

RESEARCH ARTICLE

Reciprocity Between the Cerebellum and the Cerebral Cortex: Nonlinear Dynamics in Microscopic Modules

JUN WANG,¹ GREGORY DAM,² SULE YILDIRIM,³ WILLIAM RAND,⁴ URI WILENSKY⁵ AND JAMES C. HOUK⁶

¹Department of Physiology, Northwestern University Medical School, Chicago, Illinois; ²Northwestern University Institute for Neuroscience, Chicago, Illinois; ³Department of Computer Science, Hedmark University College, Elverum, Norway; ⁴Northwestern Institute on Complex Systems, Northwestern University, Evanston, Illinois; ⁵Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, Illinois and ⁶Department of Physiology, Northwestern University Medical School, Chicago, Illinois

This article was submitted as an invited paper in the Understanding Complex Systems conference held at the University of Illinois-Urbana Champaign, May 2006

Received October 1, 2006; accepted April 29, 2008

The cerebellum and basal ganglia are reciprocally connected with the cerebral cortex, forming many loops that function as distributed processing modules. Here we present a detailed model of one microscopic loop between the motor cortex and the cerebellum, and we show how small arrays of these CB modules can be used to generate biologically plausible motor commands for controlling movement. A fundamental feature of CB modules is the presence of positive feedback loops between the cerebellar nucleus and the motor cortex. We use nonlinear dynamics to model one microscopic loop and to investigate its bistable properties. Simulations demonstrate an ability to program a motor command well in advance of command generation and an ability to vary command duration. However, control of command intensity is minimal, which could interfere with the control of movement velocity. To assess these hypotheses, we use a minimal nonlinear model of the neuromuscular (NM) system that translates motor commands into actual movements. Simulations of the combined CB-NM modular model indicate that movement duration is readily controlled, whereas velocity is poorly controlled. We then explore how an array of eight CB-NM modules can be used to control the direction and endpoint of a planar movement. In actuality, thousands of such microscopic loops function together as an array of adjustable pattern generators for programming and regulating the composite motor commands that control limb movements. We discuss the biological plausibility and limitations of the model. We also discuss ways in which an agent-based representation can take advantage of the modularity in order to model this complex system. © 2008 Wiley Periodicals, Inc. *Complexity* 00: 000–000, 2008

Key Words: neural network; motor cortex; cerebellum; movement; nonlinear dynamics; motor command; equilibrium point control; agent-based modeling

Correspondence to: James C. Houk, 303 E. Chicago Avenue, Chicago, IL 60611; e-mail: j-houk@northwestern.edu

1. INTRODUCTION

Anatomical and physiological studies have identified many loops between the cerebral cortex and two subcortical structures, the basal ganglia, and the cerebellum (CB) [1]. These loops, called distributed processing modules (or DPMs as defined by Houk [2]), have been regarded as the brain's substrate for the planning, initiation, and regulation of movement, as well as thought. It is estimated that in the human brain there may be on the order of a hundred DPMs forming a large-scale neural network. Because the different DPMs have strikingly similar neuronal architectures, their signal processing operations may be essentially identical. The resultant network of generic DPMs thus provides a promising framework for understanding many questions about how the mind thinks and controls action.

As a first step toward understanding the DPM framework, our task is to build a computational model for one well-known set of loops, called the M1-DPM, between the primary motor cortex (MC), the basal ganglia, and the CB. Even though the M1-DPM is the module that we know most about from anatomical and physiological perspectives, its complexity is far beyond conventional computational modeling. Therefore, instead of studying the whole M1-DPM, we will start by investigating some of the basic elements of this module. One basic element is the loop between the primary MC and the CB. It is responsible for the generation of the composite motor commands that control voluntary limb movements [3].

Experimental data underlying the planning and control of movement were reviewed and synthesized by Houk and Wise [4]. They concluded that, while the loop through the basal ganglia uses a combination of context, sensory, and internal events to select a salient action, the loop through the CB is responsible for the programming and execution of the selected action. Figure 2 in the Houk and Wise' paper summarizes the time course of the programming and execution phases of voluntary motor command generation. Motor commands can be programmed before they are executed by the signal processing operations in the loop through the CB.

