

The neural optimal control hierarchy for motor control

T DeWolf and C Eliasmith

Centre for Theoretical Neuroscience, University of Waterloo, Waterloo, Ontario, Canada

E-mail: tdewolf@uwaterloo.ca and celiasmith@uwaterloo.ca

Received 19 February 2011

Accepted for publication 2 September 2011

Published 4 November 2011

Online at stacks.iop.org/JNE/8/065009

Abstract

Our empirical, neuroscientific understanding of biological motor systems has been rapidly growing in recent years. However, this understanding has not been systematically mapped to a quantitative characterization of motor control based in control theory. Here, we attempt to bridge this gap by describing the neural optimal control hierarchy (NOCH), which can serve as a foundation for biologically plausible models of neural motor control. The NOCH has been constructed by taking recent control theoretic models of motor control, analyzing the required processes, generating neurally plausible equivalent calculations and mapping them on to the neural structures that have been empirically identified to form the anatomical basis of motor control. We demonstrate the utility of the NOCH by constructing a simple model based on the identified principles and testing it in two ways. First, we perturb specific anatomical elements of the model and compare the resulting motor behavior with clinical data in which the corresponding area of the brain has been damaged. We show that damaging the assigned functions of the basal ganglia and cerebellum can cause the movement deficiencies seen in patients with Huntington's disease and cerebellar lesions. Second, we demonstrate that single spiking neuron data from our model's motor cortical areas explain major features of single-cell responses recorded from the same primate areas. We suggest that together these results show how NOCH-based models can be used to unify a broad range of data relevant to biological motor control in a quantitative, control theoretic framework.

1. Introduction

Neuroscientific approaches to motor control are largely geared toward characterizing the basic anatomical elements and functions of the motor system. As such, they can be considered a largely 'bottom-up' approach to motor control—that is, an approach that attempts to identify the basic elements and functions of the system as a whole. These methods include the tracing of interconnections between different areas [12, 41], recording single cells to investigate neural sensitivity to different stimuli [27, 61], examining the effects of neural activity on motor output through stimulation experiments [71, 47] and observing lesioned animals or brain-damaged patients [39, 88]. This focus on specific, simple functions has resulted in theoretical models that tend to perform poorly when tested in novel contexts [72, 46, 81]. As a result, such models often do not provide general, unified explanations of motor system activity.

Control theoretic methods, on the other hand, can be considered a more 'top-down' approach to motor control. These methods focus on more global aspects of motor control, such as movement optimization according to a cost function [86], learning actions from observation [9], the generalization of movements such that controlled systems are robust to altered system dynamics or environmental changes [17], and so on. Adopting this approach usually entails generating simple limb or full body models, and then developing effective controllers that can perform a wide variety of movement with such models. However, because the dynamic complexity of real systems is often significantly greater than the generated models, relating the developed controllers to biological systems is often difficult. To address such challenges, additional complexity may be added incrementally to make the controlled model more like its biological counterpart. Only recently have models and corresponding control systems been created that begin to approach the complexity of a musculo-skeletal

redundant system, such as the human body [80, 14, 54]. These advancements notwithstanding, a more important concern for mapping to biological systems is that the proposed controllers are also typically not constrained by neurocomputational considerations, and, especially given their complexity, it is unclear which control algorithms are likely to be employed in real neural systems.

While some researchers adopt both neuroscientific and control theoretic approaches, there remains a tendency to focus on highly specific aspects of one or the other, to the detriment of their integration. Consider, for example, the incorporation of feedback delay into control theoretic models. The inclusion of such delay is inspired by observations that a significant amount of time is required for visual and proprioceptive signals to be relayed to motor areas in cortex [25, 76]. The problem with incorporating such information in this way, that is, out of the context of the motor system as a whole, is that it results in misleading claims about necessary neural functions. Specifically, while incorporating a system feedback delay is undoubtedly a useful exercise in motor control modeling, and it is true that cortex plays a central role in motor processing, control loops that occur in spinal circuits or cerebellum are also responsible for much motor activity. The amount of delay in some sensory signals in these areas is significantly different from that in cortex. No doubt, these systems work together to deal with delay issues effectively. Understanding how the neural system handles delay would thus be better served by a perspective that encompasses most of the motor system.

We believe that a central roadblock to integrating neuroscientific and control theoretic perspectives lies in the lack of a common framework that relates directly to both approaches. Several models developed in the past have worked toward bridging this gap, including MOSAIC [37] and MODEM [4]. However, such examples are primarily driven by control theoretic considerations, and do not provide a general mapping of control theory to neurobiological structures that capture motor-related neural activity. In particular, past work examines some specific neural system functions, and provides control theoretic characterizations, but these considerations sit in isolation from the majority of the motor system. However, we believe that just as the application of control theory to robotics control has exposed obstacles in the practical application of control theory [34, 69], so a general consideration of biological plausibility should help identify the appropriate control structures for explaining animal motor control. In short, a careful mapping of control structures onto the neural substrate will help identify control structures relevant to biological control, and will allow additional constraints from experimental observations of both normals and brain-damaged patients to be brought to bear on proposed models.

In this paper, we present the neural optimal control hierarchy (NOCH) framework as an initial attempt to realize such an integration. We argue that NOCH serves to further our understanding of the motor control system by bringing together current neurobiological research and control theoretic considerations into a biologically relevant framework. Our goal is to begin to provide a systemic and functional context for

neuroscientific investigations of the neural systems involved in motor control, while concurrently constraining the required, and plausible, computations that can be exploited by control theoretic models. While this integration is in early stages, here we argue that convincing explanations of a wide variety of neuroscientific evidence can be generated in the context of the proposed framework. To make this case, we begin with general considerations regarding the problem of motor control, and discuss past quantitative approaches to understanding the problem. We then present the NOCH framework by demonstrating how certain of these approaches map well to biologically identified functions and anatomical areas in some detail. Finally, we demonstrate how NOCH-based models are able to explain a variety of experimental observations. At the behavioral level, we demonstrate how NOCH captures the abnormal movements caused by both basal ganglia damage in Huntington's disease and various forms of cerebellar damage. At the neural level, we show how NOCH models also capture specific changes in spiking patterns of primary motor neurons generated during normal movement. Being able to integrate this breadth of empirical data with an explicit control theoretic model that is derived from a general framework can, we suggest, help set the stage for a deeper understanding of the neural underpinnings of motor control.

2. Mathematical models of motor control

Control theory research provides critical insight into the difficulties that efficient, robust control systems must overcome. Explicitly designing and building control systems highlight important practical and theoretical challenges, which is critical for addressing and surmounting these obstacles. Such an approach also offers motor control researchers' explicit functional solutions to motor control challenges that can direct how to empirically probe the underlying neural systems.

In this section, we review some current techniques used in control theory that address and overcome major problems related to designing robust controllers for noisy, complex systems. Specifically, we identify four important challenges faced by any attempt to design optimal motor controllers. In subsequent subsections, we discuss solutions to each of these challenges that have been incorporated into the NOCH framework.

The first challenge we consider is captured by the observation that human motor systems are able to reliably and repeatably execute motor actions successfully, accomplishing a given task while rarely explicitly reproducing the details of a particular motion. While there may be significant variation between different trials, the task itself (e.g. reaching to a target) is still successfully completed. This demonstrates that human movement is robust to intrinsic system noise and external perturbations. Optimal feedback control theory, discussed in section 2.1, allows for this kind of highly robust control, and as such is a basis of the motor control system model in the NOCH.

A second major obstacle faced in designing robust controllers is the generalization of control signals, or the

application of previously learned knowledge to new problems. For instance, after solving for a control signal for a complex system that performs a reach with an arm to a target location 1 foot away, it is desirable to reuse as much of this solution as possible when reaching to a position 1/2 a foot away. Similarly, it is desirable to use knowledge of a learned movement for ‘walking’ as a base for learning to run, rather than starting from a naive state. This kind of generalization process, however, is challenging to capture quantitatively. To address generalization, we look to compositionality theory, as discussed in section 2.2.

Another obstacle faced by complex motor system controllers is the coordination of the many degrees-of-freedom (DOF) required to successfully execute movements [84]. The third problem we address is this central topic in control theory research. Application of control theory to robotic systems highlights the computational costs related to DOF, and forces the consideration of solving such problems with limited system resources [30, 56]. As the number of DOF increase, the difficulty of controlling the system increases exponentially, a phenomenon known as the ‘curse of dimensionality’. To address this problem, focus has been placed on developing highly efficient control algorithms for high-dimensional spaces, and concurrently on reducing the system complexity by physically simplifying the controlled system. Here, we utilize hierarchical control techniques to allow the controller to act on a lower dimensional representation of the system [53, 54, 83], as discussed in section 2.3.

Finally, the fourth problem we consider is the high level of noise often found in motor systems, which affects both efferent and afferent signals, compromising command execution and the validity of sensory feedback. In addition to system noise, signal delays must also be accounted for to achieve effective control. Linear quadratic regulator (LQR) and Kalman filtering techniques are two methods that can be employed to operate in the context of such noise and signal delay, as we discuss in section 2.4.

In the remainder of this section, we detail technical solutions to these challenges that we have found map well to known anatomical and functional aspects of the biological control system. However, at this point we only present a technical characterization of these solutions. We return to a consideration of the specific anatomical mapping that comprises the NOCH after characterizing these technical solutions.

2.1. Optimal feedback control for motor modeling

Optimal control theory has long been incorporated into models of motor control [49]. This work has been driven by the notion that animals execute motor actions optimally, according to some cost function that defines the value of movement parameters and system resources. The system then plans and executes a movement which minimizes the cost on the way to the goal.

In traditional control systems, the planning and execution stages are separated. An optimal trajectory is pre-computed, and the completion of the task is dependent on the system’s

adherence to that path during movement execution. Optimal feedback control, however, operates differently. Planning and execution of a movement occur simultaneously, and a feedback control law is used to resolve moment-by-moment uncertainties, allowing the system to best respond to the current situation at each point in time. Additionally, instead of explicitly controlling all system parameters during a movement, a redundant subspace is identified, where variability will not affect task completion, and controller influence is exerted only over system elements moving outside this space. This is known as the ‘minimum intervention principle’ [84]. This principle has been empirically tested in several cases and shown to hold for human movement [90].

The minimum intervention principle and optimal feedback control theory are reasonable underpinnings for a motor system that has evolved over millennia, been augmented by learning and is constantly adapting to the environment. Systems designed using these principles give rise to many of the phenomena observed in human movement, such as task-constrained variability, goal-directed corrections and a very desirable robustness to load changes, noise and outside perturbations [84]. For these reasons, there has been a movement toward the use of optimal feedback control as a driving principle behind motor control models in the last decade [37, 74, 22, 84]. We also adopt optimal feedback control as a theoretical cornerstone of the NOCH. Next, we formally define optimal control and discuss several implementation techniques.

2.1.1. The Bellman equation. To determine an optimal action in a given state, it is important to consider not only the cost of performing that action, but the cost of all future actions that follow from the current state. This is difficult because the number of future states grows exponentially the farther away the current state is from the goal state. The Bellman equation, which specifies the cost to move optimally from the current state \mathbf{x} to the target state, captures this problem by defining the ‘cost-to-go’ function $v(\mathbf{x})$:

$$v(\mathbf{x}) = \min_{\mathbf{u}} \{ \ell(\mathbf{x}, \mathbf{u}) + E_{\mathbf{x}' \sim p(\cdot|\mathbf{x}, \mathbf{u})} [v(\mathbf{x}')] \}, \quad (1)$$

where $\ell(\mathbf{x}, \mathbf{u})$ is the immediate cost of performing the action \mathbf{u} from the state \mathbf{x} and $E_{\mathbf{x}' \sim p(\cdot|\mathbf{x}, \mathbf{u})} [v(\mathbf{x}')]$ is the expected cost-to-go at the next state, \mathbf{x}' . The expectation is taken with respect to the transition probability distribution over all next states, $p(\cdot|\mathbf{x}, \mathbf{u})$, given state \mathbf{x} and action \mathbf{u} . The notation ‘ $\mathbf{x}' \sim$ ’ is used to explicitly indicate that \mathbf{x}' is a random variable whose values are distributed according to the specified probability density function. Once the cost-to-go is computed for the current state, the minimum (over actions \mathbf{u}) is chosen, which selects the optimal action.

Critically, while the Bellman equation *defines* the cost-to-go function, it does not specify how to *solve* for it, and this has proven quite difficult to do. Most solutions have been found only for the case where the state transitions are linear and the cost function is quadratic (and depends only on the state and action). Methods such as dynamic programming, reinforcement learning and the iterative linear quadratic gaussian (iLQG) techniques have been used to solve

the Bellman equation for problems falling outside this class. All of these methods are based on identifying progressively closer approximations to the optimal control signal across multiple trials. A method that quickly solves for optimal controllers that overcome such limitations has been recently suggested [85].

2.1.2. Linear Bellman controllers. It has been shown that it is possible to solve the Bellman equation efficiently using a linear form for problems of a specific structure [85], specified below. To convert the Bellman equation to this linear form, we first need to change the standard characterization of action. Instead of considering actions as discrete events dictating system movement through actions such as ‘go-left’ or ‘go-right’, which are interpreted and then applied to the system’s state transition probabilities, the controller is allowed to alter the transition probabilities across potential next states directly.

