Compensating for a sensorimotor delay requires a memory buffer, a state observer, and a predictor

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: **204**

The number of figures: **6**

The number of tables: 0

Abstract

Effective motor control requires sensory feedback about our body's state. However, adjusting motor commands based on sensory feedback is seriously complicated by the time delay between the state of the body at feedback generation and the arrival of the feedback-informed motor commands in the body's muscles, the so-called sensorimotor delay (SMD). In this paper, I investigate the effect of a SMD on the control performance of a computational system that represents the central nervous system. I specify three SMD compensation mechanisms that operate the level of this computational system: memory-informed state estimation, prediction, and output gain adjustment. I use formal analysis and computer simulations of balance control, both free standing and while riding a bicycle. This study provides strong evidence for the claim that SMD compensation requires a memory buffer of efference copies, a state observer, and a predictor. In addition, for prediction to be an effective SMD compensation mechanism, it needs (1) the output of a state observer as its initial condition and (2) a memory buffer of efference copies that must be convolved with a set of weights that is to be learned. For output gain adjustment to make a substantial contribution to SMD compensation, memory-informed state estimation and prediction must be imperfect.

Introduction

Our central nervous system (CNS) controls our body via motor commands and for this it needs sensory feedback about our body's state (its position and velocity relative to the environment). Unfortunately, adjusting motor commands based on sensory feedback is complicated by the time delay between the state of the body (the origin of the sensory feedback) and the effect of the feedback-based motor command in the body's muscles. This is the sensorimotor delay (SMD). For example, in balance control, the motor commands that keep the body upright lag behind 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 demonstrates that the CNS implements computational mechanisms that compensate for the SMD. In this paper, I will describe and evaluate three such mechanisms: memory-informed state estimation, prediction, and output gain adjustment.

To describe and evaluate different computational mechanisms, I will make use of control theory [7, 8]. And as a running example, I will use balance control, both free standing and while riding a bicycle. In both use cases, the bodies (standing person, rider, bicycle) most of the time stay close to the unstable fixed point (the upright position). At the unstable fixed point, the nonlinear equations of motion (EoM) can be approximated very well by a linear function, and I therefore will use control theory results for linear dynamical systems (LDSs).

Control theory allows to formulate the three SMD compensation mechanisms in terms of a CNS-internal dynamical system (the computational system) that controls a mechanical system (the human body and its physical environment, including gravity), which is also called the plant. Control is usually specified in terms of some optimality criterion/objective. In the time domain, this criterion is usually expressed in terms of a loss function, and in the Laplace domain, it is expressed in terms of the poles of a transfer function. 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 will use the LQG in simulations, the main ideas in this paper do not depend on the exact mathematical specification of the optimality criterion.

I propose that the CNS performs state estimation. The usefulness of state estimation follows from the fact that sensory feedback is very unlikely to be a perfect copy of the state variables of the mechanical system. In fact, if the latter were the case (i.e., feedback is full-state and noise-free), then optimal control is provided by control signals (motor commands) that are a simple linear mapping (gain scaling) of the sensory feedback. The neural implementation of such a linear mapping would be a simple two-layer linear feedforward neural network (FNN). Rather than being full-state and noise-free, it is very likely that our sensory organs produce feedback in which the states are only partially observable (denoted as incomplete feedback) and/or noisy [10]. The main control theory result for a linear mechanical system with incomplete sensory feedback is that optimal control is provided by gain scaling of the output of an optimal state estimator, also called a state observer. In LQG control, this optimal state observer is the Kalman filter [11, 12], which is itself a LDS. Because optimal control involves a dynamical system (instead of a mapping), it must be implemented by a recurrent neural network (RNN) [10].

A state observer is a formal implementation of the more general claim that the CNS uses internal models for motor control [13, 14]. Internal models may serve different purposes, and state estimation is one. Another use case for internal models is compensating for the destabilizing effects of a SMD, and this is the goal of the celebrated Smith predictor [15]. The Smith predictor uses an internal model of the mechanical system to predict the delayed sensory feedback from this system and to correct for it. Several papers claim that the Smith predictor can only be successfully applied to stable mechanical systems [16, 17], but this view is not shared among control theory experts [18]. This is an important issue for balance control, which always involves an

unstable mechanical system. And it is also important for reaching and grasping tasks in 3D because the passive dynamics of these tasks involve a destabilizing gravitational torque.

In this paper, I describe three computational mechanisms for SMD compensation. These mechanisms include state prediction, which is central to the Smith predictor, but will be applied in a very different way. They also include memory-informed state estimation and output gain adjustment. I start from the time domain formulation of a CNS-internal computational system, ask how the components of this system may contribute to SMD compensation, and propose neurobiologically plausible mechanisms that may implement these possibilities. The effectiveness of these mechanisms is evaluated in simulation studies on balance control, both freely standing and while cycling. These studies show that balance control requires a memory buffer of efference copies, which is used both for state estimation and prediction; without such a memory buffer, prediction even deteriorates the control performance in comparison with a delayed state estimate. These studies also support the claim that balance control under a neurobiologically realistic SMD requires a combination of SMD compensation mechanisms. The most informative results were obtained for bicycle balance control which, different from standing balance control, is impossible without SMD compensation mechanisms. Finally, these studies also demonstrate that the Smith predictor can control an unstable mechanical system, but that its performance is drastically inferior to other SMD compensation mechanisms.

Results

A closed-loop feedback control model with SMD compensation mechanisms

In Fig. **1**A and 1B, a closed-loop feedback control model is shown [19-21] to which SMD compensation mechanisms are added. In its application to standing balance control, the mechanical system (in red) is the body's musculo-skeletal system, which is depicted separately in Fig. 1C. This mechanical system is modeled as a compound inverted pendulum (CIP) that rotates about the ankle joint. The pendulum's angular position relative to gravity is denoted by θ and the muscular forcing torque at the ankle joint by $\boldsymbol{z}$. The movements of the CIP are fully specified by its EoM, which expresses the acceleration $\ddot{\theta}$ as a function of position θ, velocity $\dot{\theta}$, and the forcing torque z (see Eq. 7). The EoM is usually written as a function of the mechanical system's state, which is the combined position θ and velocity $\dot{\theta}$, and is denoted by $\boldsymbol{x} = \left[\theta, \dot{\theta}\right]^t$. The EoM can be written as a differential equation for the state $\boldsymbol{x}$, which I will denote as $\dot{\boldsymbol{x}} = \Phi(\boldsymbol{x}, \boldsymbol{z})$. This differential equation is usually called the EoM in state space form and is described in more detail in the Methods section.

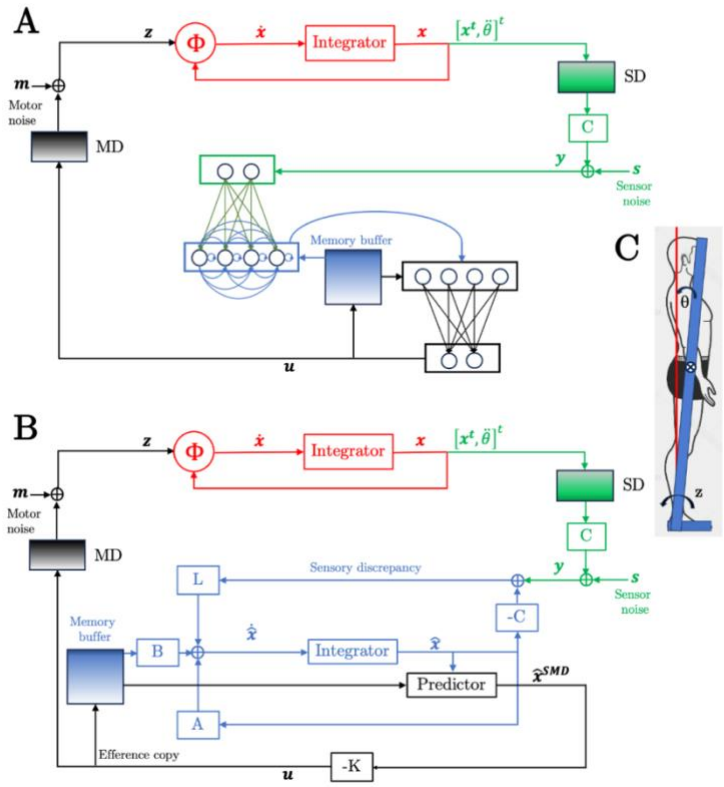

Fig. 1: Closed-loop feedback control model for standing balance with SMD compensation mechanisms. (A) Block diagram of the closed-loop feedback control model, with a mechanical (in red), a sensory (in green), a computational (green, blue, and black elements; see below), and a motor output system (in black). The computational system is a neural network that (1) receives the sensory feedback $\boldsymbol{y}$ in its input layer (green rectangle), (2) has an intermediate layer with recurrent connections (blue rectangle), and (3) produces output $\boldsymbol{u}$ via a feedforward network (two black rectangles). The black-and-white box depicts the delay between output $\boldsymbol{u}$ and the mechanical system state $\boldsymbol{x}$ (motor delay, MD), the green-and-white box depicts the delay between $\boldsymbol{x}$ and the sensory feedback $\boldsymbol{y}$ (sensory delay, SD), and the blue-and-white box depicts the memory buffer that is needed to compensate for the full sensorimotor delay (SMD). (B) The computational system is based on a linear internal 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 a sensory internal model C. These models, together with an optimality criterion for the control performance, defines the weights L and $-K$ of the neural network. (C) A standing body with a superimposed compound inverted pendulum rotating about the ankle, used

to model standing balance control. θ is the pendulum's angular position relative to gravity and z is the torque at the ankle joint.

The sensory system (in green) linearly maps the vector $\left[\boldsymbol{x}^{\boldsymbol{t}}, \ddot{\theta}\right]^{t}$ onto sensory variables, adds noise $\boldsymbol{s}$ and feeds the resulting feedback $\boldsymbol{y}$ into the computational system. Although it is not central to this paper, by including the angular acceleration $\ddot{\theta}$ 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 variables $\boldsymbol{x}$. This assumption is related to the fact that the matrices C in the sensory and the computational system (resp., green and blue in Fig. 1) can be different. This is motivated briefly in the Methods subsection *A mechanical model for standing balance constrains the relation between intrafusal fiber acceleration and the state variables* and more extensively in [10].