The essential features of cortical-cerebellar signal processing were enunciated by Berthier et al. [5] in their adjustable pattern generator (APG) model of the CB. According to the APG model, motor programs are stored mainly in the cerebellar cortex in the weights of parallel fiber synapses onto Purkinje cells (PCs). After training of these weights is complete, programs can be recalled from these memory sites by a selection process (ascribed to inhibitory interneurons called basket cells). In analogy with psychological ideas about motor programs, many of the parameters of the recalled program are determined centrally. In line with [4], after a program has been selected, APG modules wait idle until the arrival of a trigger signal that "jump starts" program execution, analogous to the role of

a cue in a reaction-time task [6]. The trigger is envisioned as a transient sensory or central input that initiates positive feedback in the loop between MC and the cerebellar nucleus (CN). As reviewed in the next section, positive feedback seems to be the driving force for the generation of the motor command, while accuracy in direction and amplitude is regulated by inhibition from PCs. Because PCs were modeled as bistable devices, motor programs were initially executed in a feedforward manner. Berthier et al. proposed that PCs detect when the endpoint of the movement is about to be achieved, whereupon they switch to activated states that inhibit positive feedback and thus terminate motor program execution [5].

The APG array model focused on learning and operational mechanisms in the cerebellar cortex, whereas this article focuses on the operational role of the loops between the CN and MC. These loops were treated abstractly in [5]. The PC input to the loops modeled here will be a topic of a future investigation, but in this article the PC activity is set to a predetermined reasonable value.

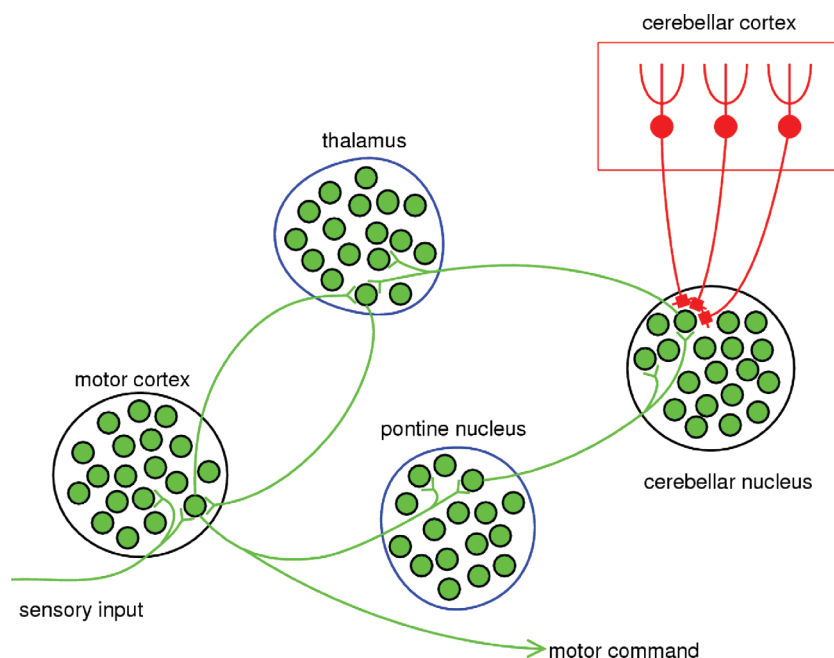
We begin by reviewing anatomical and physiological data regarding reciprocity between CB and MC, converging upon the necessity to analyze a minimal model of one microscopic loop, called a CB module. We provide an analytical analysis of the nonlinear dynamics of a single CB module and a numerical analysis of one module's capacity to generate plausible motor commands. Next we present a model of the neuromuscular (NM) modules, which are the targets of the motor commands sent from CB modules. We combine the two kinds of modular models to form CB-NM modules and use these to test the ability of CB modules to control the duration and velocity of movement. Next we test the capacity of an array of eight CB-NM modules for controlling the direction and endpoint of a planar movement. Finally we discuss the biological plausibility and limitations of the model, and we consider how modularity can be used to create an agent-based model.