Under this characterization, the cost of an action is defined as the difference between the transition probabilities without a control signal (the passive dynamics of the system), $p(\mathbf{x}'|\mathbf{x})$, and the transition probability distribution resulting from applying a control signal, $p(\mathbf{x}'|\mathbf{x}, \mathbf{u}) = u(\mathbf{x}'|\mathbf{x})$. Specifically, the Kullback–Leibler (KL) divergence is used to measure this difference, which gives a cost function

$$\begin{aligned} \ell(\mathbf{x}, \mathbf{u}) &= q(\mathbf{x}) + KL(u(\cdot|\mathbf{x})||p(\cdot|\mathbf{x})) \\ &= q(\mathbf{x}) + E_{\mathbf{x}' \sim p(\cdot|\mathbf{x}, \mathbf{u})} \left[\log \frac{u(\mathbf{x}'|\mathbf{x})}{p(\mathbf{x}'|\mathbf{x})} \right], \end{aligned} \quad (2)$$

where $q(\mathbf{x})$ is the cost of remaining in the state \mathbf{x} . To make the KL divergence well defined, a constraint must be applied to the control signal, specifically $u(\mathbf{x}'|\mathbf{x}) = 0$ whenever $p(\mathbf{x}'|\mathbf{x}) = 0$, which means that the controller cannot move the system anywhere outside the distribution of possible next states under passive dynamics. This also ensures that the controller does not issue a command requiring physically impossible state transitions to the system.

To rewrite the equation more simply, we define a desirability function

$$z(\mathbf{x}) = \exp(-v(\mathbf{x})) \quad (3)$$

which will be high when the cost of a state is low, and low when the cost of a state is high. As shown in the appendix, this allows us to specify the optimal transition probabilities u^* as

$$u^*(\mathbf{x}'|\mathbf{x}) = \frac{p(\mathbf{x}'|\mathbf{x})z(\mathbf{x}')}{\mathcal{G}[z](\mathbf{x})}, \quad (4)$$

where $\mathcal{G}[z](\mathbf{x})$ is the necessary normalization term.

The Bellman equation can now be simplified by substituting in u^* , accounting for the normalization term, \mathcal{G} , and exponentiating, to give

$$z(\mathbf{x}') = \exp(-q(\mathbf{x}))\mathcal{G}[z](\mathbf{x}). \quad (5)$$

Enumerating the set of states from 1 to n , representing $z(\mathbf{x})$ and $q(\mathbf{x})$ with the n -dimensional column vectors \mathbf{z} and \mathbf{q} , and $p(\mathbf{x}'|\mathbf{x})$ with the n -by- n matrix \mathbf{P} , where row index corresponds

to \mathbf{x} and column index to \mathbf{x}' , we can rewrite equation (5) into an eigenvector problem

$$\mathbf{z} = \mathbf{M}\mathbf{z}, \quad (6)$$

where $\mathbf{M} = \exp(\text{diag}(-\mathbf{q})\mathbf{P})$, ‘exp’ is applied element-wise and ‘diag’ transforms the vectors into diagonal matrices. This problem can now be solved using standard eigenvector methods, or through specially developed methods such as Z-iteration, described in [85].

Conceptually, linear Bellman controllers (LBCs) are important because they demonstrate how large, parallel systems like the brain may be able to quickly compute optimal control signals for complex control problems. Notably, we have said little about how the cost function is generated, which allows much flexibility in specifying in what way the control signal is optimized. Speed constraints, accuracy constraints, etc may be included naturally into such a formulation. In addition, because the controller picks an optimal set of transition probabilities, the exact control signal provided may change across different trials, as seen in biological systems. We consider other biological constraints in more detail later, in subsection 3.2. For present purposes, we take it that LBCs represent a solution to the general problem of optimal control which has the potential to be biologically plausible, addressing the first of the four control challenges we introduced.

2.2. Compositionality of motor commands

The second control challenge we identified is the generalization of learned control signals to novel circumstances. Learning movements that effectively and efficiently complete various goals can be a very costly process, in terms of both computational power and time. It is therefore desirable to be able to generalize learned movements to unlearned but similar movements. However, identifying a good procedure for combining a set of learned movements, or ‘components’, to effect a desired trajectory has long posed a significant challenge. Recent developments applying LBCs, however, have provided a means of quickly and cheaply creating optimal control signals from previously learned optimal movements.

Specifically, compositionality of commands specifies a method for weighting and summing movement components by expressing movements in terms of quickly calculated state desirabilities described by equation (3). Once such a desirability is determined, the LBC determines a corresponding control law that specifies how to move from a start state such that the target is reached optimally. The cost resides in computing such control laws. However, if we can decompose such functions into a set of learned, but widely used basis functions, computation of the control laws can be simplified.

Thus, using a established set of basis movement components that gives rise to a wide variety of desirability functions and control laws can allow any novel movement problem to be restated as a problem of recreating the corresponding desirability function through a weighted summation of the available component functions. A determined set of weights can then immediately be applied

to find corresponding control laws to create an optimal control signal for the new movement. In this way, knowledge of similar movements can aid in the execution of novel actions.

Formally, the compositionality of optimal commands can be described as follows. Let K be a collection of optimal control problems with the same dynamics, the same cost function and defined over the same set of system states, where the only difference between the problems is the target. Let $z_k(\mathbf{x})$ be the desirability function for the problem k , and let $u_k^*(\mathbf{x})$ be the corresponding optimal control law. The K problems for which a $u^*(\mathbf{x})$ is already defined are termed *component* problems, and their optimal control laws can be used to construct laws for new problems that fall into the same class.

For a given desirability function, $z(\mathbf{x})$, the goal is to find weights w_k such that the desirability function can be written as

$$z(\mathbf{x}) = \sum_{k=1}^K w_k z_k(\mathbf{x}). \quad (7)$$

The weights w_k represent the ‘compatibility’ between the control objective of the component problems, $z_k(\mathbf{x})$, and the control objective of the composite problem, $z(\mathbf{x})$. These weights can be found by projecting each component onto the new desirability function.

The composite control law u^* can also be expressed as a state-dependent combination of control laws u_k^* , using component problems k where the desirability of the state x is greater than zero, or $z_k(\mathbf{x}) > 0$. Starting with equation (4), substituting equation (7), and introducing $\mathcal{G}[z](\mathbf{x})$ in both the denominator and numerator gives

$$u^*(\mathbf{x}'|\mathbf{x}) = \sum_k \frac{w_k \mathcal{G}[z_k](\mathbf{x})}{\sum_s w_s \mathcal{G}[z_s](\mathbf{x})} \frac{p(\mathbf{x}'|\mathbf{x}) z_k(\mathbf{x}')}{\mathcal{G}[z_k](\mathbf{x})}, \quad (8)$$

where the normalization term for the optimal action is the sum (over s) of the normalization terms of all the components. In this form, the first term is the state-dependent mixture weight, which can be denoted $m_k(\mathbf{x})$, and the second term is distribution over optimal actions, i.e. u_k^* . Rewriting equation (8) with these terms, the optimal control law for composite problems becomes

$$u^*(\cdot|\mathbf{x}) = \sum_k m_k(\mathbf{x}) u_k^*(\cdot|\mathbf{x}). \quad (9)$$

For more details, please see the appendix.

This form of the control law is useful because it demonstrates how optimal control components can be reused for novel actions. Specifically, the control components can be reused by rapidly computing $m_k(\mathbf{x})$, which is a simple function of the weights identified by decomposing the desirability function. In general, such an approach shows how a control system can take advantage of past solutions when similar, though possibly novel, control problems are encountered. The efficient computation of such required functions by summing over a basis has been shown to be biologically plausible in several past applications [68, 19, 64, 42, 23]. In systems operating under variable dynamics or ever-altering environments, as animals do, it is necessary that the basis

set of actions be extensive. It has been suggested that such a set of basis functions is exactly what humans develop during their early years [1, 57, 67], and there has been much promising research into analogous techniques in robotics for the development of effective action sets for robots [13, 18, 66].

2.3. Hierarchical control systems

The third challenge for control systems that we consider is the ‘curse of dimensionality’. The curse of dimensionality refers to the exponentially increasing difficulty of solving for optimal trajectories for a system as the DOF increase [79]. Hierarchical system architectures can provide a method of effectively addressing this increased complexity by reworking the system representation into a lower dimensional space, where optimal trajectories and control signals can be efficiently planned. These low-dimensional control signals can then be used to define the commands in the higher dimensional spaces. If the mapping between the lower and higher dimensional spaces is (even approximately) known, this approach affords an effective means of controlling highly complex dynamical systems in an efficient manner.

In general, a control hierarchy is built in a pyramidal fashion, where the highest level has the fewest dimensions, and the lowest level has the most. Once a high-level signal has been identified, it is used to constrain the lower levels of the hierarchy, thus restricting the range of the low-level state space to be searched for a solution. Even constrained, however, the additional DOFs in the lower level system space may allow a multitude of solutions that respect the higher level command, due to the possibility of redundant configurations. For example, when controlling an arm to reach to a specific target we may identify a high-level path for the tip of the finger in three dimensions. However, there are endless possible shoulder, elbow and wrist configurations that will realize a given finger tip trajectory. To determine the appropriate optimal solution for every level of a hierarchy, each level of the hierarchy must hold a cost function which describes preferred movement in that space.

Formally, a hierarchical system can be described as follows. As in a standard dynamical system, the plant, or underlying system, dynamics are characterized with a low-level system representation

$$\dot{\mathbf{x}}(t) = \mathbf{a}(\mathbf{x}(t)) + B(\mathbf{x}(t))\mathbf{u}(t), \quad (10)$$

where $\mathbf{x} \in \mathbb{R}^{n_x}$ is the system state vector, $\mathbf{u} \in \mathbb{R}^{n_u}$ is the control vector, $\mathbf{a}(\mathbf{x})$ are the (possibly nonlinear) passive dynamics and $B(\mathbf{x})\mathbf{u}$ are the control-dependent dynamics. This is a control-affine system formulation, standard in control theory, where the system dynamics are linear in terms of the control signal, \mathbf{u} , but nonlinear with respect to the state, \mathbf{x} .

We can then define the high-level state, \mathbf{y} , as a function of the low-level plant state:

$$\mathbf{y} = \mathbf{h}(\mathbf{x}), \quad (11)$$

where \mathbf{h} transforms the low-level system state \mathbf{x} into a reduced representation, \mathbf{y} . It is important to identify a high-level state space that captures the salient features of the low-level

dynamics, but has fewer dimensions in both the state and control space.

Let us define the high-level dynamics as

$$\dot{\mathbf{y}}(t) = \mathbf{f}(\mathbf{y}(t)) + G(\mathbf{y}(t))\mathbf{v}(t), \quad (12)$$

where \mathbf{f} and G are the passive and control-dependent dynamics on the high level, and \mathbf{v} is the high-level control signal. To get the high-level transformation of the actual dynamics of the low-level system, we differentiate equation (11):

$$\dot{\mathbf{y}} = H(\mathbf{x})(\mathbf{a}(\mathbf{x}) + B(\mathbf{x})\mathbf{u}), \quad (13)$$

where $H(\mathbf{x}) = \partial h(\mathbf{x})/\partial \mathbf{x}$ is the Jacobian of the function h . By equating the actual and desired high-level dynamics

$$H(\mathbf{x})\mathbf{a}(\mathbf{x}) + H(\mathbf{x})B(\mathbf{x})\mathbf{u} = \mathbf{f}(\mathbf{y}) + G(\mathbf{y})\mathbf{v}, \quad (14)$$

the dynamics on both sides is defined. If we assume that a high-level control signal \mathbf{v} has been issued, the goal of the low-level control signal $\mathbf{u}(\mathbf{x}, \mathbf{v})$ is to match the actual (equation (13)) and desired (equation (12)) high-level system dynamics.

There are two standard methods for updating the high-level passive dynamics: explicit and implicit modeling. In explicit modeling, analogous to open-loop control, the high-level has a model of the low-level system dynamics for *simulating* the state of the plant. The simulation feedback is used to update the high-level control signal, which stresses the importance of having an accurate low-level dynamics model, as no actual state feedback is received from the lower level.

In implicit modeling the high-level control does not model the low-level system dynamics, rather it has online access to the transformed system plant state, $H(\mathbf{x})\mathbf{a}(\mathbf{x})$. With no discrepancy between $\mathbf{f}(\mathbf{y})$ and $H(\mathbf{x})\mathbf{a}(\mathbf{x})$ in this approach, equation (14) becomes

$$H(\mathbf{x})B(\mathbf{x})\mathbf{u} = G(\mathbf{y})\mathbf{v}. \quad (15)$$

The implicit modeling method allows the high-level controller to take advantage of the plant's exact passive dynamics while still operating on a lower dimensional system.

By constructing similar links between many levels of a hierarchy, a very high-dimensional plant may be controlled in a reasonably low-dimensional control space. Issues of redundancy are thus taken care of at each level of the hierarchy, through the use of level-specific control constraints. This provides an effective means of handling the curse of dimensionality, while also matching the known structure of biological motor systems, which we discuss in more detail shortly. First, however, we consider the fourth challenge for optimal control—noise.

2.4. LQR and Kalman filters

Noise in a system can be very damaging to the effectiveness of a controller, by both interfering with control signals causing the system actuators to receive an incorrect command, and by pervading the sensory feedback signal. Sensory feedback can be distorted either at the sensors themselves, or during transmission to the controller. In any case, such distortions mislead the controller as to the current state of the system.

Perhaps the best-known method for accounting for such expected distortions is the use of LQR. LQR accounts for the effects of noise interference on efferent signals by modeling the system noise on the pathway, and producing compensatory signals to negate its effect. Kalman filters play an analogous role on sensory feedback signals, attempting to account for and cancel out the noise resulting from sensor and afferent pathway distortions. Both LQRs and Kalman filters are dependent on having accurate models of the system dynamics, such that unwanted features can be attributed to noise rather than improper control signals. In addition, both methods assume Gaussian noise and linear dynamics.

