sensitization. Via the spinal reflex arc, these action potentials would activate alpha motor neurons and cause muscle contraction. This mechanism has been proposed in different variants, such as Merton's servo hypothesis [65] and Feldman's equilibrium point (lambda) hypothesis [66]. Different from stretch-reflex sensitization via intrafusal fiber tension, these variants state that gamma motor neurons specify a desired muscle length.

In its simplest form, the hypothesis that alpha motor neurons are activated indirectly via gamma motor neurons is unlikely to hold. An important reason is that it predicts a time delay between alpha and gamma motor neuron firing, and this time delay has not been observed. Instead, alpha and gamma motor neurons are typically activated almost simultaneously [67], a phenomenon known as alpha-gamma co-activation. Alpha-gamma co-activation is consistent with a sensory role for gamma motor neuron activation: cancellation of the reafferent acceleration feedback, as proposed in [10] and in line with [31].

Indirect activation of alpha motor neurons via stretch-reflex sensitization has been extensively criticized in the literature, and alpha-gamma coactivation is only one of the arguments in this debate [66, 68, 69]. Here, I add an argument that follows from the fact that a bicycle can only be balanced by turning the handlebars. If this control action would depend on stretch-reflex sensitization, this would have to involve reflex arcs that involve the arm muscles. However, contrary to the spindles in the axial muscles of the lower torso, the spindles in the arm muscles cannot detect a stretch over the same axis as the gravitational and centrifugal torque that act on axial muscles of the lower torso. Thus, because reflex arcs operate locally (muscle spindles and associated alpha motor neurons belong to muscles of the same joint), they cannot be responsible for control actions that involve muscle spindles and alpha motor neurons that belong to distant segments of the spinal cord.

Prediction in perception and motor control

Prediction is a crucial concept in many subfields of neuroscience, and it is useful to distinguish between prediction in perception and prediction in motor control. I have argued that, for motor control in the presence of a SMD, prediction operates at the level of the output equation. This contrasts with the perception literature, where prediction is claimed to operate at the level of the state observer [70-74].

In the perception literature, prediction usually appears as a part of predictive coding, a theory that characterizes perception as an interaction between top-down (predictive) and bottom-up (sensory) influences [75-77]. This resembles the state observer which is a mathematical specification of the idea that perception is the result of a process that combines the CNS's prior belief (top-down) with sensory information (bottom-up) and a memory of its control signals. It is therefore no surprise that the state observer has been described from the perspective of predictive coding [71, 73, 74].

Predictive coding is a hierarchical state estimation network, inspired by the known anatomical connectivity of the visual cortex [78, 79]. It proposes a hierarchical updating scheme that repeats itself in the different levels of the visual hierarchy, each of which computes a state estimate. For reference, this is the discrete time delayed-input state observer:

$$\hat{\boldsymbol{x}}_{k+1}^{exp} = A\hat{\boldsymbol{x}}_k + B\boldsymbol{u}_{k-h} \qquad \text{Eq. 16}$$

$$\hat{\boldsymbol{x}}_{k+1} = \hat{\boldsymbol{x}}_{k+1}^{exp} + L\left(\boldsymbol{y}_k - C\hat{\boldsymbol{x}}_{k+1}^{exp}\right) \qquad \text{Eq. 17}$$

The relation between this state observer and predictive coding is not formally rigorous but rather loose, and this is in large part due to the numerous variants of predictive coding in the literature [80]. Here, I only consider the simplest variant, in which the efference copy $\boldsymbol{u}_{k-h}$ is removed from Eq. 16. In more complex variants [81, 82], the efference copy is replaced by a so-called hidden cause, which represents the brain's belief about the causes of the sensory input; the hidden states $\hat{\boldsymbol{x}}_k$ are an intermediate representation in between sensory input and hidden cause.

In all variants of predictive coding, $\boldsymbol{y}_k$ represents the input from a lower to the next higher layer in the computational hierarchy. Thus, only in the first layer of the hierarchy, $\boldsymbol{y}_k$ is a sensory signal with origin in the peripheral nervous system. In all higher layers, $\boldsymbol{y}_k$ is a feedforward signal with origin in the CNS. The properties of this signal depend on the type of perceptual representation (w.r.t. tuning and receptive field size) in the sending layer.

The simplest form of predictive coding involves the following computational recipe:

1. A level in the predictive coding hierarchy sends a top-down prediction $C\hat{\boldsymbol{x}}_{k+1}^{exp}$ to the next lower level.
2. The prediction-receiving level also receives a bottom-up input $\boldsymbol{y}_k$ from its next lower level.
3. The prediction-receiving level computes the sensory prediction error $\left(\boldsymbol{y}_k - C\hat{\boldsymbol{x}}_{k+1}^{exp}\right)$ and sends it to the next higher (prediction-generating) level.
4. The prediction-generating level uses Eq. 17 to update the prediction.
5. After the update, the old prediction $C\hat{\boldsymbol{x}}_{k+1}^{exp}$ is assigned the value $C\hat{\boldsymbol{x}}_{k+1}$ and the cycle repeats.

Predictions and prediction errors for motor control

There are different views about the signal that the CNS sends to the spinal cord, and here I distinguish between a signal based on a prediction and one based on a prediction error. Starting with the first, this is the core claim of the present paper: the CNS sends a torque request that is based on the difference between a predicted and a target state: $\boldsymbol{u}(t) = -K[\hat{\boldsymbol{x}}^{SMD}(t) - \boldsymbol{r}]$. The alternative view has been proposed as a part of active inference [70,

83-85], which is an extension of predictive coding. In active inference, the CNS sends a sensory prediction error $\left(\boldsymbol{y}_k - C\hat{\boldsymbol{x}}_{k+1}^{exp}\right)$ to the spinal cord. This prediction error is to be interpreted as a request for the spinal cord to minimize this error. Crucially, in the error signal $\left(\boldsymbol{y}_k - C\hat{\boldsymbol{x}}_{k+1}^{exp}\right)$, $C\hat{\boldsymbol{x}}(t)$ does not correspond to the predicted but to the desired sensory feedback; it has the same role as the target state $\boldsymbol{r}$ in the torque signal $-K[\hat{\boldsymbol{x}}^{SMD}(t) - \boldsymbol{r}]$. The prediction error $\left(\boldsymbol{y}_k - C\hat{\boldsymbol{x}}_{k+1}^{exp}\right)$ is thus a kinematic signal, and in that respect, it is like the input to a servo motor. However, it is different from a servo motor input in that the input signal must be minimized instead of realized.