2. MICROSCOPIC CEREBELLAR MODULES AND THE ANALYSIS OF THEIR NONLINEAR DYNAMICS

The anatomy and physiology of the two macroscopic loops between the CB and the MC are treated in detail by Houk and Mugnaini [3]. The loop through the CN is predominantly excitatory whereas the loop through the cerebellar cortex is strongly inhibitory. The macroscopic loop through CN is actually comprised of thousands of microscopic loops. Figure 1 illustrates how an MC neuron that is generating a motor command sends a branch to the pontine nucleus. Pontine neurons then send a branch to CN as they project their main axons into CB cortex (only the branch to CN is shown in Figure 1). CN neurons project out of the CB to the thalamus, which is reciprocally

F1

FIGURE 1



Reciprocity between the cerebellum and the motor cortex includes a loop of predominantly excitatory connections (green) and an inhibitory pathway from Purkinje cells in the cerebellar cortex (red). A complex mixture of feedback and feed-forward inputs to the cerebellar cortex, which are not shown here, is used to regulate the generation of motor commands. Sensory inputs to motor cortex can initiate the generation of a motor command.

connected to the MC neurons. Motor commands are also generated by magnocellular red nucleus neurons, which form analogous connections with the CB [7].

Figure 1 highlights just one of the thousands of microscopic loops that are formed between MC and CN in the course of development [8]. Developmental plasticity converts a nonspecific cortical-cerebellar macroscopic loop into thousands of relatively private microscopic loops. In [8], Hua and Houk review the literature and present a computational model of this developmental process.

There is extensive empirical and computational evidence that positive feedback around the reciprocal microscopic loops is the main driving force for the generation of the high firing rates characteristic of motor commands (e.g., [6,9,10]). A potent inhibitory input to the CN from PCs in the cerebellar cortex restrains the positive feedback and shapes neural activity into precise motor commands, commands that produce accurate limb movements [11]. Individual motor commands are bursts of discharge that display a large range of intensities and durations in order to control movements of different velocities and durations [12,13].

Figure 1 illustrates the convergence of three PCs onto the CN neuron that participates in the highlighted loop. Actually more than a hundred PCs contribute to the regulation of the intensity and duration of the loop activity that is responsible for one motor command [3]. Because

this article focuses on the nonlinear dynamics of reciprocal loops, our minimal model of a microscopic module, or a CB module (see Figure 2), includes only a single equivalent PC that is set to a reasonable value. Furthermore, we represent information flow around each microscopic loop with just two neurons. Arrays of these CB modules are used to control planar movements in different directions.

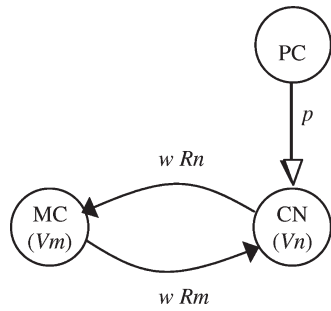
2.1. The Minimal Model

As shown in Figure 2, the model we are considering contains just two state variables: V_m and V_n , and two parameters: w and p . The variables V_m and V_n represent the membrane potentials of the MC and CN neurons, respectively. The parameter w represents the synaptic weight from MC to CN, and for simplicity, also the weight from CN to MC. The parameter p represents the inhibition from PCs; this treatment of viewing p as a parameter instead of a variable tacitly assumes that the inhibition from PC changes on a relatively slow time scale.