Formally, the effects the desired control signal, \mathbf{u}_0 , will have on the system state, \mathbf{x}_0 , are modeled, as well as the system output, \mathbf{y} , in the ideal situation of no noise or external forces being applied. The actual system state $\mathbf{x}(t)$ at time t can then be compared to the desired system state \mathbf{x}_0 and used to define the error, $\delta\mathbf{x}(t) = \mathbf{x}(t) - \mathbf{x}_0$, termed the state perturbation vector. Once $\delta\mathbf{x}(t)$ is defined, a control correction vector $\delta\mathbf{u}$ can be generated using a linear approximation of the system [59, 5]. This correction signal can then be applied to account for the noise in the system and allow accurate implementation of the desired commands in the face of noise. We do not derive LQR or Kalman filters here, as they can be found in many standard textbooks.

While we suspect that the brain uses more sophisticated methods for handling noise than the linear ones mentioned here, we have found little biological evidence that suggests which such methods may be most appropriate. For present purposes, we adopt these methods for simplicity, but feel that this is one respect in which the NOCH stands to be significantly improved.

3. Neural optimal control hierarchy

To this point, we have discussed several quantitative methods for generating optimal control signals. We have chosen these specific methods because of their biological plausibility, though we have not yet made the case that they are biologically relevant. In this section, we describe specifically how we believe these methods can be employed to capture the biological functions of the main brain areas involved in motor control. We demonstrate how this mapping is effective at providing behavioral and neural level explanations in section 5.

The NOCH framework is a product of integrating control theory with neuroscientific evidence related to motor control. In developing the NOCH we have focused on assigning function to neural structures such that they are consistent with results from current experimental studies and overall system functionality. In addition, we have attempted to identify underlying computational principles that capture the role of the various neural structures in the motor system. In short, the NOCH identifies multiple components that carry out processes in parallel, and exploits cortical circuit structure and specific neural mechanisms to underwrite these processes. However, any such characterization will remain hypothetical for the

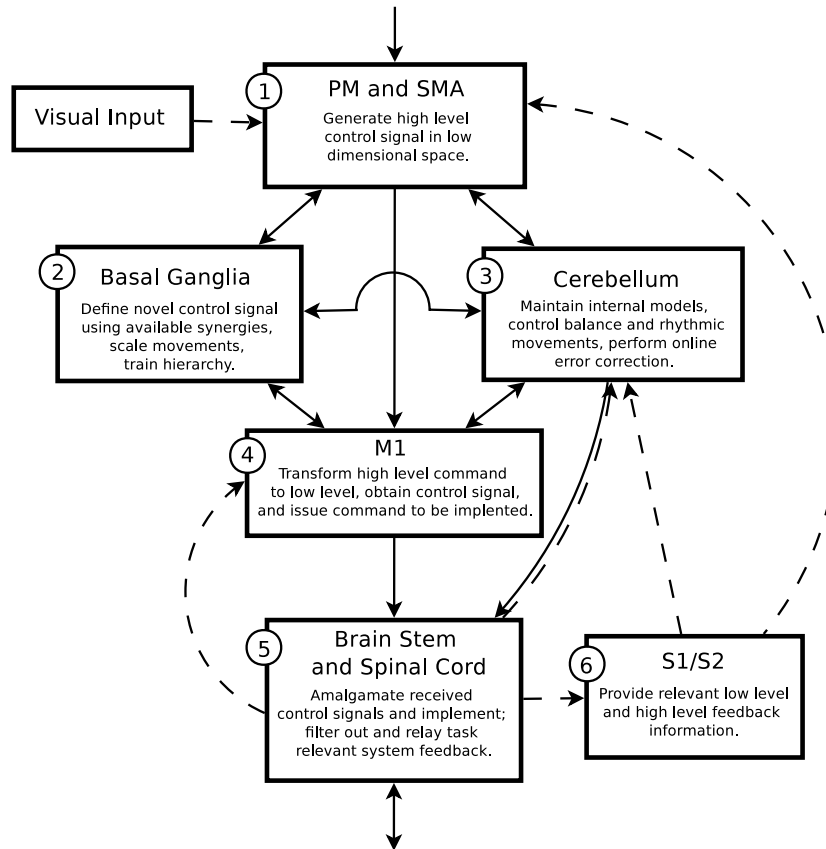


Figure 1. The NOCH framework. This diagram embodies a high-level description of the central hypotheses that comprise the neural optimal control hierarchy (NOCH). The numbering on this figure is used to aid description, and does not indicate sequential information flow. See the text for details.

foreseeable future, as the neuroscientific evidence is seldom irrefutable regarding neural function.

To be clear, we take it that the NOCH is an evolving framework which currently over-simplifies the functions of many of the relevant anatomical areas. Given the lack of alternative frameworks, however, these simplifications do not prevent the NOCH from usefully contributing to our understanding of how control theoretic functions map to the neural substrate. In the remainder of this section, we describe each of the main elements of the NOCH, appealing to both control theoretical and neuroscientific characterizations of motor control. We begin with a brief description of all the elements of the NOCH and then discuss central aspects of the framework in more detail in subsequent sections.

3.1. NOCH: a brief summary

A block diagram of the NOCH framework is displayed in figure 1. The numbering on this figure is used to aid description, and does not indicate sequential information flow. For additional details see [20].

3.1.1. Premotor cortex (PM) and the supplementary motor area (SMA). The premotor cortex (PM) and the supplementary motor area (SMA) integrate sensory information and specify target(s) in a low-dimensional, end-effector agnostic, and scale-free space. End-effector agnostic

means that at this stage, no lower level dynamics for any limbs or body segments that might carry out the action are considered. It is strictly a high-level space, which may specify control signals in terms of, for example, 3D end-point position. Scale-free refers to the fact that solutions found for LBCs (see section 2.1.2) for optimal movement in an area of a particular size can be subsequently manipulated by rescaling, due to the fact that this high-level space is end-effector agnostic [20].

An example of the PM/SMA function in arm reaching begins with the planning of an optimal path from current hand position to target, which incorporates information from the environment, such as obstacle position. Previously learned motor components (i.e. synergies) are used as a basis, and linearly combined through weighted summation to compose the desired movement, as described in section 2.2. If the desired movement cannot be created from the available set of basis synergies, the system may explore new paths through space to determine a satisfactory trajectory. These areas act as the highest levels in a motor control hierarchy (see section 2.3) that proceeds through M1 and eventually to muscle activations.

3.1.2. Basal ganglia. Recently, the basal ganglia has been characterized as a winner-take-all (WTA) circuit [35], as responsible for scaling movements or providing an ‘energy vigor’ term [89], and as performing dimension reduction [6]. Spiking neuron implementations that employ the

same methods of neural simulation employed here (see section 4.2) have been used to construct a WTA circuit model that has strong matches to neural timing data [78], and can incorporate both movement scaling and dimension reduction [87]. Consequently, in the NOCH, the basal ganglia is taken to weight movement synergies for the generation of novel actions, perform dimension reduction to extract the salient features of a space for training cortical hierarchies and developing new synergies and scale movement during directed action.

3.1.3. Cerebellum. The cerebellum is widely regarded as an adaptation device, performing online error correction, and is thus often taken as the site for the storage of internal models [21, 7, 93, 48]. In the NOCH, the cerebellum is taken to play a similar role. While in the cortex movement synergies stored are for well-learned environments and situations operating under normal system dynamics, the cerebellum stores synergies that allow the system to adapt to new environments and dynamics, and learn or recall situations to adapt current movements appropriately based on sensory feedback. Additionally, the cerebellum plays a central role in correcting for noise and other perturbations, correcting movement errors to bring the system to target states as specified by higher level controllers. Consequently, the cerebellum is also responsible for control of automatic balance and rhythmic movements, such as locomotion.

3.1.4. Primary motor cortex. In the NOCH, the primary motor cortex (M1) is understood as containing the lower levels of a hierarchical control system. The primary motor cortex thus acts as a hierarchy that accepts high-level commands, such as end-effector force in 3D space, from the PM and SMA, and translates them through hierarchical levels to muscle activation signals. The main functional role of this hierarchy is to map high-level control signals to low-level muscle activations in an efficient manner, as specified in section 2.3, allowing the PM and SMA to develop control signals in a reduced, lower dimensional space. The synergies at each level are assumed to change over time, as skills are developed or lost.

3.1.5. Brain stem and spinal cord. For our purposes, the brain stem acts as a gateway for efferent motor command and afferent sensory input pathways. Here, descending commands from different neural systems in the NOCH can be combined and passed on to the spinal neural circuits and motor neurons for execution.

3.1.6. Sensory cortices (S1/S2). The primary sensory cortex (S1) is used for sensory feedback amalgamation and processing in the NOCH, serving to produce multi-modal feedback which is then relayed to the motor areas such as M1 and the cerebellum. Both of these systems are taken to work in lower level, higher dimensional spaces, and are thus in a position to incorporate appropriate feedback signals into the working motor plan. The secondary sensory cortex (S2) is responsible for transforming the information from the primary sensory cortex into high-level sensory feedback information,

as described in section 2.3, which is then relayed to the high levels of the M1 hierarchy, as well as PM and SMA. Additionally, S1 and S2 perform a noise filtering on sensory feedback, combining different types of feedback to arrive at the most reliable prediction of body and environment state, analogous to the function of a Kalman filter (see section 2.4).

In the remaining subsections, we provide further discussion and justification for the characterizations presented in the preceding brief outline.

3.2. Hierarchies in the motor control system

One notable feature of brain structure that has proven broadly salient to neuroscientists is its hierarchical nature. The best-known example of this structure in neuroscience is the visual hierarchy. For object recognition, this hierarchy begins with the retina, and proceeds through thalamus to visual areas V1 (primary visual cortex), V2 (secondary visual cortex), V4 and IT (inferotemporal cortex) [24].

In many hierarchical models of visual cortex, each higher level attempts to build a model of the level below it. Taken together, the levels define a model of the original input data [38]. This kind of hierarchical structure naturally allows the progressive generation of more complex features at higher levels, and progressively captures higher order relationships in the data. Furthermore, these kinds of models lead to relations between hierarchical levels that are reminiscent of the variety of neural connectivity observed in cortex: feedforward, feedback and recurrent (interlayer) connections are all essential. Such models have also been shown to generate sensitivities to the input data that look like the tuning curves seen in visual cortex when constructing models of natural images [51, 63].

The motor system is also commonly taken to be hierarchical [82, 37]. There is a strong case to be made that the motor hierarchy can be considered the dual of the visual hierarchy [82]. Typically, however, we think of information as flowing down rather than up the motor hierarchy. This duality is displayed in figure 2. As an example of information flow through a three-layer implementation of this system, suppose you would like to move your hand toward a target object. First, the goal state is identified and passed to the cortical hierarchy. This information arrives at the top layer of the hierarchy, which operates in 3D end-effector space (the PM/SMA). The controller at this level identifies the direction the hand needs to move to reach the target optimally, and generates the 3D force control signal to move the hand along that path. This 3D force signal, however, does not tell the system what torques to apply to move the arm joints appropriately. This is the job of the controller of the second layer of the hierarchy (in M1). The lowest level of the hierarchy is then responsible for taking the joint torque control signal of the second level and generating the muscle activation commands to effect the specified joint movement, which is then applied to the arm itself (through M1 and cerebellum). While useful for explanatory purposes, we do not expect the hierarchy to pick out these particular hierarchical levels. Instead, we expect the actual hierarchy in M1 to be defined over spaces spanned by motor synergy bases that are useful for movement. Since we do not have a

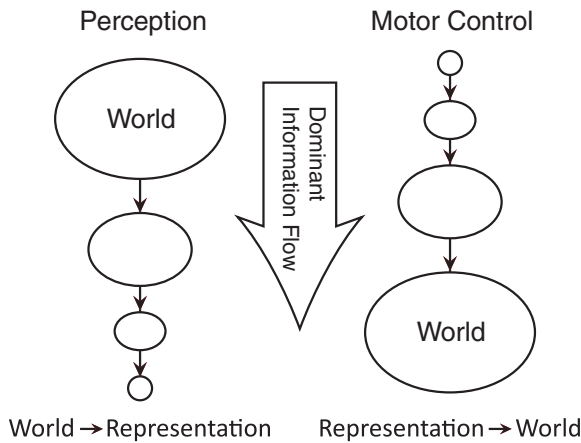


Figure 2. An illustration of the dual relationship between motor control and perception. In both cases, there is strong evidence for a hierarchical organization. Computational descriptions of both processes can consistently identify lower dimensional spaces and a more complex representational basis, as we move higher in the hierarchy. However, the typically characterized dominant direction of information flow is reversed. Hierarchical motor control models are much less common than hierarchical perceptual models, but the duality is striking both conceptually and mathematically [82].

description of such spaces, we use standard identifiable control variables as a stand-in in our subsequent models.

In short, as the motor command becomes more and more specific to the particular circumstance (e.g. including the particular part of the body that is being moved, the current orientation of the body, the medium through which the body is moving, etc), lower level controllers are recruited to determine the appropriate control signal for the next level as the command descends the hierarchy. Ultimately, this results in activity in motor neurons that cause muscles to contract executing the desired motion.

In the NOCH, the hierarchical transformations from the top level to the bottom are divided into a series of stages. While mathematically it is possible to amalgamate all of the transformations, and reduce the hierarchy to a two-level structure, multiple state spaces allow the control system to easily apply a number of different types of constraints on motor actions. For example, consider a system where the top level of the hierarchy represents movement in a 3-DOF end-effector space, the second level represents 7-DOF joint space and the bottom level represents the 27-DOF muscle space of the arm. With such a configuration, it is straightforward to specify constraints in joint space (such as holding the elbow angle constant while reaching); the cost function of the second layer is simply adjusted to make certain movements highly expensive. In a two-level hierarchy, application of such a constraint is significantly less intuitive, as it must be done in either end-effector or muscle space.

Importantly, from a theoretical perspective, the decomposition of the problem into a hierarchy plays a similar computational role as in the visual system. Specifically, this decomposition allows nonlinearities to be interleaved with linear mappings to capture sophisticated relationships between the highest and lowest levels of analysis in a resource efficient

manner. In both cases, the nonlinear mappings between linear representational spaces (which themselves may be defined over nonlinear bases) allow sophisticated transformations to be performed at each level of the hierarchy.