My interest is in how the computational system can control the mechanical system in the presence of a SMD, which is the sum of a motor delay MD and a sensory delay SD. The computational system computes (1) an internal state estimate $\hat{\boldsymbol{x}}$ by integrating a linear differential equation that takes as its input the sensory feedback $\boldsymbol{y}$ and an efference copy of a control signal, and (2) a control signal $\boldsymbol{u}$ that is based on a state prediction SMD seconds in the future ($\hat{\boldsymbol{x}}^{SMD}$). The standard in the literature is to model the computational system as a LDS that satisfies an optimality criterium, and this is depicted in 1B. This formalism is equivalent to a three-layered neural network: an input layer, a recurrently connected intermediate layer, and an output layer (see Fig. 1A). State estimation is performed by the recurrently connected intermediate layer, and I assume that it is implemented in the CNS. This is a plausible assumption given that recurrent connectivity is ubiquitous in the CNS. The equivalence between the algebraic formalism in Fig. 1B and a neural network will only be used to demonstrate its neural plausibility; no neural network implementation will be built.

The objective of the computational system is to control the mechanical system, and its formal specification requires an optimality criterion for the control performance. This criterion is a quantification of the person's objective to stay balanced using control signals that stay within the torque limits of the mechanical system. The best known optimality criterion is the one that underlies the LQG controller [9, 10], which is optimal for a linear dynamical system defined by (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 $\boldsymbol{y} = C\boldsymbol{x} + \boldsymbol{s}$ of the input to the computational system. In the LQG solution, $-K$ is the linear quadratic regulator (LQR) and L is the Kalman gain [9]. I assume that the CNS learns the state matrix A, the input matrix B, and the output matrix C, and sets the matrices K and L such that the LQG optimality criterion is achieved. The LQG solution also specifies the optimal neural network weights: the optimal input weights for the neural network are L, the optimal recurrent weights are $A - BK - LC$, and the optimal output weights are $-K$.

Crucially, optimality results such as those for the LQG controller only apply to systems without a SMD. And to the best of my knowledge, there are no optimality results for closed-loop systems with a nonzero SMD. I will therefore start from the LQG solution and will investigate the performance of different SMD compensation mechanisms that can all be formulated as modifications of the LQG solution.

The computational system without SMD compensation

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

$$\dot{\hat{\boldsymbol{x}}}(t) = A\hat{\boldsymbol{x}}(t) + B\boldsymbol{u}(t) + L[\boldsymbol{y}(t) - C\hat{\boldsymbol{x}}(t)] \qquad \text{Eq. 1}$$

$$\boldsymbol{u}(t) = -K[\hat{\boldsymbol{x}}(t) - \boldsymbol{r}] \qquad \text{Eq. 2}$$

Eq. 1 is called state equation and Eq. 2 the output equation. The state equation is also called a state observer or a Kalman filter (because L is the Kalman gain). The output equation produces 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, $\boldsymbol{r}$ specifies the stationary upright position. There is evidence that people prefer a slightly leaned forward position of approximately 0.0156 radians [4], and therefore $\boldsymbol{r} = [0.0156,0]^t$ is a plausible choice.

The state observer is a mathematical specification of the powerful idea that perception is the result of a process called state estimation 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 signals $\boldsymbol{u}(t)$. This idea dates back from Immanuel Kant in the 18th century: "The understanding can intuit nothing; the senses can think nothing. Only through their union can knowledge arise." [22].

The state observer specifies the state estimate derivative $\dot{\hat{\boldsymbol{x}}}(t)$ as the sum of two components: (1) an expected state derivative $A\hat{\boldsymbol{x}}(t) + B\boldsymbol{u}(t)$, which represents the CNS's belief about the expected change, and (2) a feedback-based correction $L(\boldsymbol{y}(t) - C\hat{\boldsymbol{x}}(t))$. The output equation computes a control signal $\boldsymbol{u}(t)$ from the state estimate $\hat{\boldsymbol{x}}(t)$. In engineering terms, the combined state and output equation specify a filter of the sensory feedback $\boldsymbol{y}(t)$ that produces the control signal $\boldsymbol{u}(t)$ as its output.

Note that the sensory feedback $\boldsymbol{y}(t)$ arrives in the computational system at time t, but is generated in the mechanical system at time $t - SD$. Thus, $y(t)$ provides feedback about $\boldsymbol{x}(t - SD)$, and the state estimate $\hat{\boldsymbol{x}}(t)$ lags the actual state by SD seconds. That is, $\hat{\boldsymbol{x}}(t)$ estimates $\boldsymbol{x}(t - SD)$. This has no implications for the length of the delay for which must compensated, which is the full SMD $(= SD + MD)$.

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 [23]. In this paper, I will use the term prediction in a very different sense, and therefore it is important to clarify this terminological difference. For that purpose, it is useful to describe the discrete time state observer as a two-step operation:

$$\begin{aligned} \hat{\boldsymbol{x}}_{k+1}^{exp} &= A\hat{\boldsymbol{x}}_k + B\boldsymbol{u}_k \\ \hat{\boldsymbol{x}}_{k+1} &= \hat{\boldsymbol{x}}_{k+1}^{exp} + L\left(\boldsymbol{y}_k - C\hat{\boldsymbol{x}}_{k+1}^{exp}\right) \end{aligned} \qquad \text{Eq. 3}$$

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 $h = SMD/\delta$ to denote the SMD length as an integer number of intervals of length δ. For notational simplicity, I have reused the symbols A, B, L and C of the continuous time state observer in Eq. 1, but the first three refer to different matrices than the corresponding ones for the continuous time observer.

The relevant term in Eq. 3 is $\hat{\boldsymbol{x}}_{k+1}^{exp}$, which I will denote as the expected state, but which is also denoted as a predicted state or an 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. In a later section on prediction, I will motivate a prediction of the state estimate at time $(k+h)\delta = k + SMD$ from the state estimate at time $k\delta$ and the efference copies $\boldsymbol{u}_k, \boldsymbol{u}_{k+1}, \dots, \boldsymbol{u}_{k+h}$. Obviously, this is a prediction over a neurobiologically meaningful interval. In this paper, I consider the state observer as a causal filter of the sensory feedback. This filter combines sensory feedback and efference copy information, but its output does not predict beyond the time of the sensory feedback, even though the discrete time version of this observer predicts over a discretization interval of length δ.

Three SMD compensation mechanisms

I will describe three SMD compensation mechanisms: (1) memory-informed state estimation, (2) prediction, and (3) output gain adjustment. The first and the last mechanism can both be motivated from prediction because (1) memory-informed state estimation is a requirement for accurate prediction,

and (2) output gain adjustment may improve control performance if prediction is inaccurate.

Motivation

Control performance would be unaffected by the SMD if it were possible to predict the state with the same accuracy as the state estimate $\hat{\boldsymbol{x}}(t)$ in the absence of a SMD. The future state that must be predicted is the mechanical state on which the control command will act, which is $\boldsymbol{x}(t + MD)$. For an approximately linear mechanical system, the predicted state is obtained by solving the forward model

$$\dot{\boldsymbol{q}}(t) = A\boldsymbol{q}(t) + B\boldsymbol{u}(t - SMD) \qquad \text{Eq. 4}$$

Here, I use the symbol $\boldsymbol{q}$ to distinguish this forward model from the original forward model $\dot{\boldsymbol{x}}(t) = A\boldsymbol{x}(t) + B\boldsymbol{u}(t)$. The forward model in Eq. 4 depends on the delayed efference copy $\boldsymbol{u}(t - SMD)$ and will be called the delayed input forward model. The original forward model depends on $\boldsymbol{u}(t)$ and will be called the forward model with non-delayed input. The correct forward model is the one with delayed input, and the predicted state is obtained by evaluating its solution at time $t + SMD$. As an initial condition for this solution, I use a state estimate $\hat{\boldsymbol{x}}(t)$ from a state observer. Memory-informed state estimation (a modification of the state observer in Eq. 1) estimates the delayed state $\boldsymbol{x}(t - SD)$, and therefore the solution at time $t + SMD$ (the prediction) estimates the future state $\boldsymbol{x}(t + MD)$, as required.

In the presence of motor noise, a prediction will always be less accurate than the state estimate that is used as the initial condition. This is because a prediction uses efference copies $\boldsymbol{u}$ from the interval $[t - SMD, t]$, which differ from the noisy motor commands $\boldsymbol{z} = \boldsymbol{u} + \boldsymbol{m}$ that enter the mechanical system. In such a situation, output gain adjustment may improve control performance.

Memory-informed state estimation

Memory-informed state estimation makes use of the following state observer:

$$\dot{\hat{\boldsymbol{x}}}(t) = A\hat{\boldsymbol{x}}(t) + B\boldsymbol{u}(t - SMD) + L[\boldsymbol{y}(t) - C\hat{\boldsymbol{x}}(t)] \qquad \text{Eq. 5}$$

The difference with the state observer in Eq. 1 is the expected state derivative $A\hat{\boldsymbol{x}}(t) + B\boldsymbol{u}(t - SMD)$, which is taken from the delayed-input forward model. The state observer in Eq. 1, on the other hand, takes the expected state derivative $A\hat{\boldsymbol{x}}(t) + B\boldsymbol{u}(t)$ from the forward model with non-delayed input.

Using the delayed-input forward model allows the state observer to estimate the mechanical state $\boldsymbol{x}(t - SD)$ (the state about which it gets sensory feedback $\boldsymbol{y}(t)$) also in the presence of a SMD, at least in a noise-free scenario (i.e., $\boldsymbol{z} = \boldsymbol{u}$, $\boldsymbol{y} = C\boldsymbol{x}$). To see this, assume that $\hat{\boldsymbol{x}}(t) = \boldsymbol{x}(t - SD)$ for some $t = t^*$. First note that the feedback-based derivative correction is 0 (because $\boldsymbol{y}(t) = C\boldsymbol{x}(t - SD) = C\hat{\boldsymbol{x}}(t)$), and the derivative of the state estimate ($\dot{\hat{\boldsymbol{x}}}(t)$, indicating where the state estimate will move) therefore equals the expected state derivative. Crucially, the state estimate $\hat{\boldsymbol{x}}(t)$ will move in the same direction as the mechanical state $\boldsymbol{x}(t - SD)$ because the expected state derivative $A\hat{\boldsymbol{x}}(t) + B\boldsymbol{u}(t - SMD)$ equals the mechanical state derivative $A\boldsymbol{x}(t - SD) + B\boldsymbol{u}(t - SMD)$. This will continue to hold for future times $t > t^*$, and therefore also $\hat{\boldsymbol{x}}(t)$ will remain equal to $\boldsymbol{x}(t - SD)$. This result depends on using the delayed-input forward model for the expected state derivative.