In addition to the two state variables and two parameters, Figure 2 also illustrates two firing rates, R_m and R_n . Among them, R_m is the output of the activation function that takes as input V_m , the membrane potential of MC. In other words, $R_m = f(V_m)$, where f denotes the activation function. It is the same for R_n . Note that it is R_m , the out-

F2

FIGURE 2



A minimal model of a CB module. The output p from the PC regulates positive feedback between a CN neuron and an MC neuron. R_n is the firing rate of the CN neuron and R_m is the firing rate of the MC neuron. Note that R_m also specifies the intensity of the motor command (see Figure 1). Both connections have a synaptic weight w .

put from the MC neuron, that determines the intensity and time course of the motor command. In this article, the activation function is implemented as the standard sigmoid function in the form of $f(x) = 1/(1 + e^{-x})$. The model of one microscopic module, therefore, is given by the following equations:

$$\tau \dot{V}_m = -V_m + wR_n - b \quad (1)$$

$$R_m = 1/(1 + e^{-V_m}) \quad (2)$$

$$\tau \dot{V}_n = -V_n + wR_m - p \quad (3)$$

$$R_n = 1/(1 + e^{-V_n}) \quad (4)$$

In Eq. (1), b is a constant that indicates some bias input to the MC neuron. Considering that our numerical analysis shows that there is no qualitative difference when b takes different values as long as it is positive, so it will be fixed to 5 hereafter. The terms $-V_m$ and $-V_n$ in Eqs. (1) and (3) represent the intrinsic membrane potential decay of the neurons MC and CN. In addition, in Eqs. (1) and (3), τ is the time constant. For convenience, in the following dynamics analysis, we suppose that the time constant $\tau = 1$; but in the numerical simulations given in Section 2.3 and the following sections, the time constant τ will be set to 0.01 to be consistent with the time scale in real human movements.

By substituting the firing rates, R_m and R_n , into Eqs. (1) and (3), we can reduce the above four equations to the following pair of differential equations:

$$\begin{cases} \dot{V}_m = -V_m + w/(1 + e^{-V_n}) - 5 \\ \dot{V}_n = -V_n + w/(1 + e^{-V_m}) - p \end{cases} \quad (5)$$

In the following subsections, we will focus on finding out under what conditions the motor command can be

initiated or terminated. The conditions will be described in terms of the two parameters of the system (synaptic weight w and PC inhibition p), and the initial values of membrane potentials of MC and CN (V_m and V_n).

2.2. Bifurcation Analysis

2.2.1. Fixed Points

We seek to understand how the fixed points and their stability change as the two parameters w and p change. First, we need to determine the fixed points of the system. A geometrical approach is used to find the fixed points. The basic idea is that the intersections between the two nullclines derived from Eq. (5) occur at the fixed points of the system [14]. For a differential equation $\dot{x} = f(x)$, its nullcline is the curve defined by function $\dot{x} = 0$. The two nullclines of the system, in terms of V_m vs. V_n , are

$$\begin{cases} V_m = w/(1 + e^{-V_n}) - 5 \\ V_m = -\log(w/(V_n + p) - 1) \end{cases}$$

As we have two parameters w and p , we will think of w as fixed, and see what will happen when p changes.¹ Figure 3 shows that when $w < w_c$ (where w_c is to be determined), the two nullclines have exactly one intersection. But when $w > w_c$, the two nullclines can have one, two, or three intersections, depending on the value of p . In this case, if $p > p_b$ or $p < p_a$ (where p_a and p_b are to be determined), there is only one intersection; if p is between p_a and p_b , there are three intersections; and if $p = p_a$ or $p = p_b$, there are two intersections.

2.2.2. Bifurcations

Next we determine the above critical values: w_c , p_a , and p_b . These critical values are bifurcation points since at those points the dynamics of the system changes qualitatively. It turns out that for our model the bifurcation points can be obtained in the following parametric form ($w(V_m), p(V_m)$), where V_m runs through all possible values.