Given the utility of motor hierarchies, it is natural to ask what kinds of neural representations we would expect at each level of the hierarchy. In the context of our previous discussion (e.g. section 2.2), this amounts to asking for the components that form the basis for the control space, which can be summed to create control signals. A natural answer to this question is the now pervasive concept of a ‘motor synergy’.

A synergy is a combination of movements across space-time. Synergies give rise to two very useful features for motor controllers, providing both movement components that can be used to simplify the creation of more complex movements, and a translation between lower and higher dimensional representations. There is good evidence that the motor system optimizes the control of synergies, rather than the actuation of individual muscles, both from experimental studies [8] and model simulations [44].

Consequently, we take the fundamental organization of the NOCH to be hierarchical, and the basic units of representation to be motor synergies. Neither of these assumptions are novel. However, in the context of the entire framework, they provide a general means of relating optimal control methods and the neural representation underlying the NOCH described in section 4.2.

3.3. The motor cortices: M1, PM and SMA

Numerous studies have linked the neural activity of the motor cortices to a vast array of different movement parameters, ranging from arm position to visual target location to joint configuration [84]. In the NOCH framework, the descending output signals generated through the cortical hierarchy are in terms of muscle activation, operating either directly on motor neurons, or by driving or modulating the inter-neurons and neural circuits of the spinal cord. Muscle activation as cortical output gives rise to all of the correlations observed in the above-mentioned studies, and can be justified from an evolutionary vantage as well [84].

The assumed hierarchical structure of the motor cortices allows for abstraction away from this intrinsically high-dimensional space to a lower dimensional representation such that the ability to perform effective, efficient control is not lost, and a mapping down to muscle activations is still available. The hierarchical structure of the motor cortices is divided into two main elements in the NOCH, the premotor cortex (PM) and supplementary motor areas (SMA), which generate an end-effector agnostic, scale-free, high-level control signal, and the primary motor cortex (M1), which receives the output from the PM/SMA, and transforms the signal into a limb-specific set of muscle activation commands to carry out the specified high-level control signal.

The functional division of the motor cortices in this manner allows for the generation of a high-level control signal that can be executed by any available or desired end-effector without reformulating the high-level trajectory. This

type of high-level control matches experimental observations noting that high-level path planning, such as handwriting style, are preserved regardless of the end-effector chosen for implementation [93].

3.4. The cerebellum

In the NOCH, the cerebellum is modeled as three anatomical areas, each responsible for a function: the intermediate spino-cerebellum for error correction, the lateral cerebro-cerebellum for maintenance of internal models and the vermis for control of automatic and rhythmic action. This functional assignment arises from examining neuroscientific pathway tracing studies of the input and output cerebellar connections, the neuroanatomical structure of the cerebellum and the required control theoretic functionality. The cerebellum constitutes only 10% of the total volume of the brain, but contains more than half of its neurons [31]. Additionally, the neural architecture of the cerebellum is remarkably uniform, with a very regular structuring repeated throughout the area [33]. These features suggest that the cerebellum is ideal for massively parallel computations, and hence the NOCH employs it to meet the demands of robustness, generalizability and feedforward predictive modeling.

When the execution of a motor command results in unexpected movement, or otherwise does not effect the desired outcome, the cerebellum is responsible for providing corrective signals to the system. These errors break down into two main categories, caused by either unexpected external forces, or by inaccurate models of the system dynamics. In both cases, a short-term solution can be enacted through error integration [3], analogous to the error correction signal an LQR mechanism provides (see section 2.4). Indeed, biological modeling simulations support this system as a site for a neural implementation of error correction [33, 50]. The intermediate zones of the cerebellum are part of the spino-cerebellum, named for the area's strong connectivity with the spinal cord sensory feedback pathways. Neural activity is relayed from the intermediate cerebellum through the interposed nuclei out to the motor cortices, specifically to areas which influence the neural activity of the lateral spinal tracts, which house motor neurons responsible for distal muscle control, suggesting it as a likely candidate for motor error correction based on system feedback.

This functionality is supplemented by the second main responsibility of the cerebellum, the storage and maintenance of internal models. These include predictive models that can be used for feedforward control, which are important for explicit high-level control (see section 2.3). In addition, such models can be employed to adapt motor actions under a given set of environmental conditions. To perform these functions, the cerebellum requires an efferent copy of the motor plans that are sent to descending pathway for execution. In combination with the massive sensory feedback projections the cerebellum receives, corrective signals can be generated, and tested on the working system. These can then be used to correct cortical motor synergies or develop internal models for context-dependent adaptation, in the case where system error

is consistent and predictable. These functional requirements match the input/output structure of the cerebro-cerebellum, located in the lateral cerebellum, which has massive reciprocal connections with the cerebral cortex, making it an ideal site for the integration of internal model signals into generated motor commands, as well as the training of motor synergies for persistent errors.

The third major functional responsibility assigned to the cerebellum is the control of balance and rhythmic movements. Damage to the cerebellum has been associated with impaired locomotor performance and balance, and it is proposed to play a role in the generation of rhythmic movements, dynamic regulation of equilibrium and adaptation of posture and locomotion through practice [60]. The assigned biological correlate of this functionality is the vermis in the cerebellum, which maintains strong spinal feedback connections and projects through the fastigial nuclei to systems influencing the activity of the medial tracts of the spinal cord. Additionally, experimental studies have confirmed vermis activity to be strongly associated with automatic and rhythmic movements, such as balance and locomotion.

3.5. The relation between cerebellar and cortical control

The motor cortices became renowned for being deeply involved in the control of motor actions from early stimulation experiments presented in [65] in 1950. While it is indisputable that the motor cortices play an integral part in motor control, it is also important to consider that they appeared relatively late in the evolutionary development of the brain. There are many species that have little-to-no cortex, but are very skilled at carrying out motor actions arising from seemingly simple interactions of spinal neural circuits [45, 43, 40].

For example, animals such as the salamander or lamprey are capable of performing very efficient and effective locomotive movements, which can be characterized solely through the activity of spinal motor nuclei and circuits [16]. As the ability of animals to produce more complex, dexterous movements increases across species, the respective size of their cortices, and cortical areas dedicated to motor control, as well as the number of cortico-spinal projections, increases [55, 11]. These observations provide some clues about the respective roles of the long-standing cerebellum versus the relatively recent motor cortices. The characterization of cerebellar and cortical motor function in the NOCH framework is based on the above observations, the connectivity of each to spinal neurons and circuitry, biological modeling results and localized damage/lesion studies.

For instance, pathways from the cerebellum relay to the spinal cord through the fastigial nucleus and project to medial spinal tracts, where motor neurons and interneuron circuits with broad projections to proximal muscle groups are housed [10]. The resulting co-activation of motor neurons offers a type of intrinsic muscle synergy structure to the cerebellum, coordinating the activation of sets of proximal muscles [32], and acts to provide a base set of features in muscle space that are useful for simplifying the creation of complex actions (see section 2.2). Dense connectivity with the brain stem

[26], strong sensory feedback from ascending spinal pathways [10], a recognized error correction functionality [31] and results from damage studies [60] suggest that the cerebellum is a centre for control of larger proximal actions, specifically balance and rhythmic movements such as locomotion. These features integrate well with the characterization of cortical–cerebellar interactions as being one of more controlled and volitional action from cortex being corrected and supported by cerebellum.

The motor cortex holds strong connections to spinal neurons controlling proximal muscles, but is also responsible for the vast majority of projections to the lateral spinal tracts [70], where motor neurons controlling distal muscles are housed. Notably, the control effected by these neurons is far more localized than that of the medial motor neurons controlling proximal muscle activation [32]. The information processing ability of cortical hierarchies discussed in the last section, along with the projections to more individually actuated distal muscles, makes the motor cortices a natural complement to the cerebellar processes, allowing controllers to operate in abstracted muscle spaces, enabling efficient, dexterous control of high-level system parameters.

In the NOCH framework, the control provided by the motor cortices is tuned to operate under a standard set of environmental and operating conditions. The negotiation of external forces and load or system parameter changes is provided by additional processing centers in the cerebellum. The cerebellum has long been assumed to provide storage of internal models used for feedforward/predictive control [93, 48, 21], ideal for the adaptation of motor commands trained for general control problems. When the system dynamics or environment are altered and performance declines, the control signals from the motor cortices are dynamically adapted through the incorporation of cerebellar output, and the resulting amalgamation is sent to the descending pathways. The function provided by the cerebellum in this case is analogous to that of the LQR, in that it attempts to account for deviations of the system from the ideal movement trajectory (see section 2.4).

Additionally, the corrective signal from the cerebellum serves to train the motor cortex if applied for an extended period of time. This is the case where the environmental or system dynamics have been altered, such as when one is at sea for a long time and adapts to the swaying of the boat, or when the efficacy of a limb is reduced during a long-term injury. This process supports modifying stored cortical synergies which may be more efficient than constantly applying the same adaptation to descending control signals under such circumstances.

3.6. The basal ganglia

The basal ganglia have long been implicated in the processes of action selection [28, 2, 58], with clear application to motor control. A recent spiking neuron implementation [78] of a biologically plausible model of the basal ganglia [35] provides a neural implementation of a winner-take-all circuit. This implementation is able to quickly determine a winner

among large numbers of input, while scaling input values and matching a wide variety of neural data. Results from this implementation have shown that with slightly altered dynamics this circuit can perform thresholding and similarity evaluation. In the NOCH, the action selection functionality of such a circuit is employed for composing novel actions from movement synergies. Specifically, the basal ganglia is assumed to compute the similarity between a desired action and the available synergies, and generate a set of normalized weights that determines the composition of the synergies and scales the movement. Each of these functions is implemented by these past models of basal ganglia.

Interestingly, these same models can be understood as performing dimensionality reduction [87]. As discussed above, optimal control is more efficient in low-dimensional spaces. However, to move from the high-dimensional muscle space to a lower dimensional synergy space for efficient control, a hierarchy must be established during development. Understanding the basal ganglia as performing dimensionality reduction lends it a natural role to play in identifying prominent features of a space. Its broad cortical connectivity, central location and known role in training cortical connections [89] places it in an ideal position to train muscle synergies on the various cortical hierarchical levels, by identifying salient features of a space, and feeding them back to cortex.

Considerations consistent with this functionality are found in infant studies of ‘motor babbling’, which is thought to perform a similar function to verbal babbling; it supports building up a set of movement components that can be combined to create more complex actions [77, 66]. Initial motor actions would be in a very high-dimensional muscle space, though controlling a much lower dimensional action state space. Dimension reduction in the basal ganglia is highly useful for developing a hierarchy, training a low-dimensional analog of the higher dimensional space for efficient control.

This dimension reduction through development is taken to be employed until a hierarchy has been created where movements can be efficiently planned without leaving any critical DOFs unspecified on lower levels. The definition of a ‘critical DOF’ is task dependent, as some movements may be concerned only with hand position, while others may also constrain joint angles or muscle activation. This variety suggests that control may influence various levels of the hierarchy, depending on the type of action being carried out.

Notably, the proposed functions of the basal ganglia are consistent across levels of the motor cortex hierarchy. In addition, this functionality is important for development, but not critical to the system’s real-time operation. That is, basal ganglia under this characterization is not explicitly involved in action selection much of the time. This view is supported by lesion/damage studies in which subjects have had a bilateral pallidectomy (i.e. there are no output pathways from basal ganglia), but show no marked motor deficiencies, except those relating to the proper scaling of movement and performance of novel movement sequences [89].

In sum, the basal ganglia is proposed to be responsible for composing novel actions from motor synergies, scaling

movements and training synergies for the cortical hierarchy. While speculative, this characterization is supported by both experimental studies and model simulation results that suggest that the basal ganglia is largely responsible for these processes.

3.7. The NOCH: a summary

In the preceding two sections, we identified four major technical challenges for developing control systems and suggested quantitative methods for addressing each challenge. Then, more speculatively, we provided a high-level characterization of the mammalian motor control system, identifying plausible functions for the main anatomical components of the system. In doing so, we suggested how the various technical solutions introduced earlier mapped into this anatomical and functional characterization of the biological motor system.

While this discussion has remained largely abstract, and our understanding of the details of the neurobiological system is largely incomplete in many cases, we believe that proposing large-scale characterizations such as the NOCH are useful for directing future research. This is because such characterizations can help identify assumptions that must be made in order to complete a functional specification of the system. In doing so, it becomes clear where our theoretical and experimental understanding is the weakest. However, it may also be the case that various problems (e.g. delay) may be more simply characterized in the context of the entire system than if they are viewed in isolation.

In any case, the utility of such a view must be grounded in specific applications of its principles. To this end, in the remainder of this paper we present simulations based on the principles of the NOCH. First, we reproduce normal human reaching trajectories and velocity profiles with a two-level hierarchical control system, that includes motor cortices, basal ganglia and cerebellum. We then lesion the system to impair specific functionality of the model and compare the results with clinical data from studies of patients where corresponding neural structures have suffered physical trauma. Second, we employ the same controller in a neural simulation environment and examine the spiking patterns of neurons in the model's primary motor cortex, as it controls a 2D arm. We compare the responses of single neurons in the model with neural recordings from monkeys performing the same task. These two examples help elucidate the utility of the NOCH, and demonstrate methods of connecting the framework to both behavioral and neurophysiological results.

4. Simulation methods

In this section, we describe the details of the simulations whose results are presented in the next section. We have chosen two different kinds of simulations to demonstrate different applications of the NOCH. In the first instance, we provide a somewhat high-level simulation of arm control, to explore the functional contributions of various components of the NOCH. By performing lesioning experiments on this model, we are able to determine if the proposed NOCH mapping of control

function to anatomy captures gross functional constraints. In the second instance, we provide a detailed mapping of motor control signals in M1 to specific neural firing patterns during arm movement. This allows us to determine if the NOCH can be used to capture the details of highly nonlinear neuron responses with a simple mapping between functionally derived control signals and single spiking neurons.