For memory-informed state estimation, the CNS must have 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, that is, between $t -$ SMD and t seconds. Memory-informed state estimation uses the oldest efference copy in this buffer, allowing it to evaluate the state observer in Eq. 5 and to compute a memory-informed state estimate.

By itself, memory-informed state estimation is a powerful SMD compensation mechanism if $\boldsymbol{x}(t+MD)$ (the mechanical state that is affected by the control signal) changes so slowly that it stays close to $\boldsymbol{x}(t-SD)$ (the mechanical state for which there is a memory-informed estimate). This mechanism is expected to perform poorly if the mechanical state changes fast, and in this case prediction may contribute substantially.

Prediction

The prediction mechanism for SMD compensation uses the same gain $-K$ as in the non-delayed scenario and applies it to a predictor $\hat{\boldsymbol{x}}^{SMD}(t)$ that estimates the future state $\boldsymbol{x}(t+MD)$:

$$\boldsymbol{u}(t) = -K[\hat{\boldsymbol{x}}^{SMD}(t) - \boldsymbol{r}] \qquad \text{Eq. 6}$$

The predictor $\hat{\boldsymbol{x}}^{SMD}(t)$ is obtained by evaluating the solution of the delayed-input forward model (see Eq. 4) at time $t+SMD$. This solution can be found in the specialized literature [24, 25] or derived from more general results in a textbook [26]:

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

in which the multiplication sign ($\times$) is only added for readability. Crucially, this solution depends on the initial condition $\boldsymbol{q}(t)$ and, technically, different choices can be made here. However, to obtain an accurate prediction of the future state $\boldsymbol{x}(t+MD)$, this initial condition must be an accurate estimate of $\boldsymbol{x}(t-SD)$, the mechanical state SMD seconds before this future state. This state estimate will be denoted by $\hat{\boldsymbol{x}}(t)$, and ideally it is obtained from memory-informed state estimation (see Eq. 5), but technically it could also be obtained from a state observer
that uses the most recent efference copy (see Eq. 1), or even a constant such as the target state $\boldsymbol{r}$. Thus, the predictor $\hat{\boldsymbol{x}}^{SMD}(t)$ can be written as follows:

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

Instead of considering Eq. 8 as a prescription for the CNS, I will use it as a starting point for motivating different predictors. For that purpose, it is

useful to approximate $\boldsymbol{u}(t)$ by a piecewise constant function, with piece length equal to δ, the defining time step for the discrete time version of the state observer. The corresponding discrete time predictions are denoted by $\hat{\boldsymbol{x}}_k^{SMD}$. The discrete time version of Eq. 8 can then be written as follows:

$$\hat{\boldsymbol{x}}_k^{SMD} = A^{SMD}\,\hat{\boldsymbol{x}}_k + \sum_{i=1}^{h} B^{i,\delta}\boldsymbol{u}_{k-i} \qquad \text{Eq. 9}$$

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

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

The prediction in Eq. 8 and Eq. 9 is well motivated for an approximately linear mechanical system, but there is no guarantee that the CNS can also compute it; the convolution in the second term in Eq. 8 and Eq. 9 is computationally heavy because it does not allow to reuse parts of previous computations. In addition, this term requires a memory buffer of efference copies $\boldsymbol{u}_{k-i}$ and a set of 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 (because 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, which 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. 9), (2) from the assumption that the efference copy remains constant at the most recent value (i.e., using $\boldsymbol{u}_{k-1}$ with a single weight $\sum_{i=1}^{h} B^{i,\delta}$), (3) using the same average prediction weight $\sum_{i=1}^{h} B^{i,\delta}/h$ for all efference copies, and (4) using the mechanically motivated Eq. 9. The first two predictions do not

make use of a memory buffer, and therefore can be considered as extrapolations based on the most recent information.

The method presented here is related but not identical to finite spectrum assignment [17, 25, 27-30]. The difference is that finite spectrum assignment does not involve a state observer. Instead, just like the Smith predictor, finite spectrum assignment assumes delayed full state feedback, that is, $\boldsymbol{y}(t)= \boldsymbol{x}(t-SD)$.

The Smith predictor [15] also uses prediction, but for a different goal: instead of using the predicted state for computing a control signal (see Eq. 6), the Smith predictor uses it to correct the sensory feedback for the SMD. I now describe the differences between the Smith predictor and the approach in this paper, for which I will use the same symbols although they are defined differently in the two approaches:

1. The Smith predictor assumes there is only a sensory delay SD and no motor delay MD. Thus, SMD=SD.
2. The Smith predictor can only be used for full state feedback. That is, $\boldsymbol{y}(t)= \boldsymbol{x}(t-SMD)$.
3. The prediction $\widehat{\boldsymbol{x}}^{SMD}(t)$ pertains to the delayed sensory feedback $\boldsymbol{x}(t-SMD)$, and is obtained by (1) evaluating the solution of the delayed-input forward model (see Eq. 7) at the time $t-SMD$, and (2) changing the limits of the integration term in Eq. 7 accordingly.
4. The solution that produces $\widehat{\boldsymbol{x}}^{SMD}(t)$ uses a fixed initial condition (e.g., the target state $\boldsymbol{r}$) rather than a state estimate; a feedback-informed initial condition would require a memory of the delayed sensory feedback $\boldsymbol{y}(t-SMD) = \boldsymbol{x}(t-2\times SMD)$.
5. The Smith predictor estimates the mechanical state $\boldsymbol{x}(t)$ using $\dot{\boldsymbol{x}}(t) = A\boldsymbol{x}(t) + B\boldsymbol{u}(t)$, the forward model with non-delayed input. The resulting state estimate $\widehat{\boldsymbol{x}}(t)$ pertains to a state that is SMD seconds in the future relative to the prediction $\widehat{\boldsymbol{x}}^{SMD}(t)$.
6. The Smith predictor computes the control signal $\boldsymbol{u}(t) = -K[\boldsymbol{y}(t) - \widehat{\boldsymbol{x}}^{SMD}(t) + \widehat{\boldsymbol{x}}(t) - \boldsymbol{r}]$. This computation has an intuitive appeal because $\widehat{\boldsymbol{x}}(t)$ is a forward model estimate of $\boldsymbol{x}(t)$ that is corrected by delayed

sensory feedback $\boldsymbol{y}(t) = \boldsymbol{x}(t - SMD)$, which is itself corrected by the predicted sensory feedback $\widehat{\boldsymbol{x}}^{SMD}(t)$.

The claim that the Smith predictor cannot be successfully used for an unstable mechanical system [16, 17] is based on the fact that, at least for a linear system and a perfect forward model, the estimation error $\boldsymbol{x}(t) - \widehat{\boldsymbol{x}}(t)$ does not converge to 0. This property is not sufficient for the intended claim, as examples have been given of unstable systems that can be stabilized by the Smith predictor [18]. However, it remains unclear whether the Smith prediction can stabilize a neurobiologically interesting system like a standing person or a rider-bicycle combination. This question will be addressed in this paper.

Output gain adjustment

Output gain adjustment has been proposed previously as a SMD compensation mechanism in balance control [31-33], and it was motivated quantitatively by a reduction in the region of stable output gains as the SMD increases. Its limits have also been described [33]. I now briefly reformulate these results using the terminology of this paper. The output torque scales with the output gain $-K$, and the larger its magnitude $|K|$, the more aggressive the corrections that bring the mechanical state to its target. With increasing SMD, it also becomes more likely that these more aggressive corrections destabilize the mechanical system. This is because the stability of a closed loop system depends on both the SMD and the magnitude of the output gain. This dependence can be expressed formally in terms of the output gain for the input frequency at which the open loop transfer function has a phase response of 180 degrees: the larger the SMD, the smaller the input frequency for which this transfer function has a 180 degrees phase response. To achieve stability, the amplitude of the open loop output gain at this critical frequency must be less than one.

Output gain adjustment can be implemented in multiple ways, and the simplest one is globally downscaling the output gain $-K$ (i.e., replacing $-K$

by $-\alpha K$ with $0 < \alpha < 1$). A more refined adjustment makes use of the fact that the first half of the columns of $-K$ correspond to the position variables (angular position) and the second half to their first derivatives (angular rate), and it involves separate scaling parameters for the two different column sets.

Performance of the SMD compensation mechanisms

In a series of simulation studies, I will evaluate the performance of the SMD compensation mechanisms that were described in the previous section. I now give a general description of these simulation studies, of which the details are given in the Methods.

1. The parameters of the mechanical systems are assigned realistic values, most of which are based on measurements.
2. The computational system uses an internal forward model that is a linear approximation of mechanical system at its unstable fixed point; the state matrix A and the input matrix B are obtained as Jacobians of the EoM.
3. I set the sensor and motor noise amplitudes such that, with SMD=0, the mechanical system exhibits a body sway/lean angle variability that is typical for unperturbed conditions. For standing balance (SB), I take the 0.20 degrees angular position RMSD from Fig. 6 (upper left panel) in [34]. And for bicycle balance (BB), I take the 1.15 degrees center of gravity (CoG) RMSD estimated from the recorded upper body and bicycle kinematics of a participant cycling on a treadmill-based bicycle simulator that requires steering to maintain balance.
4. The computational system uses an internal sensory model (specified by the output matrix C) for muscle spindle proprioceptive feedback [35]. This model is grounded in two properties of muscle spindles: (1) the firing rates of the muscle spindle primary afferents scale with intrafusal fiber acceleration [36, 37], and (2) this intrafusal fiber acceleration is controlled by fusimotor neurons, allowing for cancellation of the reafferent acceleration component [10, 37-39]. The sensory internal model that results from these properties 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 intrafusal fiber acceleration and the state variables*). Note, to avoid clutter, the fusimotor control of the sensory subsystem was not included in Fig. 1, but it is included in the corresponding Fig. 1 in [10].