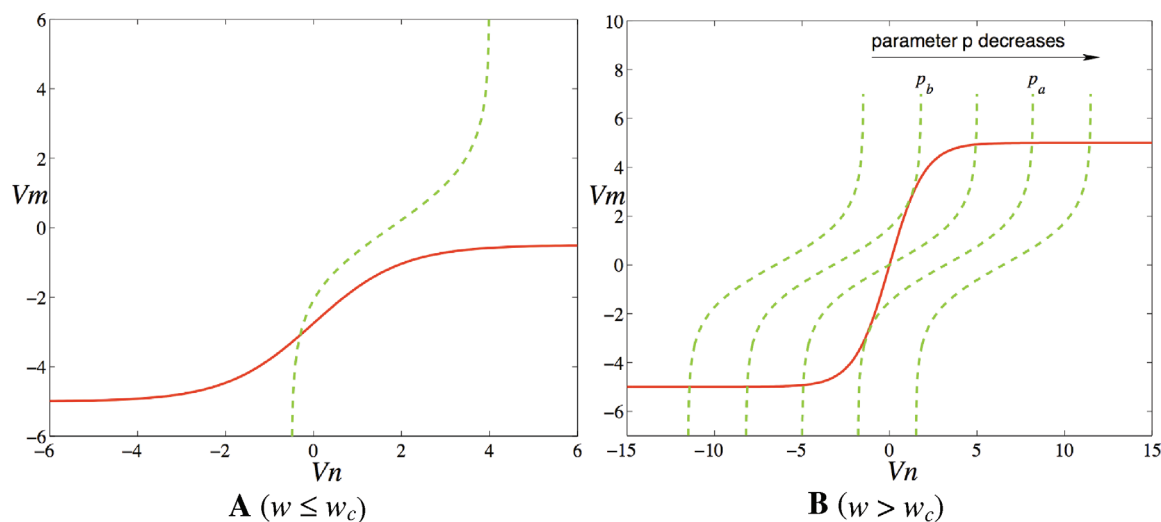
$$\begin{cases} w(V_m) = V_m + 5 + \frac{(1 + e^{-V_m})^2}{(V_m + 5)e^{-V_m}} \\ p(V_m) = V_m + \frac{V_m + 5}{1 + e^{-V_m}} + \frac{1 + e^{V_m}}{V_m + 5} - 2 \log \frac{V_m + 5}{1 + e^{-V_m}} \end{cases} \quad (6)$$

From the above parametric form, we can plot a stability diagram for the system, as shown in Figure 4. The diagram shows that there are two bifurcation curves that meet tangentially at a cusp point $(w, p) = (5.27, 0.27)$. The diagram not only shows how the number of fixed points changes

¹This treatment is better than the other way around—fixing p and see what will happen when w changes.

COLOR

FIGURE 3



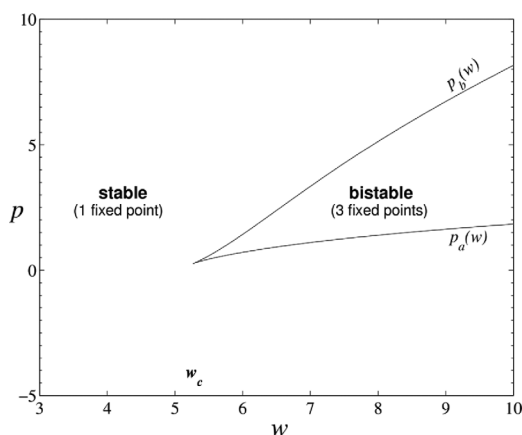
The intersections between the nullclines occur at the fixed points of the system. The red curve is the nullcline for $\dot{V}_m = 0$. The green dashed curves are the nullclines for $\dot{V}_n = 0$ for different values of p .

as parameters w and p change, but also the stability of the fixed points. When (w, p) is in the bistable region, the system has three fixed points, among which one is unstable and two are stable.