4.1. Human arm reach

To reproduce human arm reaching movements, we developed a two-level hierarchical control structure that accepts target positions in Cartesian coordinates, and optimizes arm movement in the task space. We perform simulations of standard behavioral experiments by running the model through eight reaching movements from a constant initial position to target locations placed in a circle around the starting position of the hand. The control signal is generated through compositionality of movements, by defining a weighted summation on predefined 'learned movements', or movement components, calculated *a priori* using a LBC (see section 2.1.2).

This implementation thus combines the proposed functionality of the motor cortices, the basal ganglia and the cerebellum. Specifically, the high-level control signal in these simulations determines the desired application of force to the end-effector, and represents the output from the PM/SMA. This output is given as a weighted summation of synergies in generalized coordinates and must be scaled to fit the current operating space. This scaling represents basal ganglia and cerebellar synergy adaptation. The adapted signal is then transformed into torque signals through quadratic programming optimization, analogous to the function performed by the basal ganglia-cortical loop with M1. The basal ganglia element of the model thus performs a weighted action selection function to identify the final control signal routed through M1.

Formally, the lower level system state and plant are defined as the joint angles and velocities $\mathbf{x} = [\theta_1, \theta_2, \dot{\theta}_1, \dot{\theta}_2]^T$, with system dynamics

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}, \quad (16)$$

where $\mathbf{A} = \begin{bmatrix} 0 & \mathbf{I} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$, $\mathbf{B} = \begin{bmatrix} \mathbf{0} \\ \mathbf{I} \end{bmatrix}$ and $\mathbf{u} = [\tau_1, \tau_2]^T$, where τ_1 and τ_2 are the torques applied to the shoulder and elbow joints, respectively. Recall from section 2.3 that the system is assumed to be linear in control.

Following the process described in section 2.3, we continue by defining our high-level system. In this model the high-level system state will be the Cartesian coordinates of the end-effector, $\mathbf{y} = [p_x, p_y, \dot{p}_x, \dot{p}_y]^T$, relative to an absolute reference frame placed at the shoulder, with ideal system dynamics

$$\dot{\mathbf{y}} = \mathbf{G}\mathbf{y} + \mathbf{F}\mathbf{v}, \quad (17)$$

where \mathbf{G} and \mathbf{F} are the same as \mathbf{A} and \mathbf{B} , and $\mathbf{v} = \begin{bmatrix} f_x \\ f_y \end{bmatrix}$, where f_x and f_y are the forces applied to the end-effector.

The transformation function between the two levels, \mathbf{h} , such that $\mathbf{y} = \mathbf{h}(\mathbf{x})$ is defined as

$$\mathbf{h}(\mathbf{x}) = \begin{bmatrix} \cos(\theta_1)L_1 + \cos(\theta_1 + \theta_2)L_2 \\ \sin(\theta_1)L_1 + \sin(\theta_1 + \theta_2)L_2 \\ J(\theta)\theta \end{bmatrix}, \quad (18)$$

where L_1 and L_2 are the lengths of the upper and lower arm segments, respectively, and $J(\theta)$ is the Jacobian defining the effect of the effect movement of the low-level state parameters on the high-level.

The actual dynamics of the high-level system are then defined by differentiating $\mathbf{y} = \mathbf{h}(\mathbf{x})$ as specified in equation (13), with $H(\mathbf{x}) = \begin{bmatrix} J(\theta) & \mathbf{0} \\ J(\theta) & J(\theta) \end{bmatrix}$. The hierarchical system used here will employ the implicit method, meaning that the high-level system state, \mathbf{y} , will be updated through online access to the transformation of the low-level state, $\mathbf{h}(\mathbf{x})$. When we equate the actual high-level system dynamics to the ideal high-level system dynamics, our equation is then

$$H(\mathbf{x})\mathbf{A}\mathbf{x} + H(\mathbf{x})\mathbf{B}\mathbf{u} = \mathbf{F}\mathbf{y} + \mathbf{G}\mathbf{v}, \quad (19)$$

$$H(\mathbf{x})\mathbf{B}\mathbf{u} = \mathbf{G}\mathbf{v}, \quad (20)$$

since $\mathbf{F}\mathbf{y} = H(\mathbf{x})\mathbf{A}\mathbf{x}$. Here, quadratic programming optimization is employed to minimize \mathbf{u} with equation (19) as a constraint. This generates a low-level control signal \mathbf{u} that best matches the high-level control signal \mathbf{v} .

The high-level control signal is designed to be a simplification of component-based optimal control, where four movements are predefined, specifically the situations where the target is in one of the four corners. The component signals are then scaled based on the distance between the hand and the target, a process called dynamic scaling (see [20] for details). This process allows a simplification of the weighting of components in 2D end-effector independent space, such that one is weighted at full value, and the others suppressed entirely. The cost function used in this implementation is a summation of quadratic functions of the distance from the target at the next time step and magnitude of current velocity.

4.2. Spiking neuron model of M1

To construct a simulation of M1 neurons, we employed the neural engineering framework (NEF) [23]. In the NEF, a group of spiking neurons represents a vector (e.g. a control signal) in a distributed manner. This is achieved by each neuron having a randomly chosen ‘preferred direction vector’ \mathbf{e} , the vector for which this neuron will fire most strongly (and a common feature of M1 neurons [29]). If we want a group of neurons to represent a particular vector \mathbf{x} , we can determine the input current J to each of these neurons by

$$J_i(\mathbf{x}) = \alpha_i \langle \mathbf{x} \cdot \mathbf{e}_i \rangle_m + J_i^{\text{bias}}, \quad (21)$$

where α is the neuron gain, J^{bias} is the background input current and \mathbf{e} is the neuron’s preferred direction vector. We choose J^{bias} and α from distributions to match those found in motor neurons [29].

By taking this input current and using any standard neuron model (such as the leaky-integrate-and-fire model used here), we can convert a vector into a pattern of spikes that represent

that vector. To determine how well that vector is represented, we also need to be able to convert a pattern of spikes back into a vector. This is done by calculating a decoding vector \mathbf{d}_i for each neuron using

$$\mathbf{d} = \Gamma^{-1}\Upsilon, \quad (22)$$

where

$$\Gamma_{ij} = \int a_i(\mathbf{x})a_j(\mathbf{x}) \, d\mathbf{x} \quad (23)$$

and

$$\Upsilon_j = \int \mathbf{x}a_j(\mathbf{x}) \, d\mathbf{x}, \quad (24)$$

where $a_i(\mathbf{x})$ is the average firing rate of the neuron i for a given value of \mathbf{x} , and the integration is over all values of \mathbf{x} over the relevant range (e.g. movement space).

The resulting \mathbf{d}_i values are the least-squares linearly optimal decoding vectors for recovering the original value of \mathbf{x} given only outputs from each of these neurons. That is, if we take the post-synaptic currents that would be generated in a receiving cell by each neuron (i.e. the ionic current caused by the neural spiking) and add them together weighting each one by \mathbf{d}_i , then the result is a time-varying estimate of \mathbf{x} . The amount of error for this decoding decreases linearly with the number of neurons used, so any degree of accuracy can be achieved by increasing the number of neurons [23].

One advantage of calculating \mathbf{d} is that we can use it to determine synaptic connection weights that will compute transformations of the vectors being represented as well. If one group of neurons represents \mathbf{x} and we have another group of neurons that we want to represent $\mathbf{M}\mathbf{x}$ (any linear transformation of \mathbf{x}), then this can be achieved by setting their synaptic connection weights ω_{ij} using

$$\omega_{ij} = \alpha_j \mathbf{e}_j \mathbf{M} \mathbf{d}_i. \quad (25)$$

Furthermore, we can calculate connections that compute any nonlinear function $f(\mathbf{x})$ by finding a different set of decoding weights. We proceed as before, but use

$$\Upsilon_j = \int f(\mathbf{x})a_j(\mathbf{x}) \, d\mathbf{x} \quad (26)$$

to find $\mathbf{d}^{f(\mathbf{x})}$, which will provide estimates of the function of interest.

These methods have been used in a wide variety of biologically realistic neural simulations [23, 78]. They are incorporated into the simulation software, Nengo (<http://www.nengo.ca>), which we used to construct the simulations reported here. The arm model employed here is the same one used in the other simulations, described above. Specifically, using Nengo we generate populations of 400 spiking neurons that represent the two-dimensional control space represented in M1, generated to drive the arm model described above.

A single simulation experiment consists of generating the control network and reaching to seven targets placed in a circle around the initial starting point of the hand. The neural spiking activity and arm trajectories were recorded, and the data were then exported to Matlab. The neural spikes were filtered by an 80 ms Gaussian filter to match the filtering applied to the spike

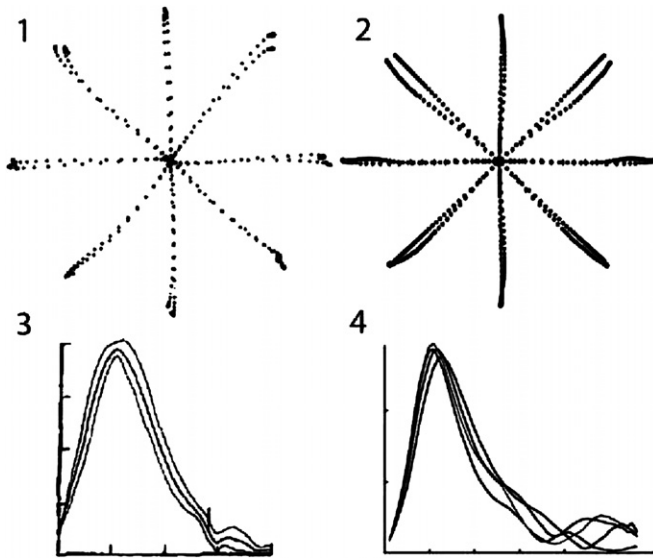


Figure 3. Comparison of human and model reaching trajectories and velocity profiles. The human data are in the left column (1 and 3), and the model data in the right (2 and 4). A comparison of figures 1 and 2 demonstrates that the model reproduces the typical, smooth reaching movements of normals. The distance between points indicates the reaching velocity. A comparison of figures 3 and 4 demonstrates that the model also reproduces the velocity profiles observed in the human data. Both axes, time and velocity, respectively, have been normalized. Data from [76], with permission.

trains gathered from monkeys performing the same task [15]. The activity of single neurons for the seven different reaches was then amalgamated, and preferred reaching movements identified by selecting the four trials with the highest overall activity. Additionally, a Gaussian filter of 70 ms was applied to the arm trajectories to account for the unmodeled movement smoothing implicit in muscle-based movements.

5. Results

5.1. Normal human arm trajectories

The resulting trajectories and velocity profiles from the simulation of normal human reaching are shown in figure 3. As can be seen, the resulting movements created by the model are highly similar to those produced by human subjects. Specifically, the total squared jerk (TSJ) of the model falls well within the 95% confidence interval of normal movements reported in [76]. For the two movements shown in figure 3, the TSJ is 0.91 and 1.08, while the confidence interval of the human data spans [0.7–1.1].

While more qualitative, it is also important that the velocity profile of the model is not a Gaussian, but is rather skewed in a manner similar to the human profile. Past models have often matched a Gaussian velocity profile explicitly, as this is often reported to be the shape of the human reaching velocity profile [62]. However, a Gaussian profile only captures the typical curve fit to the true reaching profile, which typically has a slight leftward skew, as shown in figure 3.

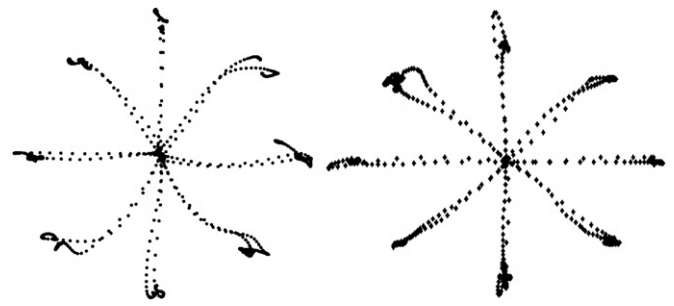


Figure 4. Huntington's patient and model trajectories. Patient trajectories on the left, and model trajectories on the right. Difficulties with ending point accuracy and additional unwanted movements at low velocities are shared between the patients and the model. The model effectively reproduces the movement termination error observed in Huntington's. Two trials are shown in each graph. Data from [76], with permission.

Given these good matches to human experimental data, this model of normal reaching provides a good basis for examining the effects of various lesions on the model's performance.

5.2. Arm trajectories in Huntington's disease

Huntington's is a neural disease present in roughly 5–10/100 000 people [36]. It is characterized by a heavy degeneration of the basal ganglia's neural structure, in particular the striatum. The symptoms usually manifest themselves during the fourth or fifth decade of a patient's lifetime, followed by steady degradation of motor function over the next 10–20 years [76]. The traditional identifying motor symptom is chorea, which is the presence of rapid, involuntary, irregular and arrhythmic complex motor actions. However, recent work has shown a stronger correlation of the progression of the disease with movement termination error in volitional movements requiring attention [52].

Here, Huntington's model was developed by impairing the performance of the basal ganglia component selection process of the working arm model described in section 4.1. Specifically, the threshold for component selection was reduced, resulting in the basal ganglia choosing additional relevant movement components. These competing components were weighted by their similarity to the ideal chosen component and summed to generate the final control signal. Similarity was computed using the dot product of the corresponding desirability maps after the application of dynamic scaling. As a result, the direction of the target, but not the distance, affects the level of 'noise' introduced by a particular component. The incorporation of additional components as noise into the chosen movement's control signal is consistent with the observed reduction of basal ganglia inhibition seen in Huntington's patients [76].