5. I set the amplitudes of the motor and the fusimotor noise such that their combined effect on the sensory feedback roughly matches the effect of the pure sensory noise.
6. I compute the Kalman gain L using the actual motor and sensor noise amplitudes.
7. I compute the LQR gain $-K$ using weight matrices that weight the contribution of the precision and the energetic component equally, as quantified by realistic maximum values for the state vector and the control signal.
8. The predictor $\hat{\boldsymbol{x}}^{SMD}(t)$ that was used for approximating $\boldsymbol{x}(t + MD)$ (see Eq. 8 and Eq. 9), was implemented in four versions: (1) passive dynamics only, (2) extrapolation based on the most recent efference copy, (3) using the average prediction weight, and (4) exactly according to Eq. 9. The predictor $\hat{\boldsymbol{x}}^{SMD}(t)$ that is a part of the Smith predictor (for estimating $\boldsymbol{x}(t - SMD)$) was only implemented using the exact computation.
9. I evaluate an implementation of output gain adjustment that starts from the LQG-optimal gain matrix $-K$ for a non-delayed system, and uses two optimal scaling values, one for the columns that correspond to the position variables, and one for the columns that correspond to their first derivatives. For bicycle balance control, this reduced the dimensionality of the optimization problem from 12 (because the size of $-K$ is 2×6) to 2. The optimization algorithm used gradient descent in the resulting 2-dimensional search space and stopped when either (1) a local minimum was found or (2) the current solution (a scaled gain matrix) was not robust to an SMD increase of 0.02 sec. (i.e., rider and bicycle fell on the ground after this increase).

10. Every SMD compensation mechanism, as well as combination of mechanisms, will be evaluated based on the largest SMD for which the body sway amplitude stays below an empirically determined maximum. For SB control, this maximum was determined as the maximum angular position RMSD that people tolerate and is taken from Fig. 6 (upper right panel) in [34]. And for BB control, I take a CoG lean angle of 15.11 degrees, which corresponds to a turn radius that most people are not willing to take at the simulated speed (15.5 km/h).

A mechanical model for standing balance

As a mechanical model for standing balance (SB), I use the compound inverted pendulum (CIP). The CIP EoM is the following second-order differential equation:

$$I\ddot{\theta} + c_{damp}\dot{\theta} - \frac{1}{2}mgl\sin(\theta) + k_{stiff}\,\theta = z \ , \qquad \text{Eq. 10}$$

The variables $\ddot{\theta}$, $\dot{\theta}$, θ and z (acceleration, velocity, position, and forcing torque) vary as a function of time, but for notational convenience I omit their argument t. In Eq. 10, m is the mass of the compound pendulum, l is its length, and g is the gravitational constant. The constant I denotes the rotational inertia $I = \frac{1}{3}ml^2$. The ankle joint has both stiffness and damping, and these are characterized by the coefficients k_{stiff} and c_{damp}. The forcing torque z is produced by the muscles that cross the ankle joint, and it is the sum of a control signal plus motor noise ($z = u + m$). An important aspect of Eq 10 is the relation between k_{stiff} and the critical stiffness $\frac{1}{2}mgl$: if $k_{stiff} < \frac{1}{2}mgl$, the CIP is unstable in the upright position (θ accelerates away from 0), and if $k_{stiff} > \frac{1}{2}mgl$, it is stable. Several studies have demonstrated that the ankle joint stiffness k_{stiff} is less than the critical stiffness $\frac{1}{2}mgl$, and that this is due to the compliance of the Achilles tendon [40-43].

SB simulation study results for single SMD compensation mechanisms

The simulation results for single SMD compensation mechanisms plus no SMD compensation are shown in Fig. 2. At large SMDs, it can happen that the simulated angular position exceeds 90 degrees, indicating that the simulated person falls on the floor. This happened for all SMD compensation mechanisms but started at different SMDs. If this happened for one simulation run at a given SMD, it happened in all runs, and therefore in Fig. 2A no results are shown for these SMDs.

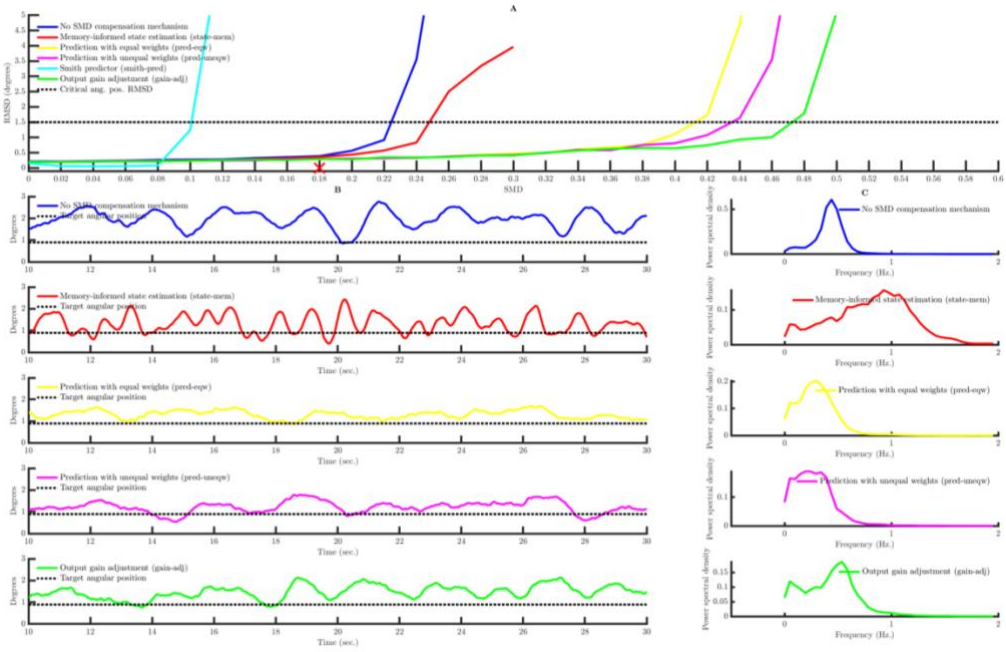

Fig. 2: SB simulation study results for single SMD compensation mechanisms. (A) Average RMSD body sway for five SMD compensation mechanisms plus no SMD compensation, as a function of SMD. Simulation runs were aborted if a simulated angular position exceeded 90 degrees (i.e., when the simulated person fell on the floor). The horizontal dotted line indicates the maximum angular position RMSD that people tolerate and is based on empirical work [34]. (B) Example angular position time series for the SMD that is indicated by a red X in panel A (SMD=0.18 sec.). (C) Power spectral density (psd) of the angular position for the same SMD as in B. The psd was calculated from the angular position time courses after removing the mean (DC component).

All SMD compensation mechanisms were successful for SMDs until 0.1 sec. This included the Smith predictor, which confirms the claim that this mechanism can stabilize an unstable system [18]. However, the Smith predictor has the worst performance (smallest maximum SMD that can be stabilized) of all mechanisms.

Without a SMD compensation mechanism, the critical angular position RMSD was exceeded for SMD>0.22 sec. (see Fig. 2A). Of the seven SMD compensation mechanisms, three clearly increase the SMD at which this critical RMSD is exceeded (pred-eqw, pred-uneqw, gain-adj; see the legend of Fig. 2A for the abbreviations), and these will be called effective. One mechanism made little difference (state-mem), and this will be called neither effective nor ineffective. Three mechanisms decrease the SMD at which the critical RMSD is exceeded, and they will be called ineffective. Of the three ineffective mechanisms, only the performance of the Smith predictor is shown in Fig. 2A. The Smith predictor uses a memory buffer of efference copies, but not to predict a state in the future. The other two ineffective mechanisms do not make use of a memory buffer of efference copies: prediction from passive dynamics only and prediction by extrapolating from the most recent efference copy.

Thus, the prediction mechanism is only effective if it makes use of a memory buffer of efference copies to predict a future state. There is little difference between the two prediction-based mechanisms that make use of a memory buffer to compute a future state (pred-eqw and pred-uneqw). This suggests that the differences between the weight matrices are not essential.

None of the single SMD compensation mechanisms can successfully compensate for an SMD of 0.56 sec., as human participants can [4]. This argues for a combination of SMD compensation mechanisms.

The angular position time courses for the effective SMD compensation mechanisms have different spectral characteristics (see Fig. 2B and 2C). Note that the average angular position exceeds its target value, which is due to the

positive gravitational torque that acts on the body when it is in its target position (slightly leaned forward). Compared to empirical body sway power spectral densities (psds), all mechanisms produce a psd that is much more concentrated around a peak frequency; empirical psds have a much more gradual drop-off between 0 and 2 Hz [44].

SB simulation study results for combined SMD compensation mechanisms

There is no a priori reason why the CNS should rely on only a single SMD compensation mechanism, and I therefore compared the performance of all pairs and all triplets of SMD compensation mechanisms (see Fig. 3). Consistent with the results for single SMD compensation mechanisms, all combinations that included a prediction mechanism that did not make use of a memory buffer of efference copies or that used prediction to compute a past state (the Smith predictor) had a worse performance than without this mechanism. It is implausible that the CNS would implement such a mechanism. Of these combinations, only a combination that included the Smith predictor is shown in Fig. 3.