To develop a better understanding of the above stability diagram, Figure 5 illustrates the cusp catastrophe surface

of V_m^* vs. w and p , where V_m^* is the equilibrium value of V_m for a given pair (w, p) . Note that there are some folds in the surface, and the projection of these folds onto the (w, p) plane gives rise to the bifurcation curves plotted in Figure 4. The cusp catastrophe surface shows that as parameters change, the equilibrium state of the system can jump down (or up) suddenly to the lower (or upper) surface. This is important because the dynamics of the system undergoes a qualitative change at these bifurcation points. Figure 6 shows a cross-section of the cusp surface at $w = 10$, which is above the threshold $w_c = 5.27$. At $w = 10$, two thresholds for PC can be observed: the lower threshold $p_a = 1.8$ and the upper threshold $p_b = 8.2$. When $p < p_a$ or $p > p_b$, there is only one stable fixed point; when $p_a < p < p_b$, there are three fixed points, among which one is unstable (see the dashed line) and two are stable.

FIGURE 4



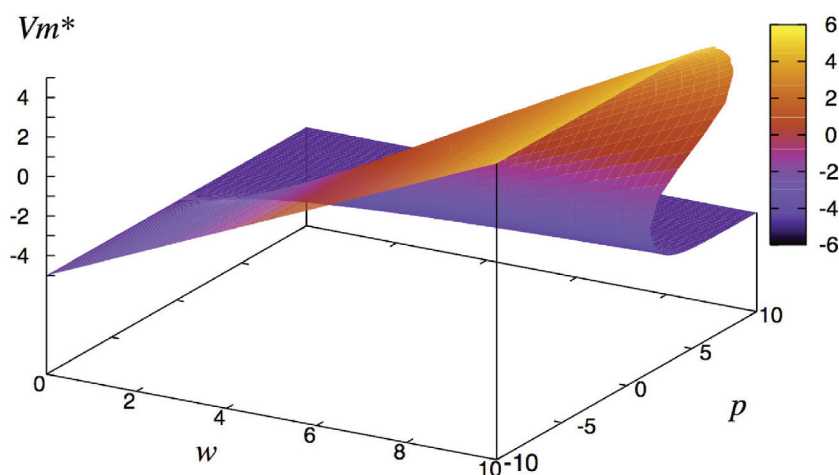
Stability diagram. The system has one fixed point when $w \leq w_c$, two fixed points when $w > w_c$ and $p = p_a(w)$ or $p = p_b(w)$, or three fixed points when $w > w_c$ and $p_a(w) < p < p_b(w)$. The region with three fixed points has two stable and one unstable fixed points. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

2.2.3. Remark 1

Biological studies have shown at least two conditions that influence how motor commands are initiated: (1) the sensory input to MC must be strong enough to initiate positive feedback between MC and CN, which we speak of as the loop's threshold; (2) the PC can not initiate motor commands by itself unless the external stimulus inputs to MC are strong enough, but it can terminate commands by itself.

The first biological condition indicates that the synaptic weight w must be greater than some threshold so that positive feedback can be initiated and sustained. The

FIGURE 5



Cusp catastrophe surface.

second biological condition implies that (a) w has to be greater than the cusp point $w_c = 5.27$ and (b) p has to be greater than a lower threshold p_a . From now on, to be consistent with the biological evidence, we will suppose that $w > w_c$ and $p > p_a$. Specifically, in the following illustrations, w will take the value of $w = 10$, and as a result, p_a will take the value of $p_a = 1.8$.

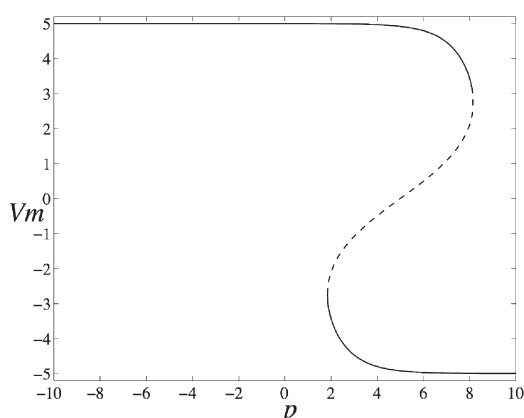
2.2.4. Remark 2

A limitation of the model is that the intensity range of a motor command could be too narrow to effectively control

the velocity of movement. To see the limitation, first we need to make it clear what we mean by the range of intensity. Recall that the intensity of a motor command is the output of MC (i.e. $R_m = f(V_m)$). However, this only makes sense when the motor command gets initiated, which in turn means (1) the PC is in the bistable range $p \in (p_a, p_b)$, and (2) the membrane potential of MC, V_m , is in the active state. Denote by $R_m(p)$ the intensity of a command when PC is in the range $p \in (p_a, p_b)$. Then the range of intensity can be defined as the difference between the maximal and minimal values that $R_m(p)$ can take.