Figure 4 displays a comparison of Huntington's patient and the model's reaching trajectories. As can be seen, the model effectively reproduces the kind of movement termination errors observed in the human behavioral data. Specifically, both the model and patient exhibit unwanted and extraneous movements, especially near the target of the

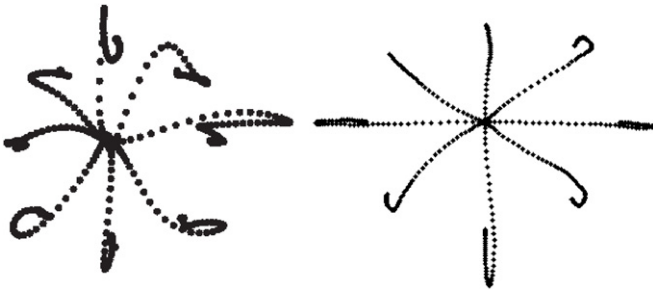


Figure 5. Cerebellar damaged patient and model trajectories. Patient trajectories on the left, and model trajectories on the right. Patients with cerebellar lesions tend to overshoot target locations, resulting in significant amounts of backtracking in the movement trajectory. The model displays a similar pattern of reaching error with perturbed cerebellar function. Data from [76], with permission.

movements. This results in a noisy, dense set of points near the targets of each reach. Using the same methods of quantifying movement described in [76], we measured the TSJ of the damaged model reaches. The HD damaged reaches display a higher than normal TSJ of 3.13 and 3.58 for the two reaches shown in figure 4. These TSJ values fall in the middle of the HD human data range [1.5–4.6] reported in [76]. As reported in the previous section, the range of TSJ for normal movements is [0.7–1.1].

5.3. Arm trajectories with cerebellar damage

In [76], the authors identify patients as having had lesions localized to the cerebellum using MRI. The cause of the lesions was not specified, though it was narrowed down to generalized cerebellar atrophy or strokes of the right posterior inferior cerebellar artery. To capture the effects of damage to the cerebellum, we add noise to the corrective signal generation process from a product of Gaussian white noise and the magnitude of difference between the two previous and new high-level control signals. The noise is taken to account for damage to synergy adaptation and error correction function attributed to the cerebellum in the NOCH. The results of the patient and model reach trajectories are displayed in figure 5.

As shown in the figure, the overshoot typical of cerebellar patients is reproduced by the model. This is a consequence of the online, fine-tuning provided by the cerebellum being largely disrupted by large random perturbations. Population level statistics were not available for a quantitative comparison of the model movements to cerebellar patients. Nevertheless, the qualitative match of the model's overshoots and slow corrections is strikingly similar to those displayed by cerebellar patients performing the same movements.

5.4. Single cell tuning in motor cortex during reach

The results from the spiking cell model of motor cortex are displayed in figure 6. Notably, the salient features of the data we discuss are identified in the original data paper [15]. As a result, while the classification of cells is qualitative, it is not model driven, but rather independently determined by

the original data analysis. Consequently, to the extent the model is able to generate cells that have the features noted by this analysis (e.g. being bi-phasic, or switching preferred directions), it can be considered to be a potential explanation of those features.

Overall, the results of this model show that there is a pre-movement convergence to a reasonably constant firing rate, followed by highly nonlinear neural activity during the execution of the movement, as is seen in the recorded neural activity. There are clearly some features of the pre-movement data not captured by the model (e.g. the pre-delay burst in A).

Nevertheless, the major movement-related features of the data neurons are found in the model neurons. For example, in A/a pre-movement relations and magnitude are preserved during the movement (i.e. black is higher than grey before and during movement), while in B/b and C/c there is a reversal of pre-movement relations during the movement (i.e. black starts higher than grey, but grey becomes higher during movement). In addition, the bi-phasic relationships of the cells in B/b and C/c are clear in both the data and the model. Finally, in D/d the neurons with different preferred directions are clearly phase-shifted with respect to one another. These properties are those identified by the experimentalists in their examination of the complete data set, so it is important that they are captured by the model.

Overall, perhaps the most important feature of the neural responses that is captured by the model is that single-cell responses are highly nonlinear during the movement, despite linear arm movements toward the target. It has been suggested that these kinds of nonlinear responses can only be captured in terms of non-plant-related parameters [15]. However, this model maps the control signals generated in the plant parameter space directly to neural responses, and finds many of the major nonlinear responses observed in the experimental setting.

6. Discussion

The simulations of a simple instantiation of the NOCH framework provided above suggest that the integration of control theoretic methods with specific anatomical mappings and possible motor functions can help provide unified accounts of a variety of motor system data.

For instance, movement termination error is successfully captured by Huntington's model. Indeed, more severe damage of the same kind described in this model also generates chorea (results not shown), as the reduced thresholding of the basal ganglia circuit can cause the involuntary execution of movements. This is caused by the unintentional selection of a movement synergy and its resulting disinhibition in the thalamo-cortical circuit that the basal ganglia projects to, causing a false 'go' signal to be given to the motor cortices, and the execution of a random movement. As a result, the model suggests a reasonably direct relationship between the amount of striatal damage and the extent of motor error. This prediction could be tested by a characterization of the complexity of involuntary movements carried out by

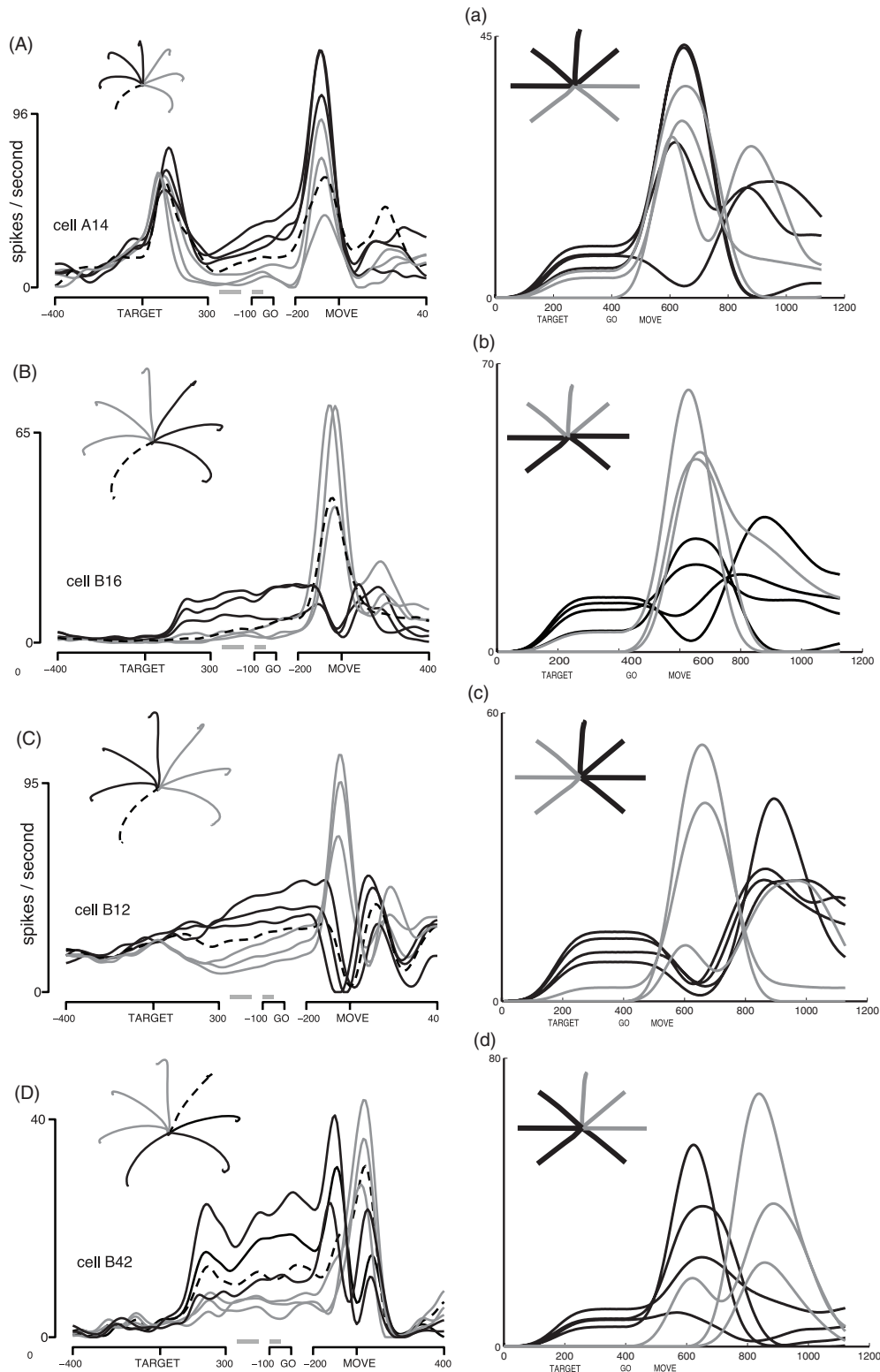


Figure 6. A comparison of spiking neuron activity from single-cell recordings in monkeys (uppercase letters) and the model (lowercase letters). The activity in monkeys was recorded from dorsal pre-motor and primary motor cortex. It includes both pre-movement activity and activity during the execution of an arm reach. Both sets of data stabilize during the pre-movement period, and then display highly nonlinear activity during movement. The model contained cells that displayed many of the salient properties of recorded cells. Black movements are in the preferred direction of the cell and grey are in the non-preferred direction. Each pair shows the model capturing a salient feature of classes of cells identified by the experimentalists. For instance, A/a show cells that generally respond well for movement initiation regardless of direction. B/b show cells whose preferred direction reverses during movement (i.e. black and grey lines cross). C/c show cells with a reversal and a biphasic response (i.e. black cells are ‘on’, ‘off’, ‘on’). D/d show cells with a preservation of preferred direction but a strong phase-shifted response related to the preferred direction (i.e. black cells fire first and grey cells second). Data from [15], with permission. (This figure is in colour only in the electronic version)

Huntington's patients at different stages of disease on a similar reaching task.

Also at the behavioral level, the utility of having the NOCH framework is evident in our modeling of cerebellar damage. Although the characterization of the patient damage is quite general, we were able to introduce perturbations to the model system that suggest specific functional consequences of cerebellar damage in the motor system. Notably, it is the same model that accounts for Huntington's, normal movement and the neural activity in primary motor cortex. Both this kind of unification and the ability to test reasonably coarse characterizations of cerebellar damage, for instance, are a result of the broad perspective provided by the NOCH.

This same perspective allows us to account for the highly nonlinear responses of cells in motor cortex during simple reaching movements. The traditional characterization of motor cortex cells assumes that they have reasonably linear relationships between movement and neural activity [29]. However, as emphasized in recent work, the activity of such cells is better characterized as the movement into a preparatory state space, and then a highly nonlinear response during reach [15]. Given these highly nonlinear responses, it has been suggested that neurons do not have a clear relationship to movement parameters, such as joint angles, torques or muscle force. Instead, it has been proposed that neurons encode an abstract 'movement-activity space' [15]. Here, however, we have shown that model neurons providing a largely linear map between the space of control signals, which control movement parameters, and neural activity are able to account for the general kinds of neural responses observed in monkey cortex (see section 4.2). The complexity of the neural response is not a function of their having a complex mapping to control space, but rather due to the complexity of the control signal itself. Understanding how this control signal is generated for movement is straightforward in the context of the broad system perspective provided by the NOCH.

A better match between the model and data could likely be achieved in a number of ways which we have not explored. For instance, the single-cell model could include additional dynamics of cortical cells, such as adaptation. In addition, the arm model could be more sophisticated, resulting in a higher dimensional, and hence more realistic control signal. This would likely help capture additional subtleties in the data, such as the kinds of ramping responses seen during the delay period. Nevertheless, even without these potential improvements the spiking model demonstrates that a number of surprising features of the neural response—such as the degree of nonlinearity during linear movement, the kinds of reversals in sensitivity observed and the kinds of phasic responses seen—are effectively captured by this model.

Taken together, these results suggest that the NOCH can be used to help generate models that can account for both high-level behavioral data and low-level neural data, in a framework based on optimal control theory. Such a unified account should prove helpful in improving our understanding of animal motor behavior from both a computational and mechanistic perspective. There is, of course, much additional work that needs to be done. For instance, the complexity of

the plant model, the depth of the hierarchy and the integration of sensory information can all be improved. In addition, the implementation of such aspects in a biologically realistic spiking simulation is crucial for detailed neurophysiological comparisons. As described above, methods for accomplishing this are available, but we have used them only to a small degree in this work. More generally, adapting currently available biologically detailed models of basal ganglia [78], cerebellum [48], and so on to the NOCH will be important for improving the biological plausibility of NOCH-based models, and for testing past models that have been proposed in isolation from the rest of the motor system. Despite these many possible improvements, we believe that even at this early stage, the NOCH has demonstrated its usefulness for integrating behavioral and neural data, and for selecting appropriate control theoretical structures to map to biological control.

It is also interesting to consider the possible ramifications of the NOCH for more applied problems in neural engineering, such as brain-computer interfaces (BCIs). Neural BCIs record the activity of a group of neurons in the cortex, and use these signals to control devices ranging from computer cursors to robotic limbs. The goal of BCIs is to allow subjects with neural implants to fluidly interact with such artificial effectors by harnessing the motor control system. The NOCH framework may prove useful in translating such signals to intended movements in two respects.