In active inference, the alpha motor neurons are not controlled directly by the CNS but indirectly via reflex arcs that are assumed to minimize the sensory prediction error [70, 83-85]. The obvious candidate for this indirect activation is stretch-reflex sensitization via gamma motor neuron activity, which is also central in the equilibrium (lambda) point hypothesis [66]. An important difference between the two theories is that, under the equilibrium point hypothesis the kinematic input signal is realized by the reflex arcs, whereas under active inference it is minimized. However, many of the same criticisms that argue against indirect activation of alpha motor neurons via stretch-reflex sensitization [66, 68, 69] also apply to active inference. This includes the implausibility of bicycle balance control via stretch-reflex sensitization.

Whether based on predictions or prediction errors, to be useful, a computational mechanism must be able to control the states of the CNS-external mechanical system. That is, an important challenge lies in demonstrating that computations on CNS-internal states can control a CNS-external system. In this paper, I have demonstrated for two mechanical systems that this possible using a predictor that is computed as a convolution over a memory buffer of efference copies.

Methods

Matlab toolbox

All simulations were performed using the Matlab Balance Control (BalCon) toolbox that is shared in the supplementary information, together with the scripts that produced the simulation results, including the figures. For every mechanical system, the BalCon toolbox contains one function that computes the EoM and its linearization. The toolbox's core functions are generic, in the sense that they can be used for all mechanical systems for which both the nonlinear and the linearlized EoM are provided, and for all sensory systems that can be linearized.

A mechanical model for standing balance constrains the relation between exafferent joint acceleration and the state variables

Based on results from sensory neurophysiology, it can be argued that muscle spindle output is proportional to the exafferent joint acceleration component [10]. From the DCIP EoM, a sensory model can be derived. Here, I repeat this EoM:

$$\ddot{\boldsymbol{\theta}} = -C(\boldsymbol{\theta})\dot{\boldsymbol{\theta}} - K\boldsymbol{\theta} + F\sin(\boldsymbol{\theta}) + \boldsymbol{z}$$

And I rewrite it as a function of acceleration instead of torque:

$$\ddot{\boldsymbol{\theta}} = M(\boldsymbol{\theta})^{-1}\left[-C(\boldsymbol{\theta})\dot{\boldsymbol{\theta}} - K\boldsymbol{\theta} + F\sin(\boldsymbol{\theta}) + \boldsymbol{z}\right] \qquad \text{Eq. 18}$$

The right side of Eq. 18 can be linearized by computing the Jacobian matrices with respect to $\boldsymbol{\theta}$, $\dot{\boldsymbol{\theta}}$ and $\boldsymbol{z}$, and evaluating them at the unstable fixed point $\mathbf{0}$. I will denote these matrices by K_{stiff} , C_{damp} and M^{-1}, and they correspond to, respectively, stiffness, damping and mass moment of inertia. Importantly, K_{stiff} equals $M^{-1}(F-K)$ (see Eq. 18) and thus specifies the combined contribution of gravity and joint-intrinsic stiffness. The linearized EoM can then be written as follows:

$$\ddot{\boldsymbol{\theta}} \approx -C_{damp}\dot{\boldsymbol{\theta}} + K_{stiff}\,\boldsymbol{\theta} + M^{-1}\boldsymbol{z} \qquad \text{Eq. 19}$$

I start from Eq. 19 to derive an expression for the exafferent acceleration feedback provided by muscle spindles under fusimotor control [10]:

$$\ddot{\boldsymbol{\theta}} - M^{-1}\boldsymbol{w} \approx -C_{damp}\dot{\boldsymbol{\theta}} + K_{stiff}\,\boldsymbol{\theta} \qquad \text{Eq. 20}$$

In Eq. 20, I have replaced $\boldsymbol{z}$ (noisy control signal sent to the mechanical system) by $\boldsymbol{w}$ (noisy fusimotor input to the muscle spindles) to allow for a different noise component in the two signals. This is in line with the simulations, in which there was a 50% shared variance in $\boldsymbol{z}$ and $\boldsymbol{w}$. The term $M^{-1}\boldsymbol{w}$ approximates the reafferent joint acceleration and its subtraction from the total acceleration $\ddot{\boldsymbol{\theta}}$ cancels this term from the spindle output. Eq. 20 can be written in matrix notation as follows:

$$[I \quad -M^{-1}]\begin{bmatrix}\ddot{\boldsymbol{\theta}}\\ \boldsymbol{w}\end{bmatrix} \approx [K_{stiff} \quad -C_{damp}]\begin{bmatrix}\boldsymbol{\theta}\\ \dot{\boldsymbol{\theta}}\end{bmatrix} \qquad \text{Eq. 21}$$

The coefficient matrices on the left and the right side of Eq. 21 are, respectively, the green and the blue matrix C_{Sens} and C_{Comp} in Fig. 2B. Thus,

$$C_{Sens}\begin{bmatrix}\ddot{\boldsymbol{\theta}}\\ \boldsymbol{w}\end{bmatrix} \approx C_{Comp}\begin{bmatrix}\boldsymbol{\theta}\\ \dot{\boldsymbol{\theta}}\end{bmatrix} = C_{Comp}\boldsymbol{x}$$

This shows that there exists a linear combination of the sensory feedback variables $\ddot{\theta}$ and w that equals a linear combination of the state variables $\boldsymbol{x}$. The matrix product $C_{Comp}\boldsymbol{x}$ is a sensory internal model.