Now we explain the limitation using Figure 6. The figure shows the bistable range for p is $(1.8, 8.2)$. For this range, the maximal equilibrium value that V_m can take is $V_m^*|_{p=1.8} \approx 5$, and the minimal equilibrium is $V_m^*|_{p=8.2} \approx 3$. The range of intensity then is $f(5) - f(3) \approx 0.04$. It is possible that the range of intensity can get widened by increasing the synaptic weight w between MC and CN. However, because oscillatory activity can occur at high gains, there is an upper limit to w . Therefore, in our current model, the range of command intensity is not biologically plausible. In future work we will revise the model (especially the activation function) to overcome this limitation.

FIGURE 6

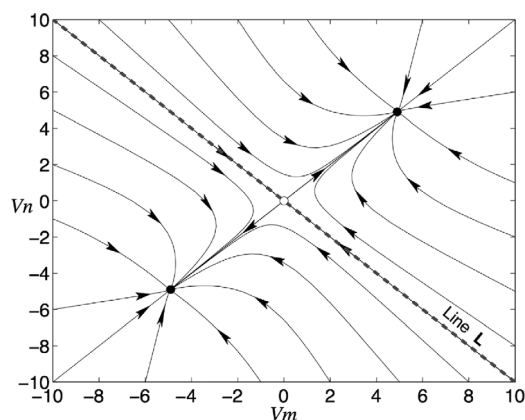


Cross-section of the cusp surface at $w = 10$.

2.2.5. Phase Portrait

Recall that our task is to find under what conditions motor commands will be initiated or terminated. Through the above bifurcation analysis, we have found that the synaptic weight w has to be larger than a threshold ($w > w_c$) and the inhibition from PC has to be larger than a lower threshold ($p > p_a$). We also know that when p is in the range $p_a < p < p_b$, the system is bistable: it can stay in the

FIGURE 7



Phase portrait when PC firing is low ($p = 5$). [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

quiescent state, or be driven into an active state, depending on the initial system state (i.e., the initial values of V_m and V_n).

Now we want to know how the initial state of the system decides the eventual state of the system. To answer this question, we plot a phase portrait for the case of PC being in disinhibition state, as shown in Figure 7. In the figure, the separatrix line L (dashed line) divides all possible initial states of the system into two classes: above line L, the system will converge to the active state; below line L, the system will converge to the quiescent state.

2.3. Time Course of Individual Movement Commands

Voluntary motor commands are driven by a well-studied sequence of changes in the activity of neurons that comprise the CB module. In this section, we demonstrate how the minimal model of the CB module can simulate the most salient properties of neural dynamics associated with the programming of motor commands. In what follows we present a step-by-step account of the simulated changes in modeled neural activity along with an explanation of the relevant physiological counterparts. In a further analysis, the results from a number of simulations show that the model can emulate cerebellar control over the duration of motor commands in a manner consistent with physiological data. The model, however, fails to properly control the intensity of motor commands. Reasons for this shortcoming are discussed.

Prior to the programming of a motor command, the CB module is found in a resting state, shown from 0 to 100 ms (see Figure 8). During this time PC neurons are found in a spontaneous state of high discharge rate, inhibiting

the buildup of positive feedback in the CB module. MC and CN membrane potentials are at equilibrium in a quiescent state ($V_m = -5$, $V_n = -8.9$), and there is no external sensory input.