First, in the NOCH, motor cortex is proposed to be a hierarchical control structure where control can be effected on any level of the hierarchy, representing different abstractions of the end-effector space. The movement synergies that are available at these levels are used to generate the descending control signal through composition of movements, which results in descending control signals in terms of muscle activation. The interactions with artificial effectors, however, typically deal only in two- or three-dimensional space, and the recorded neural signals are typically mapped directly to target position or velocity commands [75, 73, 91]. To identify a decoder more analogous to the motor control system, the NOCH suggests that such decoding may more naturally be made into an intermediate space, rather than directly to the space used to control the effector. For example, in systems with robotic or simulated limbs, in addition to allowing control in 3D space, the system could be set up to interpret commands in joint space first or simultaneously. Instead of allowing control over position or velocity, it may be more effective to decode the recorded cortical signals in terms of force or torque applied to the end-effector or joints.

Second, in the NOCH, synergies are critical in the efficient functioning of the motor system, from cortex to spinal circuits. In fact, the spaces into which it is natural to decode neural signals are presumed to be defined over sets of synergies. Consequently, identifying a set of synergies that recorded cortical signals are likely to modulate should allow more natural motor command decoding, by taking advantage of the ability of the motor system to exploit synergies to efficiently generate movements. To identify appropriate synergies, it is imperative to characterize them in a natural biological system

before attempting to perform decoding. This would allow the identification of motor synergies for constructing BCIs that are similar to the synergies used for controlling regular limbs. Ideally, this would allow for some transference of learned synergies between the regular and artificial effectors, aiding in their effective control by drawing upon the already established set of efficient control signals.

Although many challenges lay ahead in the construction of BCIs, attempting to match the properties of the decoder well to the kinds of representations, dynamics and computational structure we expect in the biological system should help improve current performance. In short, we suspect that the NOCH can be used to guide the implementation of such suggestions to aid in the creation of efficient, robust BCI systems. Thus, even though our focus has been on relating optimal control models to behavioral and single-cell data, successfully doing so has practical implications for addressing BCI problems.

Appendix. Simplifying Bellman's equation

The Bellman equation expresses the optimal cost-to-go function, $v(\mathbf{x})$, as the minimum over actions \mathbf{u} , of the immediate cost, $\ell(\mathbf{x}, \mathbf{u})$ plus the expected cost-to-go $E[v(\mathbf{x}')]$ at the next state \mathbf{x}' . After [85], this can be written as

$$v(\mathbf{x}) = \min_{\mathbf{u}} \{ \ell(\mathbf{x}, \mathbf{u}) + E_{\mathbf{x}' \sim p(\cdot|\mathbf{x}, \mathbf{u})} [v(\mathbf{x}')] \}. \quad (\text{A.1})$$

In this notation, $\mathbf{x}' \sim p(\cdot|\mathbf{x}, \mathbf{u})$ indicates that the random variable \mathbf{x}' is distributed as the probability distribution defined by p , and the \cdot indicates that the random variable takes on all its possible values.

Often the set of actions considered in optimal control problems are taken to be from a discrete state space (e.g. 'move forward' or 'move backward'). Although such seemingly symbolic actions are typically thought of as discrete, in the LBC formulation, the action being selected is not a specific next state, but rather the transition probability to the next state. As a result, what is being selected (i.e. probabilities) are continuous. In short, what the controller is doing in choosing an action is adjusting the transition probabilities of the underlying system dynamics. This perspective on optimal control has several important consequences.

Instead of having the controller specify an action, which is then translated into an effect on the system's transition probabilities, the controller is allowed to directly specify the system's transition probabilities. The probability of the next state \mathbf{x}' , given the current state \mathbf{x} under passive dynamics, is denoted $p(\mathbf{x}'|\mathbf{x})$. This distribution determines how the system would move through the state space (possibly stochastically) without any input from the controller. The probability of the next state, given the state and the control signal \mathbf{u} , is denoted $p(\mathbf{x}'|\mathbf{x}, \mathbf{u}) = u(\mathbf{x}'|\mathbf{x})$. Notably, the probability distribution $u(\mathbf{x}'|\mathbf{x})$ is different than the chosen action \mathbf{u} . The former specifies a transition probability distribution and the latter is an action that was selected based on that distribution. The LBC controls only the former explicitly. However, the controller can only move the system to points reachable under passive dynamics.

So defined, the controller is very powerful, in that it can arbitrarily specify transition probabilities. However, different distributions come with different costs. Specifically, the control cost of an action is determined by the KL divergence between the controlled distribution and the passive dynamics distribution. KL divergence is a standard measurement of difference between two probability distributions. In this case, it is a measure of the difference between the state distribution under passive dynamics $p(\cdot|\mathbf{x})$ and under control $u(\cdot|\mathbf{x})$. Specifically, the KL divergence measures the expected logarithmic ratio between two distributions, and is denoted $KL(a(\cdot|\mathbf{x})||b(\cdot|\mathbf{x})) = E_{\mathbf{x}' \sim a(\cdot|\mathbf{x})} [\log(\frac{a(\mathbf{x}'|\mathbf{x})}{b(\mathbf{x}'|\mathbf{x})})]$.

To keep the KL divergence well defined, $u(\mathbf{x}'|\mathbf{x})$ will be set to zero whenever $p(\mathbf{x}'|\mathbf{x}) = 0$, which enforces that the controller can only move to points reachable under passive dynamics. Thus, the immediate cost of the Bellman equation, $\ell(\mathbf{x}, \mathbf{u})$, can be specified as

$$\begin{aligned} \ell(\mathbf{x}, \mathbf{u}) &= q(\mathbf{x}) + KL(u(\cdot|\mathbf{x})||p(\cdot|\mathbf{x})) \\ &= E_{\mathbf{x}' \sim u(\cdot|\mathbf{x})} \left[\log \left(\frac{u(\mathbf{x}'|\mathbf{x})}{p(\mathbf{x}'|\mathbf{x})} \right) \right], \end{aligned} \quad (\text{A.2})$$

where $q(\mathbf{x})$ is a state cost function specifying the penalty for being in a state. For example, the state cost might increase with increasing Euclidean distance from the target.

To simplify the notation, let $z(\mathbf{x}) = \exp(-v(\mathbf{x}))$ be known as the desirability function. It is equal to the exponentiation of the negative cost-to-go of the state \mathbf{x} . It is referred to as the 'desirability' function to reflect the fact that the function is large wherever the cost-to-go is small, and small wherever the cost-to-go is large.

Rewriting the Bellman equation in terms of z and substituting in equation (A.2) gives

$$-\log(z(\mathbf{x})) = q(\mathbf{x}) + \min_{\mathbf{u}} \left\{ E_{\mathbf{x}' \sim u(\cdot|\mathbf{x})} \left[\log \frac{u(\mathbf{x}'|\mathbf{x})}{p(\mathbf{x}'|\mathbf{x})z(\mathbf{x}')} \right] \right\}. \quad (\text{A.3})$$

The term being minimized is almost a proper KL divergence, except that the distribution $p(\cdot|\mathbf{x})z(\cdot)$ does not sum to 1. To correct this, a normalization term $\mathcal{G}[z](\mathbf{x})$ is introduced. Let

$$\mathcal{G}[z](\mathbf{x}) = \sum_{\mathbf{x}'} p(\mathbf{x}'|\mathbf{x})z(\mathbf{x}') = E_{\mathbf{x}' \sim p(\cdot|\mathbf{x})} [z(\mathbf{x}')], \quad (\text{A.4})$$

and multiply and divide the denominator of equation (A.4) to give

$$\begin{aligned} -\log(z(\mathbf{x})) &= q(\mathbf{x}) \\ &+ \min_{\mathbf{u}} \left\{ E_{\mathbf{x}' \sim u(\cdot|\mathbf{x})} \left[\log \left(\frac{u(\mathbf{x}'|\mathbf{x})}{p(\mathbf{x}'|\mathbf{x})z(\mathbf{x}')\mathcal{G}[z](\mathbf{x})/\mathcal{G}[z](\mathbf{x})} \right) \right] \right\} \end{aligned} \quad (\text{A.5})$$

$$\begin{aligned} &= q(\mathbf{x}) + \min_{\mathbf{u}} \left\{ E_{\mathbf{x}' \sim u(\cdot|\mathbf{x})} \left[-\log(\mathcal{G}[z](\mathbf{x})) \right. \right. \\ &\quad \left. \left. + \log \left(\frac{u(\mathbf{x}'|\mathbf{x})}{p(\mathbf{x}'|\mathbf{x})z(\mathbf{x}')\mathcal{G}[z](\mathbf{x})} \right) \right] \right\} \end{aligned} \quad (\text{A.6})$$

$$\begin{aligned} &= q(\mathbf{x}) + \min_{\mathbf{u}} \left\{ E_{\mathbf{x}' \sim u(\cdot|\mathbf{x})} \left[-\log(\mathcal{G}[z](\mathbf{x})) \right. \right. \\ &\quad \left. \left. + KL \left(u(\mathbf{x}'|\mathbf{x}) \parallel \frac{p(\mathbf{x}'|\mathbf{x})z(\mathbf{x}')}{\mathcal{G}[z](\mathbf{x})} \right) \right] \right\}. \end{aligned} \quad (\text{A.7})$$

The global minimum of a KL divergence occurs when the two distributions being compared are equal. This means that the optimal control signal, u^* , will be reached when

$$u^*(\mathbf{x}'|\mathbf{x}) = \frac{p(\mathbf{x}'|\mathbf{x})z(\mathbf{x}')}{\mathcal{G}[z](\mathbf{x})}. \quad (\text{A.8})$$

We can further simplify the Bellman equation. First we must exponentiate and incorporate the normalization term, giving

$$z(\mathbf{x}) = \exp \left(-q(\mathbf{x}) - \min_u \left\{ E_{\mathbf{x}' \sim u(\cdot|\mathbf{x})} \left[-\log(\mathcal{G}[z](\mathbf{x})) + KL \left(u(\mathbf{x}'|\mathbf{x}) \left\| \frac{p(\mathbf{x}'|\mathbf{x})z(\mathbf{x}')}{\mathcal{G}[z](\mathbf{x})} \right\| \right) \right] \right\} \right). \quad (\text{A.9})$$

Because $-\log(\mathcal{G}[z](\mathbf{x}))$ is not dependent on \mathbf{u} it can be moved outside the minimization, and assuming optimal control u^* , we can set the KL divergence to zero giving

$$z(\mathbf{x}) = \exp(-q(\mathbf{x}) + \log(\mathcal{G}[z](\mathbf{x})) + 0) \quad (\text{A.10})$$

$$z(\mathbf{x}) = \exp(-q(\mathbf{x}))\mathcal{G}[z](\mathbf{x}). \quad (\text{A.11})$$

Since the expectation $\mathcal{G}[z](\mathbf{x})$ is a linear operator, equation (A.11) is linear in terms of z . We can rewrite this in vector notation by enumerating the set of possible states 1 to n and letting $z(\mathbf{x})$ and $q(\mathbf{x})$ be represented by the n -dimensional column vectors \mathbf{z} and \mathbf{q} , respectively, $p(\mathbf{x}'|\mathbf{x})$ by the n -by- n matrix \mathbf{P} , where the row indices correspond to \mathbf{x} and the column indices correspond to \mathbf{x}' . Using these terms, equation (A.11) can be rewritten as

$$\mathbf{z} = \mathbf{Q}\mathbf{P}\mathbf{z}, \quad (\text{A.12})$$

where $\mathbf{Q} = \exp(-q(\mathbf{x}))$. In this form we see that \mathbf{z} is an eigenvector of the matrix $\mathbf{Q}\mathbf{P}$, and \mathbf{z} can be solved for once the state cost matrix \mathbf{Q} and passive dynamics matrix \mathbf{P} have been defined. Using \mathbf{z} the controller can recover the optimal action at each defined state in the environment.

Notably, as has been emphasized elsewhere [85], this formulation effectively captures the stochastic nature of many optimal control strategies. Optimal strategies are clearly not always deterministic, as in the case of various escape behavior. In general, it is important that 'optimal' does not mean deterministic in the LBC framework. As a result, the cost function can be altered to capture various constraints such as time to target, reducing the accuracy costs, or removing constraints on system parameters not relevant for the particular problem. The LBC approach attempts to find a stochastic policy that best satisfies the imposed constraints, and does so with a relatively simple operation. However, the cost of having this operation be simple is that the complexity of the passive dynamics matrix itself may be quite high. Nevertheless, this may be consistent with the highly parallel computations most naturally performed by the brain.