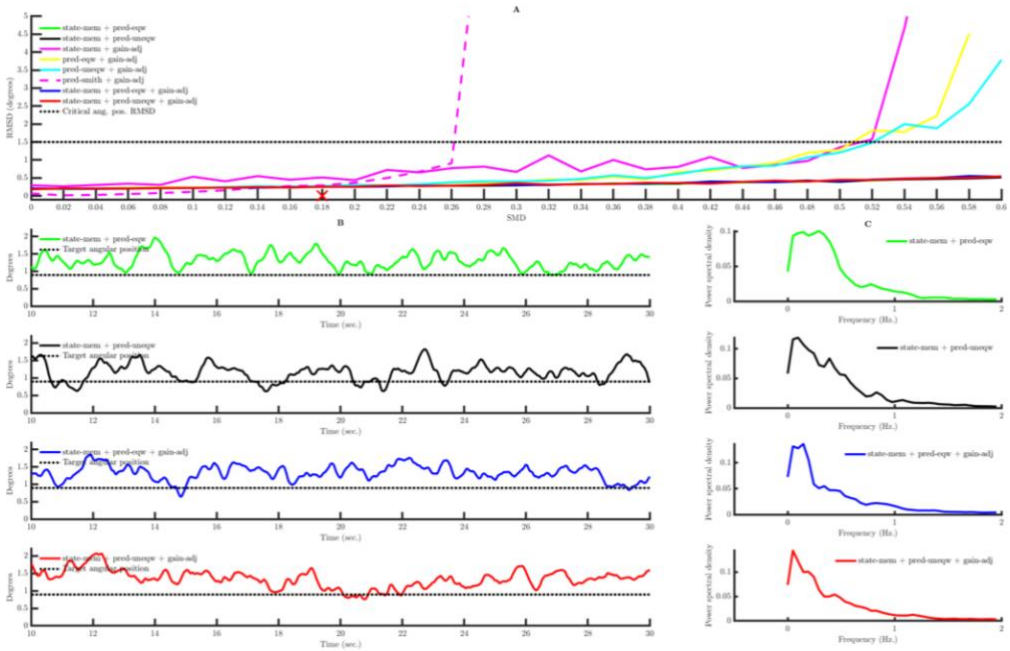

Fig. 3: SB simulation study results for combined SMD compensation mechanisms. See also the Fig. 2 caption. (A) Average RMSD body sway for eight combinations of SMD compensation mechanisms as a function of SMD.

(B) Example angular position time series for the four best combined SMD compensation mechanisms and the SMD that is indicated by a red X in panel A (SMD=0.18 sec.). (C) Power spectral density (psd) of the angular position for the same noise scale as in B.

Of the seven plausible combinations, four successfully compensate for the largest SMD that humans have been shown to deal with (0.56 sec.). These successful combinations all include memory-informed state estimation (state-mem) and a prediction mechanism that makes use of a memory buffer (pred-eqw or pred-uneqw); adding output gain adjustment (gain-adj) to these two mechanisms does not visibly improve the control performance (see the four overlapping traces in Fig. 3A). The performance of these four combinations remains under the critical RMSD for SMDs as large as 0.82 sec. Their psds all show the gradual drop-off between 0 and 2 Hz that is typical for empirical body sway psds (see Fig. 3C) [44].

Adding output gain adjustment to only one of the other three effective SMD compensation mechanisms (state-mem, pred-eqw, pred-uneqw) visibly improves the control performance (compare Fig. 2A and Fig. 3A): over the three mechanisms, the critical SMD is increased by at least 0.1 sec. Thus, in a suboptimal scenario (either no memory-informed state estimation, or no memory-informed prediction), output gain adjustment contributes to the control performance.

This simulation study has shown that (1) to be effective, prediction must be based on a memory buffer of efference copies, and this prediction must pertain to a future state, (2) to successfully compensate for the largest SMD that humans can deal with, a combination of SMD compensation mechanisms is required, (3) this combination must include memory-informed state estimation and prediction of a future state, and (4) output gain adjustment improves the control performance, but only in a suboptimal scenario.

This study made use of a mechanical system that, in the presence of biological SMD (0.14 sec.), can be controlled/balanced successfully without SMD

compensation mechanisms. I now consider a form of balance control that is much more challenging for the CNS: bicycle balance control.

A mechanical model for bicycle balance

Problem definition and bicycle mechanics

From a physics perspective, there are two important differences between 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 [45]. 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 within a fixed range. Besides the forces that act on the CoG, there are also forces that are responsible for the turning of the bicycle's front frame, and some of these are independent of the rider's actions [46]. These rider-independent forces are responsible for the bicycle's self-stability within a limited speed range [46].

For investigating bicycle balance control, 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 such), 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.

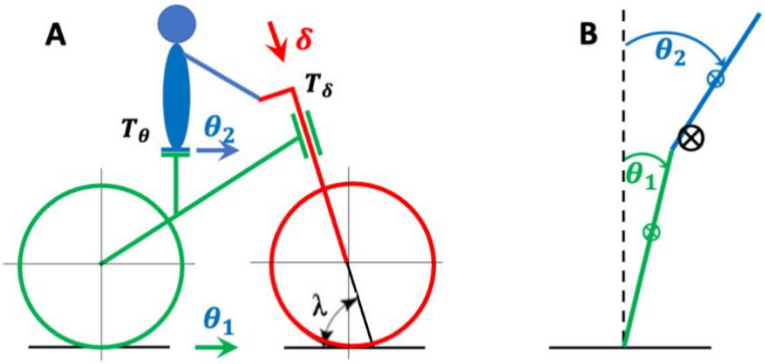

Fig 4: Kinematic variables of the bicycle model plus the rider-controlled forcing torques. (A) Side view. In green, the bicycle rear frame, characterized by its angular position θ_1 over the roll axis (green arrow). In red, the bicycle front frame, characterized by its angular position δ over the steering 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 pelvic torque T_θ, which are both applied by the rider, and (2) the steering axis angle λ (see text). (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 $\otimes$ denotes the CoG of the upper body (in blue), the lower body (in green), and the combined CoG (in black).

Cycling involves a double balance problem, and until now I have only described the first part: 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 AoS, the saddle. I only consider the balance over the roll axis (parallel to the LoS), which corresponds to upper body movements to the left and the right. I thus ignore the 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 accelerations and braking.

For both balance problems (with respect to the combined and the upper body CoG), the relevant control actions 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 pelvic torque T_θ). At this point, it is convenient to make use of Fig. 4B, which is a schematic of a double compound pendulum (DCP). 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, there is consensus that 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). An important argument in favor of this view 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 (i.e., 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 [46, 47].

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 [46]. The extension involved replacing the rear frame of the benchmark model by a linearized double pendulum [48]. This model is called the Benchmark Double Pendulum (BDP, see Methods).

Compared to the ankle joint in the CIP model for SB control, much less is known about the stiffness and damping of the steering and pelvic 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 pelvic 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 [48]. And for the pelvic 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 pelvic) 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.

<u>What sensory information informs the CNS about the lean angle?</u>

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 formal description of this problem is given elsewhere [10], and here I will only summarize it. The problem is best understood from the hypothesis that a bicycle can be balanced using the same type of proprioceptive feedback as for standing balance control. The latter involves ankle joint accelerations over the same axis as the angular position that the person wants to control. In bicycle balance control, there is only a single controllable joint over the same axis as the CoG angular position that the rider wants to control, namely the pelvis. Crucially, the joint between the rear frame and the road surface is not actuated by joint-crossing muscles, and therefore this joint does not provide proprioceptive feedback. This leaves only the muscles around the pelvis as a source of roll axis proprioceptive feedback. Note that this aspect of the bicycle geometry also has consequences for the actuation that can be used for control: the non-actuated joint between the rear frame and the road surface must be controlled indirectly by the

actuation the steering and the pelvic joint. For this reason, the rider-bicycle combination belongs to the category of underactuated systems [49].

BB simulation study results for single SMD compensation mechanisms

The results for single SMD compensation methods are straightforward: (1) without a SMD compensation mechanism, even a very small SMD (smallest delay tested was 10 ms.) resulted in angular positions exceeding 90 degrees (i.e., the bicycle touched the road surface), and (2) only memory-informed state estimation (state-mem) was successful in controlling the bicycle for larger SMDs, and it did so for a maximum of 0.12 sec. For this SMD, the RMSD of the combined CoG lean angle was 1.15 degrees, very close to the RMSD for a zero SMD (1.13 degrees); for larger SMDs, the bicycle touched the road surface in all simulation runs. Thus, no single SMD compensation mechanism can deal with a neurobiological SMD of 0.14 sec.

BB simulation study results for combined SMD compensation mechanisms

Consistent with the results for SB control, all combinations that included a prediction mechanism that did not make use of a memory buffer of efference copies or that did so predict a state in the past (i.e., the Smith predictor), had a worse performance than without this mechanism; they deteriorated the control performance. Except for one combination that included the Smith predictor, the combinations that included these mechanisms are not shown in Fig. 5. Of the seven remaining combinations, four compensate for a biologically plausible SMD (0.14 sec.) with a comfortable safety margin. These four successful combinations are the same as for SB control: they all include memory-informed state estimation and a prediction mechanism that makes use of a memory buffer.

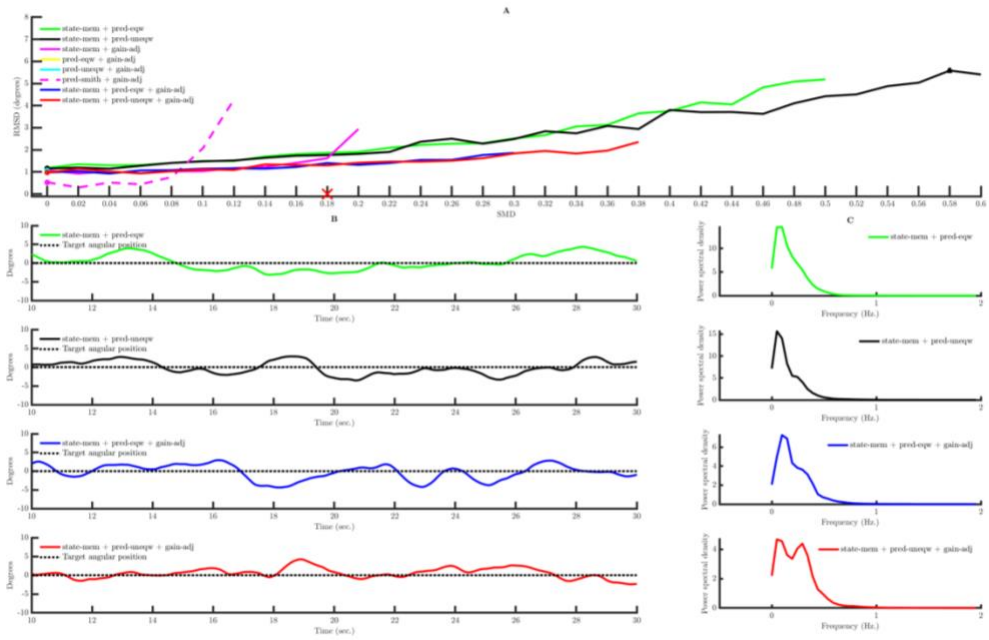

Fig. 5: BB simulation study results for combined SMD compensation mechanisms. See also the Fig. 2 caption. (A) Average RMSD combined CoG lean angles for eight combinations of SMD compensation mechanisms as a function of SMD. The black triangle indicates the SMD for which the combined CoG lean angles exceed a critical lean angle (see text) for more than 1 percent of the time. (B) Example angular position time series for the four best combined SMD compensation mechanisms and the SMD that is indicated by a red X in panel A (SMD=0.18 sec.). (C) Power spectral density (psd) of the angular position for the same noise scale as in B.