Simulations of standing balance control

The simulation algorithm

I now describe how to simulate the model without SMD in Fig. 2B. I will also mention the few adaptations that are required for a nonzero SMD, but I will not repeat the description of the SMD compensation mechanisms in the main text. I first describe the simulation of the mechanical system, the nonlinear DCIP. For this, I used the Matlab function ode45, which is based on an explicit Runge-Kutta (4,5) formula [86]. This function solves the DCIP EoM in state space form, which follows from Eq. 19:

$$\dot{\boldsymbol{x}} = \begin{bmatrix}\dot{\theta}\\ \ddot{\theta}\end{bmatrix} = \begin{bmatrix}\dot{\theta}\\ -C_{damp}\dot{\boldsymbol{\theta}} + K_{stiff}\,\boldsymbol{\theta} + M^{-1}\boldsymbol{z}\end{bmatrix}$$
$$= \Omega(\boldsymbol{x}, z)$$

Simulating a mechanical system in discrete time requires a solution of its EoM. In fact, $\boldsymbol{x}_{k+1}$ is obtained as the outcome of the function ode45 with input $\boldsymbol{x}_k$ and $\boldsymbol{z}_k$. And $\ddot{\boldsymbol{\theta}}_{k+1}$ is obtained from Eq. 18 with input $\boldsymbol{x}_{k+1}$ and $\boldsymbol{z}_k$. For simulating a mechanical system with a SMD, $\boldsymbol{z}_k$ is replaced by $\boldsymbol{z}_{k-h}$, in which h is the SMD as a multiple of the discretization time step.

Exafferent acceleration feedback $\boldsymbol{y}_{k+1}$ is computed by combining the acceleration output of the mechanical system ($\ddot{\boldsymbol{\theta}}_{k+1}$) with the output of the fusimotor system ($\boldsymbol{w}_{k+1}$), and the pure sensory noise ($\boldsymbol{s}_{k+1}$):

$$\boldsymbol{y}_{k+1} = C_{Sens}\begin{bmatrix}\ddot{\boldsymbol{\theta}}_{k+1}\\ \boldsymbol{w}_{k+1}\end{bmatrix} + \boldsymbol{s}_{k+1}$$

The computational system uses the following linear approximation of $\Omega(\boldsymbol{x}, z)$:

$$\begin{aligned}\dot{\boldsymbol{x}} &\approx \begin{bmatrix}0 & I\\ K_{stiff} & -C_{damp}\end{bmatrix}\boldsymbol{x} + \begin{bmatrix}0\\ M^{-1}\end{bmatrix}\boldsymbol{z}\\ &\approx A\boldsymbol{x} + B\boldsymbol{z}\end{aligned}$$

I assume that the mechanical system is deterministic, and therefore the motor noise m enters the mechanical system via the control signal $\boldsymbol{z}$: $Bz = B(u + m) = Bu + Bm$. Thus, the noise that is added to the mechanical system (Bm) is a linear function of the noise that is added to the control signal.

The computational system implements optimal control for the following linear dynamical system:

$$\begin{aligned}\dot{\boldsymbol{x}} &= A\boldsymbol{x} + B\boldsymbol{u} + B\boldsymbol{m}\\ \boldsymbol{y} &= C_{Comp}\boldsymbol{x} + M^{-1}\boldsymbol{m} - M^{-1}\boldsymbol{f} + \boldsymbol{s}\end{aligned}$$

with Gaussian motor ($\boldsymbol{m}$), fusimotor ($\boldsymbol{f}$), and sensor ($\boldsymbol{s}$) noise. The forward model in this system has a non-delayed input $\boldsymbol{u}$. To generalize this forward model to delayed input, I must express it explicitly as a function of time t: the forward model with non-delayed input involves $\boldsymbol{u}(t)$, and the one with delayed input involves $\boldsymbol{u}(t - SMD)$. This has only minor implications for the discrete time simulation algorithm: this algorithm must keep a memory buffer of efference copies $\boldsymbol{u}(t)$ and select the appropriate one.

The optimal control problem is a LQG and its solution is given by a linear differential equation that governs the state estimates $\hat{\boldsymbol{x}}$:

$$\dot{\hat{\boldsymbol{x}}} = A\hat{\boldsymbol{x}} + B\boldsymbol{u} + L(\boldsymbol{y} - C_{Comp}\hat{\boldsymbol{x}})$$

in which the control signal $\boldsymbol{u}$ equals $-K\hat{\boldsymbol{x}}$, $-K$ is the LQR gain and L is the Kalman gain [9]. Because of the definition of the motor and the fusimotor noise, there are a few minor adaptations to the usual way of calculating the Kalman gain: (1) the system noise covariance scaling matrix equals $B\mathrm{var}(\boldsymbol{m})B^t$ and thus depends on B, (2) the sensor noise covariance

scaling matrix contains a term that depends on the fusimotor noise (i.e., $M^{-1}\mathrm{var}(\boldsymbol{f})M^{-1^t}$), and (3) because the motor ($\boldsymbol{m}$) and the fusimotor ($\boldsymbol{f}$) noise can be correlated, this also holds for the system and the sensor noise.