References

[1] Lorinez A 1997 Common control principles of basal ganglia—thalamocortical loops and the hippocampus *Neural Netw. World* 1–30
 [2] Abdelghani M N 2011 The role of sensitivity derivatives in sensorimotor learning *PhD Thesis* University of Toronto

[3] Andani M E, Bahrami F, Jabejdar Maralani P and Ijspeert A J 2009 MODEM: a multi-agent hierarchical structure to model the human motor control system *Biol. Cybern.* **101** 361–77
 [4] Athans M 1971 The role and use of the stochastic linear-quadratic-Gaussian problem in control system design *Trans. Autom. Control* **16** 529–52
 [5] Bar-Gad I, Morris G and Bergman H 2003 Information processing, dimensionality reduction and reinforcement learning in the basal ganglia *Prog. Neurobiol.* **71** 439–73
 [6] Bastian A 2006 Learning to predict the future: the cerebellum adapts feedforward movement control *Curr. Opin. Neurobiol.* **16** 645–9
 [7] Berniker M, Jarc A, Bizzi E and Tresch M C 2009 Simplified and effective motor control based on muscle synergies to exploit musculoskeletal dynamics *Proc. Natl Acad. Sci. USA* **106** 7601–6
 [8] Billard A, Epars Y, Calinon S, Schaal S and Cheng G 2004 Discovering optimal imitation strategies *Robot. Auton. Syst.* **47** 69–77
 [9] Bloxham C A, Mindel T A and Frith C D 1984 Initiation and execution of predictable and unpredictable movements in Parkinson's disease *Brain* **107** 371–84
 [10] Blumenfeld H 2002 *Neuroanatomy Through Clinical Cases* (Sunderland, MA: Sinauer Associates)
 [11] Bortoff G and Strick P L 1993 Corticospinal terminations in two new-world primates: further evidence that corticomotoneuronal connections provide part of the neural substrate for manual dexterity *J. Neurosci.* **13** 5105–18
 [12] Bostan A C, Dum R P and Strick P L 2010 The basal ganglia communicate with the cerebellum *Proc. Natl Acad. Sci. USA* **107** 8452–6
 [13] Caligiore D, Ferrauto T, Parisi D, Accornero N, Capozza M and Baldassarre G 2008 Using motor babbling and Hebb rules for modeling the development of reaching with obstacles and grasping *Int. Conf. on Cognitive Systems* pp E1–8
 [14] Chan S S and Moran D W 2006 Computational model of a primate arm: from hand position to joint angles, joint torques and muscle forces *J. Neural Eng.* **3** 327–37
 [15] Churchland M M, Cunningham J P, Kaufman M T, Ryu S I and Shenoy K V 2010 Cortical preparatory activity: representation of movement or first COG in a dynamical machine? *Neuron* **68** 387–400
 [16] Crespi A and Ijspeert A J 2009 Salamandra robotica: a biologically inspired amphibious robot that swims and walks *Robotics* pp 3–36
 [17] da Silva M, Durand F and Popovic J 2009 Linear bellman combination for control of character animation *Proc. SIGGRAPH ACM Trans. Graph.* **28** 82:1–82:10
 [18] Demiris Y and Meltzoff A 2008 The robot in the crib: a developmental analysis of imitation skills in infants and robots *Infant Child Dev.* **53** 43–53
 [19] Deneve S and Pouget A 2003 Basis functions for object-centered representations *Neuron* **37** 347–59
 [20] DeWolf T 2010 NOCH: a framework for biologically plausible models of neural motor control *PhD Thesis* University of Waterloo
 [21] Doya K 1999 What are the computations of the cerebellum, the basal ganglia and the cerebral cortex *Neural Netw.* **12** 961–74
 [22] Doya K 2009 How can we learn efficiently to act optimally and flexibly? *Proc. Natl Acad. Sci. USA* **106** 11429–30
 [23] Eliasmith C and Anderson C H 2003 *Neural Engineering: Computation, Representation and Dynamics in Neurobiological Systems* (Cambridge, MA: MIT Press)
 [24] Felleman D J and Van Essen D C 1991 Distributed hierarchical processing in the primate cerebral cortex *Cereb. Cortex* **1** 1–47

- [25] Fishbach A, Roy S A, Bastianen C, Miller L E and Houk J C 2007 Deciding when and how to correct a movement: discrete submovements as a decision making process *Exp. Brain Res.* **177** 45–63
- [26] Fix J 2007 *Atlas of the Human Brain and Spinal Cord* (Jones & Bartlett Learning)
- [27] Fried I, Katz A, McCarthy G, Sass K J, Williamson P, Spencer S S and Spencer D D 1991 Functional organization of human supplementary motor cortex studied by electrical stimulation *J. Neurosci.* **11** 3656–66
- [28] Gail A and Andersen R A 2006 Neural dynamics in monkey parietal reach region reflect context-specific sensorimotor transformations *J. Neurosci.* **26** 9376–84
- [29] Gentilucci M and Negrotti A 1999 The control of an action in Parkinson's disease *Exp. Brain Res.* **269**–77
- [30] Georgopoulos A P, Schwartz A B and Kettner R E 1986 Neuronal population coding of movement direction *Science* **233** 1416–9
- [31] Gershwin S B 1987 A hierarchical framework for discrete event scheduling in manufacturing systems *Discrete Event Systems: Models and Application (Lecture Notes in Computer Science vol 103)* ed P Varaiya and H Kurzhanski (Berlin: Springer) pp 197–216
- [32] Ghez C and Thach T 2000 *The Cerebellum* 4th edn (New York: McGraw-Hill) chapter 42
- [33] Ghez C and Thach T 2000 *The Organization of Movement* 4th edn (New York: McGraw-Hill) chapter 33
- [34] Gluck M A, Allen M T, Myers C E and Thompson R F 2001 Cerebellar substrates for error correction in motor conditioning *Neurobiol. Learn. Mem.* **76** 314–41
- [35] Gopnik A, Meltzoff A N and Kuhl P K 2000 *The Scientist in the Crib: What Early Learning Tells us About the Mind* (Perennial)
- [36] Guibas L, Sharir M and Sifrony S 1988 On the general motion planning problem with two degrees of freedom *Proc. 4th Annu. Symp. on Computational Geometry* pp 289–98
- [37] Gurney K, Prescott T J and Redgrave P 2001 A computational model of action selection in the basal ganglia: I. A new functional anatomy *Biol. Cybern.* **84** 401–10
- [38] Harper P S 1992 Review Article: the epidemiology of Huntington's disease *Hum. Genet.* **365**–76
- [39] Haruno M, Wolpert D M and Kawato M 2001 Mosaic model for sensorimotor learning and control *Neural Comput.* **13** 2201–20
- [40] Hinton G E and Salakhutdinov R R 2006 Reducing the dimensionality of data with neural networks *Science* **313** 504–7
- [41] Hoffman D S and Strick P L 1995 Effects of a primary motor cortex lesion on step-tracking movements of the wrist *J. Neurophysiol.* **73** 891–5
- [42] Hooper S L and DiCaprio R A 2004 Crustacean motor pattern generator networks *Neuro-Signals* **13** 50–69
- [43] Hoshi E, Tremblay L, Féger J, Carras P L and Strick P L 2005 The cerebellum communicates with the basal ganglia *Nat. Neurosci.* **8** 1491–3
- [44] Hyvärinen A and Hoyer P 1999 Emergence of topography and complex cell properties from natural images using extensions of ICA *Advances in Neural Information Processing Systems* (Cambridge, MA: MIT Press) pp 827–33
- [45] Ivashko D G, Prilutsky B I, Markin S N, Chapin J K and Rybak I A 2003 Modeling the spinal cord neural circuitry controlling cat hindlimb movement during locomotion *Neurocomputing* **52–4** 621–9
- [46] Ting L and Lucas Mckay J 2010 Muscle synergies produce expansive behavioral biases during postural control *Neural Control of Movement 2010* pp 14–5
- [47] Jung R, Kiemel T and Cohen A H 1996 Dynamic behavior of a neural network model of locomotor control in the lamprey *J. Neurophysiol.* **75** 1074–86
- [48] Kalaska J F, Cohen D A, Hyde M L and Prud'homme M 1989 A comparison of movement direction-related versus load direction-related activity in primate motor cortex, using a two-dimensional reaching task *J. Neurosci.* **9** 2080–102
- [49] Kawato M 1999 Internal models for motor control and trajectory planning *Curr. Opin. Neurobiol.* **9** 718–27
- [50] Kleinman D L 1970 An optimal control model of human response: part I. Theory and validation *Automatica* **3** 357–69
- [51] Kuroda S, Yamamoto K, Miyamoto H, Doya K and Kawat M 2001 Statistical characteristics of climbing fiber spikes necessary for efficient cerebellar learning *Biol. Cybern.* **84** 183–92
- [52] Lee H, Ekanadham C and Ng A 2007 Sparse deep belief net model for visual area v2 *Energy* **19** 1–8
- [53] Lemay M, Chouinard S, Richer F and Lesperance P 2008 Huntington's disease affects movement termination *Behav. Brain Res.* **187** 153–8
- [54] Li W, Todorov E and Pan X 2005 Hierarchical feedback and learning for multi-joint arm movement control *IEEE Eng. Med. Biol. Soc.* **4** 4400–3
- [55] Liu D and Todorov E 2009 Hierarchical optimal control of a 7-dof arm model *2009 IEEE Symp. on Adaptive Dynamic Programming and Reinforcement Learning* pp 50–7
- [56] Maier M A, Olivier E, Baker S N, Kirkwood P A, Morris T and Lemon R N 1997 Direct and indirect corticospinal control of arm and hand motoneurons in the squirrel monkey (*saimiri sciureus*) *J. Neurophysiol.* **78** 721–33
- [57] McKay N 1985 Minimum cost-control of robotic manipulators with geometric path constraints *PhD Thesis* University of Michigan
- [58] Meltzoff A and Keith Moore M 1997 Explaining facial imitation: a theoretical model *Early Dev. Parenting* **6** 179–92
- [59] Morari M, Garcia C E and Pretz D M 1989 Model predictive control: theory and practice—a survey *Automatica* **25** 335–48
- [60] Morton S M and Bastian A J 2004 Cerebellar control of balance and locomotion *Neuroscientist* **10** 247–59
- [61] Mulliken G H, Musallam S and Andersen R A 2008 Decoding trajectories from posterior parietal cortex ensembles *J. Neurosci.* **28** 12913–26
- [62] Nenadic Z, Anderson C H and Ghosh B 2000 Control of arm movement using population of neurons *Proc. 39th IEEE Conf. on Decision and Control* vol 35 pp 1776–81
- [63] Olshausen B A and Field D J 1996 Emergence of simple-cell receptive field properties by learning a sparse code for natural images *Nature* **381** 607–9
- [64] Olshausen B A 2002 *Sparse Codes and Spikes* (Cambridge, MA: MIT Press) pp 257–72
- [65] Penfield W and Rasmussen T 1950 *The Cerebral Cortex of Man* vol 40 (London: Macmillan)
- [66] Peters J and Schaal S 2007 Reinforcement learning by reward-weighted regression for operational space control *Proc. 24th Int. Conf. on Machine Learning* pp 745–50
- [67] Piaget J 1952 *The Origins of Intelligence in Children* (New York: International University Press)
- [68] Pouget A and Sejnowski T J 1997 Spatial transformations in the parietal cortex using basis functions *J. Cogn. Neurosci.* **9** 222–37
- [69] Riehle A and Vaadia E (ed) 2005 *Motor Cortex in Voluntary Movements: A Distributed System for Distributed Functions* (Boca Raton, FL: CRC Press)
- [70] Sanes J N and Donoghue J P 2000 Plasticity and primary motor cortex *Annu. Rev. Neurosci.* **23** 393–415
- [71] Schwartz A B 1994 Direct cortical representation of drawing *Science* **265** 540–2
- [72] Schwartz A B, Tracy Cui X, Weber D J and Moran D W 2006 Brain-controlled interfaces: movement restoration with neural prosthetics *Neuron* **52** 205–20

- [73] Scott S H 2004 Optimal feedback control and the neural basis of volitional motor control *Nat. Rev. Neurosci.* **5** 532–46
- [74] Shenoy K V, Meeker D and Cao S 2003 Neural prosthetic control signal from plan activity *Neuroreport* **14** 591–6
- [75] Smith M A, Brandt J and Shadmehr R 2000 Motor disorder in Huntington's disease begins as a dysfunction in error feedback control *Nature* **403** 544–9
- [76] Souza A D, Vijayakumar S and Schaal S 2001 Learning inverse kinematics *Proc. IROS* pp 298–303
- [77] Stewart T C, Choo X and Eliasmith C 2010 Dynamic behaviour of a spiking model of action selection in the basal ganglia *Int. Conf. on Cognitive Modeling* pp 235–40
- [78] Sutton R S and Barto A G 1998 *Reinforcement Learning—An Introduction* (Cambridge, MA: MIT Press)
- [79] Tahara K, Arimoto S, Sekimoto M and Luo Z-W 2009 On control of reaching movements for musculo-skeletal redundant arm model *Appl. Bionics Biomech.* **6** 11–26
- [80] Taira M, Boline J, Smyrnis N, Georgopoulos A and Ashe J 1996 On the relations between single cell activity in the motor cortex and the direction and magnitude of three-dimensional static isometric force *Exp. Brain Res.* **109** 367–76
- [81] Todorov E 2008 Parallels between sensory and motor information processing *The Cognitive Neurosciences* 4th edn (Cambridge, MA: MIT Press)
- [82] Todorov E, Li W and Pan X 2005 From task parameters to motor synergies: a hierarchical framework for approximately-optimal feedback control of redundant manipulators *J. Robot. Syst.* **22** 691–710
- [83] Todorov E 2003 On the role of primary motor cortex in arm movement control *Cogn. Sci.* **3** 125–66
- [84] Todorov E 2009 Efficient computation of optimal actions *Proc. Natl Acad. Sci. USA* **106** 11478–83
- [85] Todorov E 2006 *Optimal Control Theory* (Cambridge, MA: MIT Press) chapter 12
- [86] Trinkle J C, Abel J M and Paul R P 1988 An investigation of frictionless enveloping grasping in the plane *Int. J. Robot. Res.* **7** 33–51
- [87] Tripp B 2009 A search for principles of basal ganglia function *PhD Thesis* University of Waterloo
- [88] Turella L, Pierno A C, Tubaldi F and Castiello U 2009 Mirror neurons in humans: consisting or confounding evidence? *Brain Lang.* **108** 10–21
- [89] Turner R S, Desmurget M, Turner C and S. R 2010 Basal ganglia contributions to motor control: a vigorous tutor *Curr. Opin. Neurobiol.* **20** 704–16
- [90] Valero-Cuevas F J, Venkadesan M and Todorov E 2009 Structured variability of muscle activations supports the minimal intervention principle of motor control *J. Neurophysiol.* **102** 59–68
- [91] Velliste M, Perel S, Spalding M C, Whitford A S and Schwartz A B 2008 Cortical control of a prosthetic arm for self-feeding *Nature* **453** 1098–101
- [92] Wing A M 2000 Motor control: mechanisms of motor equivalence in handwriting *Curr. Biol.* **10** R245–8
- [93] Wolpert D M, Miall R C and Kawato M 1998 Internal models in the cerebellum *Trends Cogn. Sci.* **2** 338–47