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

Of the four successful combinations, two included output gain adjustment (gain-adj), and this improved the control performance relative to memory-informed state estimation and prediction until a SMD of 0.3 sec. (for state-

mem+pred-eqw) and 0.38 sec. (for state-mem+pred-uneqw) (see Fig. 5A). After these SMDs, the combinations that included output gain adjustment resulted in a fall. Crucially, this is not an intrinsic property of output gain adjustment, but a consequence of a non-global minimum found by the algorithm that was used to solve the optimization problem that defined the optimal output gain. This optimization problem turned out to be non-convex. Without output gain adjustment, the control performance of memory-informed state estimation and prediction is only slightly inferior (see Fig. 5A).

Different from SB control, for BB 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. This happened only for the state-mem+pred-uneqw combination at SMD=0.58 sec. (see Fig. 5B).

Discussion

I have investigated the effect of a SMD on the control performance of a computational system (the CNS) that sends motor commands to a mechanical system (the body and its environment) from which it receives sensory feedback. The computational system is modelled as a LDS and I used this model to specify three SMD compensation mechanisms. I have used formal analysis and simulations to investigate the requirements for effective SMD compensation. These are my conclusions:

1. Compensating for a SMD requires (1) a memory buffer of efference copies, (2) a state observer, and (3) a predictor.
2. Updating the state observer requires the oldest efference copy in the memory buffer.
3. The predictor requires the output of the state observer as its initial condition and a convolution of efference copies as its forcing term.
4. The predictor must compute a future state.
5. In the optimal scenario of perfect memory-informed state estimation and prediction, output gain adjustment is not required for effective SMD compensation. Outside of this optimal scenario, it is likely to contribute to SMD compensation.

I will now elaborate on these points.

Compensating for a SMD requires a memory buffer, a state observer, and a predictor

By combining a predictor with an observer, I deviate from the classical Smith predictor, which predicts a state in the past using a fixed initial condition. The simulations have convincingly demonstrated the superiority of a prediction mechanism that computes a future state using an observer-produced initial condition and a convolution of efference copies.

The combination of a predictor and an observer combines aspects of feedforward and feedback motor control. The feedforward aspect is the predictor-based control signal which, at movement initiation, is computed

with a fixed initial condition; at this point, the observer does not yet provide a reliable state estimate. After SMD seconds, the observer can update the state estimate using sensory feedback and the associated efference copy. Control then switches to feedback mode: the predictor is computed with an observer-produced initial condition, which depends on the sensory feedback. This combination of feedforward and feedback control is especially relevant for reaching and grasping, which are finite horizon motor control problems. Such finite horizon problems do not require a movement trajectory that is completely specified at movement initiation. Instead, at movement initiation, it is sufficient to specify a feedback-independent initial condition.

The Smith predictor has received a lot of attention in neuroscience because it has been linked to a neurobiological implementation involving two cortico-cerebellar loops [15]: one loop feeds a non-delayed state estimate $\hat{\boldsymbol{x}}(t)$ into the motor cortex, and the other corrects the delayed sensory feedback $\boldsymbol{y}(t)$ by the prediction $\hat{\boldsymbol{x}}^{SMD}(t)$. This model assumes that the cerebellum implements the forward model on which both the state estimate and the prediction are based.

The combination of a state observer and a predictor suggests a different neurobiological implementation (see Fig. 6). In this implementation, the delayed sensory feedback is used for updating the state estimate and the prediction is sent to the motor cortex. The most plausible implementation for the state observer is parietal cortex, because it receives direct and short-range input from the sensory areas and possesses the recurrent connectivity necessary to implement a differential equation. The motor cortex combines the input from both the state observer (a state estimate) and the cerebellum (the effect of the delayed motor commands on the state estimate) and computes a control signal using a feedforward neural network (FNN). The cerebellum computes convolutions over a memory buffer of efference copies: for the state observer, the convolution operator selects the oldest efference copy from the memory buffer, and for the predictor, the convolution is a low-pass filter (see Fig. 6). This 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, which is collectively denoted as dysmetria.

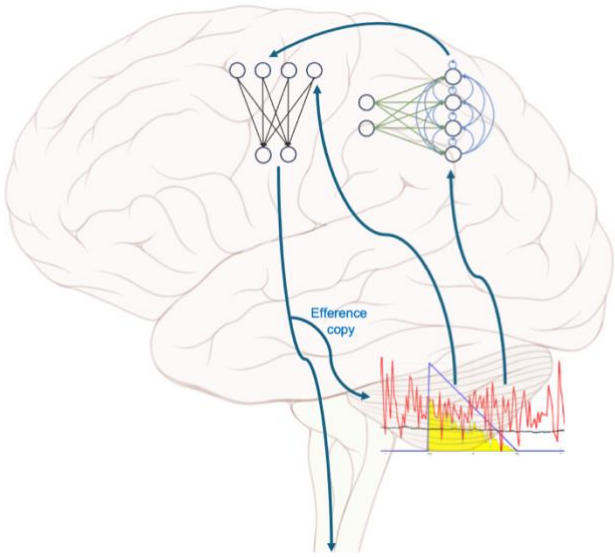

Fig. 6: A possible neurobiological implementation of a set of SMD compensation mechanisms that depend on a memory buffer, a state observer, and a predictor. Parietal cortex implements a state observer as a RNN and motor cortex implements a FNN. Both areas receive input from the cerebellum, which computes convolutions over a memory buffer of efference copies. 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).

A challenge of the prediction mechanism is its high computational load: a convolution must be evaluated and its kernel (the weight matrices $B^{i,\delta}$) must be learned from experience with the mechanical system. However, this prediction mechanism is robust against at least one deviation from the mechanically motivated Eq. 9: the mechanically motivated convolution kernel can be replaced by a boxcar with suitable height (the average weight matrix, resulting in the pred-eqw mechanism) without the control performance suffering much. This suggests that it is possible to start from an approximate convolution kernel and to improve it based on experience. However, and crucially, the convolution cannot be dropped without drastically

deteriorating the control performance; prediction using only passive dynamics or the most recent efference copy produces a worse control performance as compared to when no prediction is used. This demonstrates (1) the importance of the history of motor output signals for the prediction of a future body state, and (2) the necessity of a memory buffer that stores accurate memory traces of these signals (efference copies).

Why and how does the CNS use prediction?

Prediction is an important concept in many subfields of neuroscience, and it is useful to distinguish between perception and motor control. This paper is about motor control in the presence of a SMD, and my proposal is that prediction operates at the level of the output equation. This contrasts with the perception literature, where prediction is assumed to operate at the level of the state observer [50-54].

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 [55-57].
I have previously introduced the state observer as a mathematical specification of the idea that perception is the result of a process that combines the CNS's prior belief with sensory information and a memory of its control signals. Therefore, it is no surprise that the state observer has been described from the perspective of predictive coding [51, 53, 54]. However, the prediction that operates at the level of the state observer (predictive coding) differs from prediction that operates at the level of the output equation (this paper).

The starting point is the discrete time state observer in Eq. 3, which I copy here for convenience:

$$\hat{\boldsymbol{x}}_{k+1}^{exp} = A\hat{\boldsymbol{x}}_k + B\boldsymbol{u}_{k-h}$$
$$\hat{\boldsymbol{x}}_{k+1} = \hat{\boldsymbol{x}}_{k+1}^{exp} + L\left(\boldsymbol{y}_k - C\hat{\boldsymbol{x}}_{k+1}^{exp}\right)$$

The state estimates $\hat{\boldsymbol{x}}_{k+1}^{exp}$ and $\hat{\boldsymbol{x}}_{k+1}$ both play a role in predictive coding: the expected state $\hat{\boldsymbol{x}}_{k+1}^{exp}$ is called a prediction (a priori estimate) and $\hat{\boldsymbol{x}}_{k+1}$ is the new state estimate (a posteriori estimate) based on the prediction $\hat{\boldsymbol{x}}_{k+1}^{exp}$ and the sensory feedback $\boldsymbol{y}_k$. Central to predictive coding is the fact that the new state estimate depends on the sensory feedback via the sensory prediction error $\left(\boldsymbol{y}_k - C\hat{\boldsymbol{x}}_{k+1}^{exp}\right)$, in which $C\hat{\boldsymbol{x}}_{k+1}^{exp}$ is the predicted/expected sensory feedback. Predictive coding assumes a hierarchical perception/state estimation network, inspired by the known anatomical connectivity of the visual cortex [58, 59], and proposes a recurrent updating scheme for the different levels of this network: old state estimate → predicted/expected state → predicted/expected sensory feedback→ sensory prediction error → new state estimate. In the continuous time version of predictive coding, there is no prediction in the sense of predicting into the future. Instead, at every level of this continuous time hierarchical network, sensory feedback is integrated with the current state estimate and the efference copy memory as specified by the continuous time state observer in Eq. 1.

In this paper, prediction has been used in three different meanings: (1) predicting sensory feedback, which is a part of predictive coding and does not involve predicting into to future (nor the past) but creating an expectation about the current sensory feedback, (2) predicting a future state, which is an effective SMD compensation mechanism, and (3) predicting a past state, which is used in the Smith predictor to correct for a delayed sensory feedback but is an ineffective SMD compensation mechanism. As long as the different meanings of prediction remain separated, predicting a future or a past state is compatible with predictive coding (i.e., creating an expectation about the current sensory feedback). However, it is incompatible with an extension of predictive coding known as active inference [50, 60-62], which is a model of motor control. In the present paper, I used a standard model of motor control, of which the output is written as a scaled version of the difference between the predicted and the target state (see Eq. 6).