Noise amplitude settings

To simulate DCIP dynamics under closed-loop feedback control, one must add noise. I set the noise amplitudes such that the effects of motor and pure sensor noise on the sensory feedback are equal. As a common scale for the effects of these three noise types, I use the noise variance of the sensory feedback $\boldsymbol{y}_k$: if the noise source is only sensory, then $var(\boldsymbol{y}_k) = var(\boldsymbol{s}_k)$, and if the noise source is only motor, then

$$var(\boldsymbol{y}_k) = B^{lo}\mathrm{var}(\boldsymbol{m}_k)B^{lo^t} + C_{Comp}B\mathrm{var}(\boldsymbol{m}_k)B^t C_{Comp}^t$$

The matrix B is the discrete time version of the corresponding input matrix B in the continuous time linear dynamical system in Eq. 21. And the matrix B^{lo} is the lower half of the discrete time input matrix B. The details of the derivation are given elsewhere [10].

In the simulations, I start from a noise scaling variable σ and (1) set $var(\boldsymbol{s}_k) = \sigma I$, in which I is an identity matrix of the appropriate dimension, (2) scale $var(\boldsymbol{m}_k)$ such that the expression on the right side of Eq. 12 equals σI, and (3) set $var(\boldsymbol{f}_k) = var(\boldsymbol{m}_k)$. I set the covariance $cov(\boldsymbol{m}_k, \boldsymbol{f}_k)$ such that the shared variance of $\boldsymbol{m}_k$ and $\boldsymbol{f}_k$ is 50%.

The variance of the noise terms $B\boldsymbol{m}_k$ (for the state equation, but also fed into the output equation) and $-B^{lo}\boldsymbol{f}_k + \boldsymbol{s}_k$ (for the output equation only) and their covariance are used as weights in the calculation of the Kalman gain L. This Kalman gain is a part of the computational system, and we thus implicitly assume that the CNS learns the sensor and the motor noise amplitudes from experience.

Physical and biophysical parameter settings

The simulations depend on several physical parameters that were assigned realistic and/or empirical values. I now list these parameters. First, the DCIP is specified by realistic and/or empirical values for its parameters:

1. Upper and lower body length: 0.75, resp., 1.1 m.

2. Upper and lower body mass: 50, resp., 35 kg.
3. Gravitational constant $g = 9{,}8066$.
4. Ankle stiffness $= 493.47$ Nm/rad. This number is based on measurements by [37], demonstrating that the ankle stiffness is 64% of the critical ankle stiffness (i.e., the ankle stiffness that would keep a standing person upright without active control).
5. Hip stiffness $= 117.68$ Nm/rad. This number is obtained in the same way as the ankle stiffness, assuming that the 64% also applies to the hip: hip stiffness is 64% of the critical hip stiffness.
6. Ankle damping $= 30$ Nm/(rad./s). This number was estimated by [37].
7. Hip damping $= 19.10$ Nm/(rad./s). This number was set such that the hip and the ankle damping ratios are equal. The hip damping was obtained by correcting the ankle damping for the different time constants of the two joints.

Second, for the simulations to be realistic, the computed ankle and hip torque must be less than the maximum voluntary contraction (MVC) of the muscle groups that must generate these two torque. Therefore, the output of the computational system is truncated at empirically determined MVC of these two muscle groups. For the ankle torque, I use plantarflexion MVC because humans prefer a leaned forward position [4]. In a study with 20 participants, the mean plantarflexion MVC was estimated to be 195 Nm [87]. To compute the hip torque MVC, I start from a setup involving a Roman chair in horizontal position on which a person lies sideways in a stretched position acting against gravity. The hip torque MVC is computed as the torque over the antero-posterior axis that is required to keep this challenging position. Using the length and mass parameter values in the previous paragraph, this MVC was estimated to be 184 Nm.

LQR gain weight matrices

The LQR gain $-K$ is the result of an optimization that involves an objective function specified by two weight matrices: a 4-by-4 precision matrix Q and a 2-by-2 energetic cost matrix R. These weights are set such that the precision (Q-dependent) and the energetic cost (R-dependent) component have an equal contribution to the objective function. I use the maximum metric to compute matrices Q and R that weight precision and energetic cost equally: Q and R are scaled such that $\boldsymbol{x}_{max}^{t} Q \boldsymbol{x}_{max}$ equals $\boldsymbol{u}_{max} R \boldsymbol{u}_{max}^{t}$.

The precision matrix Q has a block-diagonal structure $\mathrm{diag}(Q^{pos}, Q^{vel})$ with Q^{pos} a 2-by-2 matrix that corresponds to the two angular position state variables (for upper and lower body) and Q^{vel} a 2-by-2 matrix that corresponds to the two angular velocity state variables. The matrices Q^{pos} and Q^{vel} are set such that the position and the velocity component have an equal contribution to the objective function: Q^{pos} and Q^{vel} are scaled such that $\boldsymbol{\theta}_{max}^t Q^{pos} \boldsymbol{\theta}_{max}$ equals $\dot{\boldsymbol{\theta}}_{max}^t Q^{vel} \dot{\boldsymbol{\theta}}_{max}$. For the calculation of $\boldsymbol{x}_{max}$, I started from the critical CoG angular position interval, which is the interval in which the CoG must stay such that the center of pressure does not leave the area of support. I calculated this interval from the ankle position, the foot length, and the CoG for a configuration with identical upper and lower body angular positions (no hip flexion of extention). I found this interval to be [-0.0758, 0.2293] rad., which spans a width of 0.3051 rad. Thus, $\boldsymbol{\theta}_{max} = [0.3051, 0.3051]$. Next, I calculated the maximum angular velocity from the maximum body sway frequency, which was estimated to be 1.875 Hz in [53]. From this maximum frequency, I calculated a maximum angular velocity of 0.3655 rad./sec. $\dot{\boldsymbol{\theta}}_{max} = [0.3655, 0.3655]$. These values are sufficient to scale Q^{pos} and Q^{vel} such that the position and the velocity component have an equal contribution to the objective function.

The energetic cost matrix R is a diagonal 2-by-2 matrix, and it is scaled such that it has an equal contribution to the objective function as the precision (Q-dependent) component: $\boldsymbol{x}_{max}^t Q \boldsymbol{x}_{max} = \boldsymbol{u}_{max} R \boldsymbol{u}_{max}^t$. The maximum torques were inferred from the MVCs at the ankle and the hip joint, as described in an earlier paragraph: $\boldsymbol{u}_{max} = [195, 184]$ Nm.

Optimal gain scaling

In this paper, I applied optimal global scaling of the output gain: replace $-K$ (the LQR gain for SMD=0) by $-\alpha K$, in which α minimizes the LQR objective function. The minimization was performed using the Golden Section line search algorithm. As a part of this minimization routine, a function was called that computed a Monte Carlo estimate of the LQR objective function. This Monte Carlo estimate was computing the average LQR objective function over 10 simulation runs of 60 sec. each. Crucially, these simulations were ran with a nonzero SMD, and the outcome of the minimization is therefore specific for that SMD.

The benchmark double pendulum bicycle model

The benchmark double pendulum (BDP) is based on three ideas. The first idea is to follow the approach of [55] and derive linearized EoM for a bicycle with the rider's lower body rigidly attached to the rear frame and no upper body. These linearized EoM depend on a number of constants, and I chose these constants such that (1) the front frame is as similar as possible to the self-stable benchmark bicycle model described by [55]. The second idea is to model the interactions between the upper body and the rear frame (which includes the lower body) by the linearized EoM of the double compound pendulum, similar to [88]. Finally, the third idea is to first derive the BDP EoM without stiffness and damping terms, and to add these terms only in the last step.

The approach of [55] involves a method to calculate the defining matrices of linearized EoM of the following type:

$$\boldsymbol{M}\begin{bmatrix}\ddot{\delta}\\ \ddot{\theta}_1\\ \ddot{\theta}_2\end{bmatrix}+\boldsymbol{C}\begin{bmatrix}\dot{\delta}\\ \dot{\theta}_1\\ \dot{\theta}_2\end{bmatrix}+\boldsymbol{K}\begin{bmatrix}\delta\\ \theta_1\\ \theta_2\end{bmatrix}=0 \qquad \text{Eq. 22}$$

The matrices $\boldsymbol{M}, \boldsymbol{C}$ and $\boldsymbol{K}$ are functions of several constants (angles, lengths, masses, mass moments of inertia, gravitational acceleration, speed) that characterize the bicycle components and the internal forces that act on them. It is straightforward to express the EoM in state space form by rearranging the terms in Eq. 22.

[55] only derived linearized EoM for bicycles with a rider that was rigidly attached to the rear frame. Thus, the upper body angular position θ_2 is absent from their EoM. This missing component can be added by linearizing the DCIP EoM which models the interactions between θ_1 and θ_2. Schematically, each of the matrices $\boldsymbol{M}, \boldsymbol{C}$ and $\boldsymbol{K}$ is composed as follows:

$$\begin{bmatrix}\mathrm{MP}(1,1) & \mathrm{MP}(1,2) & 0\\ \mathrm{MP}(2,1) & \mathrm{MP}(2,2) & 0\\ 0 & 0 & 0\end{bmatrix}+\begin{bmatrix}0 & 0 & 0\\ 0 & \mathrm{DCIP}(1,1) & \mathrm{DCIP}(1,2)\\ 0 & \mathrm{DCIP}(2,1) & \mathrm{DCIP}(2,2)\end{bmatrix}$$

in which MP denotes matrix elements taken from Meijaard, Papadopoulos et al [55], and DCIP denotes matrix elements of DCIP EoM. The MP calculations were performed by means of the Matlab toolbox Jbike6 [89], in which I entered the constants for a bicycle with the rider's lower body rigidly attached to the rear frame and no upper body. This produced the constants $\mathrm{MP}(i,j)$ $(i,j=1,2)$ for $\boldsymbol{M}, \boldsymbol{C}$ and $\boldsymbol{K}$.

I now model the interactions between the upper and the lower body by the linearized DCIP EoM. I model the rear frame as a part of the lower body. I start from the following nonlinear EoM:

$$\begin{bmatrix} d_3 & d_4\cos(\theta_1-\theta_2) \\ d_4\cos(\theta_1-\theta_2) & d_5 \end{bmatrix}\begin{bmatrix}\ddot{\theta}_1 \\ \ddot{\theta}_2\end{bmatrix} + \begin{bmatrix} 0 & d_4\sin(\theta_1-\theta_2)\,\theta_2 \\ d_4\sin(\theta_1-\theta_2)\,\theta_1 & 0 \end{bmatrix}\begin{bmatrix}\dot{\theta}_1 \\ \dot{\theta}_2\end{bmatrix} + \begin{bmatrix}-f_1\sin(\theta_1) \\ -f_2\sin(\theta_2)\end{bmatrix} = \begin{bmatrix}0 \\ T_\theta\end{bmatrix}$$

These EoM include the hip torque input T_θ, but the ankle torque is set equal to 0. The constants d_3, d_4 and d_5 contain elements that must be added to the matrix $\boldsymbol{M}$, and the constants f_1 and f_2 contain elements that must be added to the matrix $\boldsymbol{K}$. These constants are defined as follows:

$$\begin{aligned} d_3 &= m_1 {l_1}^2 + m_2 {L_1}^2 + I_1 \\ d_4 &= m_2 L_1 l_2 \\ d_5 &= m_2 {l_2}^2 + I_2 \\ f_1 &= (m_1 l_1 + m_2 L_1) g \\ f_2 &= m_2 l_2 g \end{aligned}$$

The constants m_1, L_1, l_1 and I_1 are, respectively, the mass, the length, the CoG ($L_1/2$) and the mass moment of inertia of the DCIP's first rod, which represents the rear frame and the rider's lower body. The constants m_2, L_2, l_2 and I_2 are defined in the same way, but now for the second rod, which represents the rider's upper body. Finally, g is the gravitational constant.