In active inference, the CNS controls movements by outputting a sensory prediction error $\boldsymbol{y}(t) - C\hat{\boldsymbol{x}}(t)$. Crucially, in this output, $\hat{\boldsymbol{x}}(t)$ corresponds to the target state $\boldsymbol{r}$. This is because, in active inference, (1) $\hat{\boldsymbol{x}}(t)$ does not corresponds to the actual but to the desired state, and (2) the muscles are not controlled directly by the CNS (as in the standard model by means of the torque commands $\boldsymbol{z}(t)$) but by reflex arcs that are assumed to minimize the sensory prediction error $\boldsymbol{y}(t) - C\hat{\boldsymbol{x}}(t)$ [50, 60-62]. Because spinal reflexes involve the activation of muscle spindles, the desired sensory states (also called sensory predictions) $C\hat{\boldsymbol{x}}(t)$ are assumed to be proprioceptive. Given that reflex arcs operate locally (muscle spindle afferents feed into alpha motor neurons of nearby muscles) such a scheme might work for standing balance control, in which the calf muscles could be both the origin of the sensory feedback and the actuators. The situation is very different for bicycle balance control, which requires the arm and shoulder muscles for turning the handlebars (i.e., their role as actuators), but whose muscle spindles do not provide sensory feedback about the CoG lean angle, which is over a different axis of rotation than the steering axis.

Active inference is very different from the mechanisms presented in this paper, which all assume that the CNS computes the motor commands. None of the crucial components of this paper (a memory buffer of efference copies, a state observer, a predictor) are required by active inference. Both approaches most likely would agree that the CNS can bring the mechanical state $\boldsymbol{x}(t)$ close to a target state $\boldsymbol{r}$, but they differ with respect to the computational mechanisms that are required to achieve this. Most importantly, they differ with respect to the internal state that is central in these computational mechanisms: in this paper, there is a central role for the predicted state $\hat{\boldsymbol{x}}^{SMD}(t)$, whereas in active inference this central role is for the desired state $\hat{\boldsymbol{x}}(t)$. Both approaches postulate mechanisms that attempt to minimize the difference between these internal states and some other variable (target state, resp., sensory feedback). For both approaches, the challenge lies in demonstrating that this minimization can also bring the mechanical state close to a target state (i.e., controlling the CNS-external world).

Output gain adjustment and combining SMD compensation mechanisms

The simulation studies showed that, if memory-informed state estimation and prediction are included as SMD compensation mechanisms, adding output gain adjustment improved the control performance but not in a spectacular way. For the interpretation of this result, it is important to consider that the simulations were performed in an optimal scenario for the ingredients of a linear internal model (best linear approximations A, A^{SMD}, $B^{i,\delta}$, C and L, and noise-free efference copies $\boldsymbol{u}_{k-i}$) or a deliberate approximation of this scenario (replacing $B^{i,\delta}$ by its average over i). Outside of this optimal scenario, output gain adjustment may contribute more to SMD compensation.

Related to this observation, there is good evidence that an improved prediction does not automatically imply improved control performance. Such evidence was obtained in a motor learning task, in which participants could predict the consequences of their actions before they could generate appropriate control actions [63]. Thus, the mapping between a prediction and a control action is probably not as simple as scaling with a fixed gain (see Eq. 4); output gain adjustment is one of the mechanisms that makes this mapping more flexible.

Output gain adjustment is especially attractive because of its low computational load: once the CNS has determined the optimal output gain for a given SMD, it can be kept fixed and does not require more resources than control without a SMD. In this respect, output gain adjustment is like joint stiffness/impedance adjustment as a mechanism to optimize performance in the presence of external perturbations [64].

It is highly plausible that the CNS uses a combination of SMD compensation mechanisms, especially if the ingredients for individual SMD compensation mechanisms (e.g., internal model parameters, accuracy of the efference copies, output gain scaling parameters) still can be improved. This is very

relevant for the interpretation of the results of empirical studies on SMD compensation [6, 16]: rather than looking for data patterns that are consistent or inconsistent with a single SMD compensation mechanism, the experimentalist faces the challenge that the CNS may compensate the weakness of one mechanism by relying more on another.

In sum, this study has provided evidence for three claims. First, SMD compensation requires a memory buffer of efference copies, a state observer, and a predictor that computes a future state. Second, for prediction to be an effective SMD compensation mechanism, it needs the output of a state observer and a memory buffer of efference copies. And third, in an optimal scenario for memory-informed state estimation and prediction, there is no spectacular contribution of output gain adjustment to SMD compensation, but this does not rule out its role in suboptimal scenario's.

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 their linearization. Most of the toolbox's functions are generic, in the sense that they can be used for all mechanical systems.

A mechanical model for standing balance constrains the relation between intrafusal fiber acceleration and the state variables

Based on results from sensory neurophysiology, it can be argued that muscle spindle output is proportional to the acceleration of the intrafusal fibers [10]. For muscle spindles that inform the CNS about the ankle joint, starting from the EoM in Eq. 10, one can demonstrate that this intrafusal fiber acceleration can be approximated by a linear combination of the state variables θ and $\dot{\theta}$:

$$\ddot{\theta} - I^{-1}w \approx I^{-1}\left[-c_{damp}\dot{\theta} + \left(\frac{1}{2}mgl - k_{stiff}\right)\theta\right] \qquad \text{Eq. 11}$$

The left side of Eq. 11 is the intrafusal fiber acceleration, which is obtained by subtracting the predicted reafferent joint acceleration $I^{-1}w$ (generated by fusimotor neuron activity) from the observed acceleration $\ddot{\theta}$. In matrix notation:

$$\begin{bmatrix}1 & -I^{-1}\end{bmatrix}\begin{bmatrix}\ddot{\theta}\\ w\end{bmatrix} \approx \begin{bmatrix}I^{-1}\left(\frac{1}{2}mgl - k_{stiff}\right) & -I^{-1}c_{damp}\end{bmatrix}\begin{bmatrix}\theta\\ \dot{\theta}\end{bmatrix} \qquad \text{Eq. 12}$$

The coefficient matrix on the left side of Eq. 12 is the green matrix C in the sensory system of Fig. 1A and 1B, and the coefficient matrix on the right side is the blue matrix C in Fig. 1B. These two matrices will be denoted by, respectively, C_{Sens} and C_{Comp}. Thus,

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

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 [13].

Simulations of standing balance control

For reproducing the simulations, one must know the noise input. This requires information about the different roles of the motor, fusimotor and sensor noise which has not been given in the main text. I now describe how to simulate the model in Fig. 1B. The starting point is the nonlinear CIP EoM in state space form, which follows from Eq. 10:

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} \dot{\theta} \\ I^{-1}\left(-c_{damp}\dot{\theta} + \frac{1}{2}mgl\sin(\theta) - k_{stiff}\,\theta + z\right) \end{bmatrix}$$
$$= \Omega(\boldsymbol{x}, z)$$

The variables $\ddot{\theta}$, $\dot{\theta}$, θ and z vary as a function of time, but for notational convenience I omit their argument t, and I will continue to do in this Methods section for all derived variables. The computational system uses a linear approximation of $\Omega(\boldsymbol{x}, z)$, which is obtained by taking the first order Taylor expansion around 0 (for θ, $\dot{\theta}$ and z):

$$\dot{\boldsymbol{x}} \approx \begin{bmatrix} 0 & 1 \\ I^{-1}\left(\frac{1}{2}mgl - k_{stiff}\right) & -I^{-1}c_{damp} \end{bmatrix}\boldsymbol{x} + \begin{bmatrix} 0 \\ I^{-1} \end{bmatrix} z$$
$$\approx A\boldsymbol{x} + Bz$$

Note that $Bz = B(u + m) = Bu + Bm$, which implies that the system noise (Bm) is a linear function of the motor noise. This follows from the assumption that the mechanical system is deterministic for a given input z because, in that case, the noise in the system's output only depends on the noise in its input.

The computational system implements optimal control for the following LDS:

$$\begin{aligned} \dot{\boldsymbol{x}} &= A\boldsymbol{x} + B\boldsymbol{u} + B\boldsymbol{m} \\ \boldsymbol{y} &= C_{Comp}\boldsymbol{x} + I^{-1}\boldsymbol{m} - I^{-1}\boldsymbol{f} + \boldsymbol{s} \end{aligned} \qquad \text{Eq. 13}$$

with Gaussian motor ($\boldsymbol{m}$), fusimotor ($\boldsymbol{f}$), and sensor ($\boldsymbol{s}$) noise. This 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., $I^{-1}\mathrm{var}(\boldsymbol{f})I^{-1^t}$), 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.

The mechanical system (the CIP) is specified by realistic and/or empirical values for its parameters: body length $l = 1{,}85$ m., body mass $m = 85$ kg., gravitational constant $g = 9{,}8066$, ankle stiffness $k_{stiff} = 493.4706$ Nm/rad. (64% of the critical stiffness), and ankle damping $c_{damp} = 30$ Nm/(rad./s) [43]. The CIP dynamics are simulated by the Matlab function ode45, which is based on an explicit Runge-Kutta (4,5) formula [65]. The input to the mechanical system is $\boldsymbol{z}_k = \boldsymbol{u}_k + \boldsymbol{m}_k$, with $\boldsymbol{u}_k$ the discrete time control signal, and the output is $\left[\boldsymbol{x}_{\mathrm{k+1}}^t, \ddot{\boldsymbol{\theta}}_{k+1}\right]^t$.

The feedback loop is closed by the sensory system that maps the output of the mechanical system ($\ddot{\boldsymbol{\theta}}_{k+1}$), the output of the fusimotor system ($\boldsymbol{w}_{k+1}$), and the pure sensory noise ($\boldsymbol{s}_{k+1}$) to the sensory feedback $\boldsymbol{y}_{k+1}$:

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

To simulate CIP dynamics under closed-loop feedback control, one must add noise. I set the noise parameters 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 LDS in Eq. 13. And the matrix B^{lo} is the lower half of the continuous 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 $var(s_k) = \sigma$, (2) scale $var(m_k)$ such that the expression on the right side of Eq. 10 equals σ, and (3) set $var(f_k) = var(m_k)$. I set the covariance $cov(m_k, f_k)$ such that the shared variance of m_k and f_k is 50%.