I evaluate these EoM at $\theta_1 = \theta_2$ and replace $\sin(x)$ by its linear approximation near 0: $\sin(x) \approx x$. This results in

$$\begin{bmatrix} d_3 & d_4 \\ d_4 & d_5 \end{bmatrix}\begin{bmatrix}\ddot{\theta}_1 \\ \ddot{\theta}_2\end{bmatrix} + \begin{bmatrix} -f_1 & 0 \\ 0 & -f_2 \end{bmatrix}\begin{bmatrix}\theta_1 \\ \theta_2\end{bmatrix} = \begin{bmatrix}0 \\ T_\theta\end{bmatrix}$$

I will use the notation $\mathrm{DCIP}(i,j)$ $(i,j = 1,2)$ to denote these elements. For $\boldsymbol{M}$, the following elements are added:

- $\mathrm{DCIP}(1,1) = m_2 {L_1}^2$
- $\mathrm{DCIP}(1,2) = \mathrm{DP}(2,1) = d_4 = m_2 L_1 l_2$
- $\mathrm{DCIP}(2,2) = d_5 = m_2 {l_2}^2 + I_2$

And for $\boldsymbol{K}$, the following elements are added:

- $\mathrm{DCIP}(1,1) = m_2 L_1 g$
- $\mathrm{DCIP}(2,2) = -f_2 = -m_2 l_2 g$

Finally, I added stiffness and damping terms to, respectively, $\boldsymbol{K}$ and $\boldsymbol{C}$. As described in the next paragraph, these terms affect the dynamics of the steering and the hip joint.

Compared to the ankle joint in the DCIP model for standing balance control, much less is known about the stiffness and damping of the steering and hip joint in the BDP. These are not joints in the strict biomechanical sense because they involve more than the interface between two bones; the steering joint involves both the arms and part of the upper body, and the hip joint involves both the hip joint (head of femur and acetabulum) and the lumbosacral joint (lumbar spine and sacrum). For the steering joint, I calculated the stiffness from an empirically determined time constant, as described in [57]. And for the hip joint, I chose a stiffness coefficient such that the elastic force was 10 percent of the average (over upper and lower body) gravitational force; this allowed the upper and the lower body to fall with different accelerations. Evaluating the control performance using different damping ratios for the two joints (steering and hip) demonstrated robustness with respect to this parameter; control was possible over a range of four orders of magnitude around the critical damping ratio of 1 (i.e., for damping ratios between 0.01 and 100). I set the damping ratio for both joints at 20, which is a strongly overdamped system.

Simulations of bicycle balance control

Except for the definitions and values of the physical and biophysical parameters, the simulation algorithm is identical to the one for standing balance control. The dimensionality of the state space representation increases from 4 to 6, but this does not affect the operations that must be performed to simulate the trajectories of the state variables.

Compared to standing balance control, torque limits are much less important in bicycle balance control. This is because (1) turning the handlebars requires very little torque, and (2) upper body position can also be controlled by the centrifugal torque, and thus not only depend on muscular torque. This was clear from the simulations, which produced much smaller torque values for bicycle than for standing balance control.

References

1. Smith WM, McCrary JW, Smith KU. Delayed visual feedback and behavior. Science. 1960;132(3433):1013-4.
2. Miall R, Weir D, Stein J. Visuomotor tracking with delayed visual feedback. Neuroscience. 1985;16(3):511-20.
3. Gifford RN, Lyman J. Tracking performance with advanced and delayed visual displays. Human factors. 1967;9(2):127-32.
4. Rasman BG, Forbes PA, Peters RM, Ortiz O, Franks I, Inglis JT, et al. Learning to stand with unexpected sensorimotor delays. Elife. 2021;10:e65085.
5. Cunningham DW, Chatziastros A, Von der Heyde M, Bülthoff HH. Driving in the future: temporal visuomotor adaptation and generalization. Journal of vision. 2001;1(2):3-.
6. Miall R, Jackson J. Adaptation to visual feedback delays in manual tracking: evidence against the Smith Predictor model of human visually guided action. Experimental Brain Research. 2006;172:77-84.
7. Anderson BD, Moore JB. Optimal control: linear quadratic methods: Courier Corporation; 2007.
8. Stengel RF. Optimal control and estimation: Courier Corporation; 1994.
9. Åström KJ. Introduction to stochastic control theory: Courier Corporation; 2012.
10. Maris E. Internal sensory models allow for balance control using muscle spindle acceleration feedback. Neural Networks. 2025;189:107571. Epub 20250516. doi: 10.1016/j.neunet.2025.107571. PubMed PMID: 40412019.
11. Mirkin L, Palmor ZJ. Control issues in systems with loop delays. Handbook of networked and embedded control systems2005. p. 627-48.
12. Mirkin L, Raskin N. Every stabilizing dead-time controller has an observer–predictor-based structure. automatica. 2003;39(10):1747-54.
13. Molnár TG, Insperger T, editors. On the stabilizability of the delayed inverted pendulum controlled by Finite Spectrum Assignment in case of parameter uncertainties. International Design Engineering Technical Conferences and Computers and Information in Engineering Conference; 2013: American Society of Mechanical Engineers.
14. Mondié S, Michiels W. Finite spectrum assignment of unstable time-delay systems with a safe implementation. IEEE Transactions on Automatic Control. 2003;48(12):2207-12.
15. Palmor ZJ. Time-delay compensation-Smith predictor and its modifications. In: Levine WS, editor. The control handbook: CRC Press, Boca Raton; 1996. p. 224-37.
16. Crevecoeur F, Gevers M. Filtering compensation for delays and prediction errors during sensorimotor control. Neural computation. 2019;31(4):738-64.
17. Manitius A, Olbrot A. Finite spectrum assignment problem for systems with delays. IEEE transactions on Automatic Control. 1979;24(4):541-52.
18. Mehta B, Schaal S. Forward models in visuomotor control. Journal of Neurophysiology. 2002;88(2):942-53.
19. Miall RC, Weir DJ, Wolpert DM, Stein J. Is the cerebellum a smith predictor? Journal of motor behavior. 1993;25(3):203-16.
20. Hajdu D, Insperger T, editors. Time domain analysis of the Smith Predictor in case of parameter uncertainties: A case study. International Design Engineering Technical Conferences and Computers and Information in Engineering Conference; 2013: American Society of Mechanical Engineers.

21. Van Der Kooij H, Peterka RJ. Non-linear stimulus-response behavior of the human stance control system is predicted by optimization of a system with sensory and motor noise. Journal of computational neuroscience. 2011;30:759-78.
22. Bingham JT, Choi JT, Ting LH. Stability in a frontal plane model of balance requires coupled changes to postural configuration and neural feedback control. Journal of neurophysiology. 2011;106(1):437-48.
23. Le Mouel C, Brette R. Anticipatory coadaptation of ankle stiffness and sensorimotor gain for standing balance. PLoS computational biology. 2019;15(11):e1007463.
24. Miall RC, Wolpert DM. Forward models for physiological motor control. Neural networks. 1996;9(8):1265-79.
25. Wolpert DM, Ghahramani Z, Jordan MI. An internal model for sensorimotor integration. Science. 1995;269(5232):1880-2.
26. Wolpert DM, Kawato M. Multiple paired forward and inverse models for motor control. Neural networks. 1998;11(7-8):1317-29.
27. Franklin DW, Wolpert DM. Computational mechanisms of sensorimotor control. Neuron. 2011;72(3):425-42.
28. Wolpert DM. Computational approaches to motor control. Trends in cognitive sciences. 1997;1(6):209-16.
29. Macefield VG, Knellwolf TP. Functional properties of human muscle spindles. Journal of neurophysiology. 2018;120(2):452-67.
30. Dimitriou M. Human muscle spindles are wired to function as controllable signal-processing devices. Elife. 2022;11:e78091.
31. Dimitriou M. Human muscle spindle sensitivity reflects the balance of activity between antagonistic muscles. Journal of Neuroscience. 2014;34(41):13644-55.
32. Kalman RE. Contributions to the theory of optimal control. Bol soc mat mexicana. 1960;5(2):102-19.
33. Kalman RE, Bucy RS. New results in linear filtering and prediction theory. 1961.
34. Morasso PG, Sanguineti V. Ankle muscle stiffness alone cannot stabilize balance during quiet standing. Journal of neurophysiology. 2002;88(4):2157-62.
35. Morasso PG, Schieppati M. Can muscle stiffness alone stabilize upright standing? Journal of neurophysiology. 1999;82(3):1622-6.
36. Loram ID, Lakie M. Direct measurement of human ankle stiffness during quiet standing: the intrinsic mechanical stiffness is insufficient for stability. The journal of physiology. 2002;545(3):1041-53.
37. Casadio M, Morasso PG, Sanguineti V. Direct measurement of ankle stiffness during quiet standing: implications for control modelling and clinical application. Gait & posture. 2005;21(4):410-24.
38. Edin BB, Vallbo A. Dynamic response of human muscle spindle afferents to stretch. Journal of neurophysiology. 1990;63(6):1297-306.
39. Blum KP, Lamotte D'Incamps B, Zytnicki D, Ting LH. Force encoding in muscle spindles during stretch of passive muscle. PLoS computational biology. 2017;13(9):e1005767.
40. Dimitriou M, Edin BB. Discharges in human muscle spindle afferents during a key-pressing task. The Journal of physiology. 2008;586(22):5455-70.
41. Dimitriou M, Edin BB. Discharges in human muscle receptor afferents during block grasping. Journal of Neuroscience. 2008;28(48):12632-42.
42. Todorov E. Optimality principles in sensorimotor control. Nature neuroscience. 2004;7(9):907-15.
43. Scott SH. Optimal feedback control and the neural basis of volitional motor control. Nature Reviews Neuroscience. 2004;5(7):532-45.