The variance of the noise terms Bm_k (for the state equation, but also fed into the output equation) and $-B^{lo}f_k + s_k$ (for the output equation only) and their covariance are used as weights in the calculation of the Kalman gain L_{dt}. 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.

To be realistic, the ankle torque in the simulations must be less than the maximum voluntary contraction (MVC) of the ankle muscles. Therefore, the output of the computational system is truncated at an empirically determined MVC. Because humans prefer a leaned forward position [4], I use plantarflexion MVC. In a study with 20 participants, the mean plantarflexion MVC was estimated to be 195 Nm [66].

The LQR gain $-K$ depends on the weights of an optimality criterium that is specified by two weights: a 2-by-2 matrix Q and a scalar R. These weights are set such that the precision (Q-dependent) and the energetic cost (R-dependent) component have an equal contribution to the optimality criterium. I use the maximum metric to calculate a diagonal matrix Q and a scalar R that produce equal values for $\boldsymbol{x}_{max}^t Q \boldsymbol{x}_{max}$ and $u_{max} R u_{max}$. For the

calculation of $\boldsymbol{x}_{max}$, I start from the critical angular position interval [-0.0758, 0.2293] rad., and calculate the critical angular velocity from the maximum angle position frequency of 1.875 Hz reported in [44]. This results in $\boldsymbol{x}_{max} = [0.2137 \quad 0.3655]^t$. I give angular position and angular velocity an equal contribution to the precision component by setting Q_{11} and Q_{22} such that $0.2137^2 Q_{11} = 0.3655^2 Q_{22}$. The critical torque $u_{max} = 195$ is obtained from [66].

The control signals were calculated as $-K(\hat{\boldsymbol{x}} - \boldsymbol{r})$, in which $\hat{\boldsymbol{x}}$ can be a state estimate or a state prediction, and the target state $\boldsymbol{r}$ equals $[0.0156 \quad 0]^t$ rad. This is in agreement with the observation that humans prefer a leaned forward position around 0.0156 rad. [4].

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 [46] 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 [46]. 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 [67]. 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 [46] involves a method to calculate the defining matrices of linearized EoM of the following type:

$$\boldsymbol{M}\begin{bmatrix}\ddot{\delta}\\ \ddot{\theta}_1\\ \ddot{\theta}_2\end{bmatrix} + \boldsymbol{C}\begin{bmatrix}\dot{\delta}\\ \dot{\theta}_1\\ \dot{\theta}_2\end{bmatrix} + \boldsymbol{K}\begin{bmatrix}\delta\\ \theta_1\\ \theta_2\end{bmatrix} = 0 \qquad \text{Eq. 14}$$

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. 14.

[46] 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 obtained by linearizing the double pendulum 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:

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

in which "MP" denotes "Meijaard, Papadopoulos et al" [46], and "DP" denotes "Double Pendulum". The MP calculations were performed by means of the Matlab toolbox Jbike6 [68], 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 body and the rear frame by the linearized EoM of the double compound pendulum. The nonlinear EoM of the double compound pendulum are the following:

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

These EoM include the torque input T_θ at the pelvic joint (between the upper and the lower body). 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:

$$d_3 = m_1 {l_1}^2 + m_2 {L_1}^2 + I_1$$

$$d_5 = m_2 {l_2}^2 + I_2$$
$$f_1 = (m_1 l_1 + m_2 L_1) g$$
$$f_2 = m_2 l_2 g$$

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 double pendulum's first rod, which represents the bicycle 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 $\mathrm{DP}(i,j)$ $(i,j = 1,2)$ to denote these elements. For $\boldsymbol{M}$, the following elements are added:

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

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

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

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

Simulations of bicycle balance control

Except for the definitions and values of a whole set of constants, the simulation algorithm is identical to the one for standing balance control. The dimensionality of the state space representation increases from 2 to 6, but this does not affect the operations that must be performed to simulate the trajectories of the state variables. The main difference with respect to the constants pertains to the weights Q and a scalar R for the LQR gain. These

weights are now calculated using maxima for the lean and the steering angle, the pelvic and the steering torque. 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) for rotating the upper body there are strong axial muscles, producing a large pelvic torque. As a maximum, I took the torque that is required to keep the upper body in a horizontal position on a Roman chair (a bed with an unsuspended upper body), but this maximum was never reached in the simulations.

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. An internal sensory model allows for balance control based on non-actionable proprioceptive feedback. arXiv preprint arXiv:240300951. 2024.
11. Kalman RE. Contributions to the theory of optimal control. Bol soc mat mexicana. 1960;5(2):102-19.
12. Kalman RE, Bucy RS. New results in linear filtering and prediction theory. 1961.
13. Miall RC, Wolpert DM. Forward models for physiological motor control. Neural networks. 1996;9(8):1265-79.
14. Wolpert DM, Ghahramani Z, Jordan MI. An internal model for sensorimotor integration. Science. 1995;269(5232):1880-2.
15. Miall RC, Weir DJ, Wolpert DM, Stein J. Is the cerebellum a smith predictor? Journal of motor behavior. 1993;25(3):203-16.
16. Mehta B, Schaal S. Forward models in visuomotor control. Journal of Neurophysiology. 2002;88(2):942-53.
17. Mirkin L, Palmor ZJ. Control issues in systems with loop delays. Handbook of networked and embedded control systems2005. p. 627-48.
18. 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.
19. Wolpert DM. Computational approaches to motor control. Trends in cognitive sciences. 1997;1(6):209-16.
20. Scott SH. Optimal feedback control and the neural basis of volitional motor control. Nature Reviews Neuroscience. 2004;5(7):532-45.
21. Todorov E. Optimality principles in sensorimotor control. Nature neuroscience. 2004;7(9):907-15.
22. Kant I. Critique of pure reason. New York: St. Martin's Press; 1781.

23. Shadmehr R, Smith MA, Krakauer JW. Error correction, sensory prediction, and adaptation in motor control. Annual review of neuroscience. 2010;33(1):89-108.
24. Manitius A, Olbrot A. Finite spectrum assignment problem for systems with delays. IEEE transactions on Automatic Control. 1979;24(4):541-52.
25. 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.
26. Strang G. Differential equations and linear algebra. Cambridge: Cambridge University Press; 2015.
27. Mirkin L, Raskin N. Every stabilizing dead-time controller has an observer–predictor-based structure. automatica. 2003;39(10):1747-54.
28. 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.
29. 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.
30. Crevecoeur F, Gevers M. Filtering compensation for delays and prediction errors during sensorimotor control. Neural computation. 2019;31(4):738-64.
31. 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.
32. 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.
33. Le Mouel C, Brette R. Anticipatory coadaptation of ankle stiffness and sensorimotor gain for standing balance. PLoS computational biology. 2019;15(11):e1007463.
34. Peterka RJ. Sensorimotor integration in human postural control. Journal of neurophysiology. 2002;88(3):1097-118.
35. Proske U. What is the role of muscle receptors in proprioception? Muscle & Nerve: Official Journal of the American Association of Electrodiagnostic Medicine. 2005;31(6):780-7.
36. Edin BB, Vallbo A. Dynamic response of human muscle spindle afferents to stretch. Journal of neurophysiology. 1990;63(6):1297-306.
37. Macefield VG, Knellwolf TP. Functional properties of human muscle spindles. Journal of neurophysiology. 2018;120(2):452-67.
38. Dimitriou M. Human muscle spindles are wired to function as controllable signal-processing devices. Elife. 2022;11:e78091.
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. Morasso PG, Sanguineti V. Ankle muscle stiffness alone cannot stabilize balance during quiet standing. Journal of neurophysiology. 2002;88(4):2157-62.
41. Morasso PG, Schieppati M. Can muscle stiffness alone stabilize upright standing? Journal of neurophysiology. 1999;82(3):1622-6.

42. 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.
43. 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.
44. Soames R, Atha J. The spectral characteristics of postural sway behaviour. European journal of applied physiology and occupational physiology. 1982;49:169-77.
45. 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.
46. 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.
47. 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.
48. Maris E. A bicycle can be balanced by stochastic optimal feedback control but only with accurate speed estimates. PLoS one. 2023;18(2):e0278961.
49. 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.
50. Friston K. What is optimal about motor control? Neuron. 2011;72(3):488-98.
51. Millidge B, Tang M, Osanlouy M, Harper NS, Bogacz R. Predictive coding networks for temporal prediction. PLOS Computational Biology. 2024;20(4):e1011183.
52. 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.
53. Huang Y, Rao RP. Predictive coding. Wiley Interdisciplinary Reviews: Cognitive Science. 2011;2(5):580-93.
54. Shi Z, Burr D. Predictive coding of multisensory timing. Current Opinion in Behavioral Sciences. 2016;8:200-6.
55. 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.
56. Clark A. Whatever next? Predictive brains, situated agents, and the future of cognitive science. Behavioral and brain sciences. 2013;36(3):181-204.
57. Lee TS, Mumford D. Hierarchical Bayesian inference in the visual cortex. JOSA a. 2003;20(7):1434-48.
58. Van Essen DC, Maunsell JH. Hierarchical organization and functional streams in the visual cortex. Trends in neurosciences. 1983;6:370-5.
59. Van Essen DC, Anderson CH, Felleman DJ. Information processing in the primate visual system: an integrated systems perspective. Science. 1992;255(5043):419-23.
60. Friston K, Mattout J, Kilner J. Action understanding and active inference. Biological cybernetics. 2011;104:137-60.

61. Adams RA, Shipp S, Friston KJ. Predictions not commands: active inference in the motor system. Brain Structure and Function. 2013;218:611-43.
62. Pezzulo G, Rigoli F, Friston KJ. Hierarchical active inference: a theory of motivated control. Trends in cognitive sciences. 2018;22(4):294-306.
63. Flanagan JR, Vetter P, Johansson RS, Wolpert DM. Prediction precedes control in motor learning. Current biology. 2003;13(2):146-50.
64. Selen LP, Franklin DW, Wolpert DM. Impedance control reduces instability that arises from motor noise. Journal of neuroscience. 2009;29(40):12606-16.
65. Shampine LF, Reichelt MW. The matlab ode suite. SIAM journal on scientific computing. 1997;18(1):1-22.
66. 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.
67. 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.
68. 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.