44. Shadmehr R, Smith MA, Krakauer JW. Error correction, sensory prediction, and adaptation in motor control. Annual review of neuroscience. 2010;33(1):89-108.
45. Friston K. A theory of cortical responses. Philosophical transactions of the Royal Society B: Biological sciences. 2005;360(1456):815-36.
46. Wolpert DM, Flanagan JR. Motor prediction. Current biology. 2001;11(18):R729-R32.
47. Strang G. Differential equations and linear algebra. Cambridge: Cambridge University Press; 2015.
48. Wolpert DM, Miall RC, Kawato M. Internal models in the cerebellum. Trends in cognitive sciences. 1998;2(9):338-47.
49. Michiels W, Roose D, editors. Time-delay compensation in unstable plants using delayed state feedback. Proceedings of the 40th IEEE Conference on Decision and Control (Cat No 01CH37228); 2001: IEEE.
50. Peterka RJ. Sensorimotor integration in human postural control. Journal of neurophysiology. 2002;88(3):1097-118.
51. Maris E. Getting seniors back on their bicycle; a case-control study on the improvement of bicycle balance control. medRxiv. 2024:2024.04.17.24305755. doi: 10.1101/2024.04.17.24305755.
52. Forbes PA, Chen A, Blouin J-S. Sensorimotor control of standing balance. Handbook of clinical neurology. 2018;159:61-83.
53. Soames R, Atha J. The spectral characteristics of postural sway behaviour. European journal of applied physiology and occupational physiology. 1982;49:169-77.
54. Dong O, Graham C, Grewal A, Parrucci C, Ruina A. The bricycle: a bicycle in zero gravity can be balanced or steered but not both. Vehicle system dynamics. 2014;52(12):1681-94.
55. Meijaard JP, Papadopoulos JM, Ruina A, Schwab AL. Linearized dynamics equations for the balance and steer of a bicycle: a benchmark and review. Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences. 2007;463(2084):1955-82.
56. Kooijman J, Meijaard JP, Papadopoulos JM, Ruina A, Schwab A. A bicycle can be self-stable without gyroscopic or caster effects. Science. 2011;332(6027):339-42.
57. Maris E. A bicycle can be balanced by stochastic optimal feedback control but only with accurate speed estimates. PLoS one. 2023;18(2):e0278961.
58. Tedrake R. Underactuated Robotics: Algorithms for Walking, Running, Swimming, Flying, and Manipulation (Course Notes for MIT 6.832). Downloaded on 20-08-2021 from http://underactuated.mit.edu/. 2021.
59. Kuo AD. An optimal state estimation model of sensory integration in human postural balance. Journal of neural engineering. 2005;2(3):S235.
60. van der Kooij H, Jacobs R, Koopman B, Grootenboer H. A multisensory integration model of human stance control. Biological cybernetics. 1999;80:299-308.
61. Hogan N. The mechanics of multi-joint posture and movement control. Biological cybernetics. 1985;52(5):315-31.
62. Burdet E, Osu R, Franklin DW, Milner TE, Kawato M. The central nervous system stabilizes unstable dynamics by learning optimal impedance. Nature. 2001;414(6862):446-9.
63. Doyle J. Guaranteed margins for LQG regulators. IEEE Transactions on automatic Control. 2003;23(4):756-7.
64. Scott SH. A functional taxonomy of bottom-up sensory feedback processing for motor actions. Trends in neurosciences. 2016;39(8):512-26.
65. Merton P, editor Speculations on the servo-control of movement. Ciba Foundation Symposium-The Spinal Cord; 1953: Wiley Online Library.

66. Feldman AG. Once more on the equilibrium-point hypothesis (λ model) for motor control. Journal of motor behavior. 1986;18(1):17-54.
67. Vallbo Å. Muscle spindle response at the onset of isometric voluntary contractions in man. Time difference between fusimotor and skeletomotor effects. The Journal of physiology. 1971;218(2):405-31.
68. Matthews PB. The human stretch reflex and the motor cortex. Trends in neurosciences. 1991;14(3):87-91.
69. McIntyre J, Bizzi E. Servo hypotheses for the biological control of movement. Journal of motor behavior. 1993;25(3):193-202.
70. Friston K. What is optimal about motor control? Neuron. 2011;72(3):488-98.
71. Millidge B, Tang M, Osanlouy M, Harper NS, Bogacz R. Predictive coding networks for temporal prediction. PLOS Computational Biology. 2024;20(4):e1011183.
72. Deneve S, Duhamel J-R, Pouget A. Optimal sensorimotor integration in recurrent cortical networks: a neural implementation of Kalman filters. Journal of neuroscience. 2007;27(21):5744-56.
73. Huang Y, Rao RP. Predictive coding. Wiley Interdisciplinary Reviews: Cognitive Science. 2011;2(5):580-93.
74. Shi Z, Burr D. Predictive coding of multisensory timing. Current Opinion in Behavioral Sciences. 2016;8:200-6.
75. Rao RP, Ballard DH. Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. Nature neuroscience. 1999;2(1):79-87.
76. Clark A. Whatever next? Predictive brains, situated agents, and the future of cognitive science. Behavioral and brain sciences. 2013;36(3):181-204.
77. Lee TS, Mumford D. Hierarchical Bayesian inference in the visual cortex. JOSA a. 2003;20(7):1434-48.
78. Van Essen DC, Maunsell JH. Hierarchical organization and functional streams in the visual cortex. Trends in neurosciences. 1983;6:370-5.
79. Van Essen DC, Anderson CH, Felleman DJ. Information processing in the primate visual system: an integrated systems perspective. Science. 1992;255(5043):419-23.
80. Millidge B, Seth A, Buckley CL. Predictive coding: a theoretical and experimental review. arXiv preprint arXiv:210712979. 2021.
81. Friston K, Kiebel S. Predictive coding under the free-energy principle. Philosophical transactions of the Royal Society B: Biological sciences. 2009;364(1521):1211-21.
82. Friston K. Hierarchical models in the brain. PLoS computational biology. 2008;4(11):e1000211.
83. Friston K, Mattout J, Kilner J. Action understanding and active inference. Biological cybernetics. 2011;104:137-60.
84. Adams RA, Shipp S, Friston KJ. Predictions not commands: active inference in the motor system. Brain Structure and Function. 2013;218:611-43.
85. Pezzulo G, Rigoli F, Friston KJ. Hierarchical active inference: a theory of motivated control. Trends in cognitive sciences. 2018;22(4):294-306.
86. Shampine LF, Reichelt MW. The matlab ode suite. SIAM journal on scientific computing. 1997;18(1):1-22.
87. Vette AH, Masani K, Nakazawa K, Popovic MR. Neural-mechanical feedback control scheme generates physiological ankle torque fluctuation during quiet stance. IEEE Transactions on Neural Systems and Rehabilitation Engineering. 2010;18(1):86-95.
88. Dialynas G, Christoforidis C, Happee R, Schwab A. Rider control identification in cycling taking into account steering torque feedback and sensory delays. Vehicle System Dynamics. 2022:1-25.

89. A.L. Schwab JMP, A. Ruina and A. Dressel. JBike6, a benchmark for bicycle motion [cited 2022 14-7-2022]. Available from: http://ruina.tam.cornell.edu/research/topics/bicycle_mechanics/JBike6_web_folder.