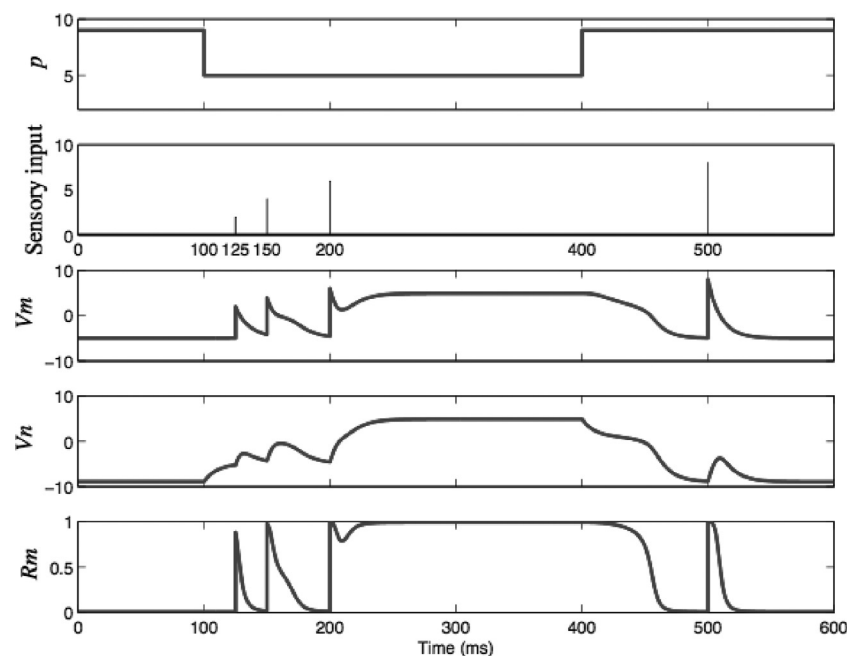
The programming period of the motor command is simulated from 100 to 400 ms, including the initiation, execution, and termination of the motor command (see Figure 8). During the programming period the firing rate of the PC drops and remains low until it is the time to terminate the movement. Note how the reduction in PC activity causes a change in CN membrane potential (V_m) from -8.9 to -5 , while the MC membrane potential (V_m) shows nearly no change. The physiological significance of this stage is to prepare the network for generating a motor command. The reduction in PC inhibition, allows the microscopic loop between MC and CN to achieve the high firing rates necessary for driving a motor command, pending the requisite sensory input.

The system is now prepared to initiate the motor command. The initiation stage is simulated from 125 to 200 ms. During this time the MC neuron will receive external sensory inputs allowing for sensory-motor integration. The sensory inputs to MC are shown as spikes in the second plot of Figure 8. There are two external inputs at 125 and 150 ms. Since these two inputs are not sufficiently intense, the neurons MC and CN go quickly back to the previous (quiescent) equilibrium state ($V_m = -5$ and $V_n = -5$), and no motor command is initiated. A third sensory input is simulated at 200 ms. This input is sufficiently strong to establish positive feedback in the microscopic loop responsible for driving the motor command. In Figure 7, this transition is equivalent to a shift in the state of the system to above the separatrix line L, which results in a fast convergence onto the active fixed point. The effect is that MC and CN begin to rapidly increase their discharge rates, and consequently transit from the quiescent state to the active state ($V_m = 5$ and $V_n = 5$). This marks the beginning of the execution phase of the motor command.

Finally, the motor command must end at the desired point in time. The termination of the motor command is modeled at 400 ms. This is marked by the quick return of the PC neuron to a high rate of inhibition to the loop. As a consequence, MC and CN return to the resting state ($V_m = -5$, $V_n = -8.9$), and thus the motor command is terminated. The inhibition from the PC is sufficiently intense to prevent a re-initiation of the motor command even in the presence of a strong external input. This is shown at 500 ms, where a fourth sensory stimulus is simulated. This final sensory input only effects a temporary change in activity of the module, demonstrating the importance of programming for initiating and terminating motor commands.

The simulation graphed in Figure 8 represents the time course of a single motor command with a fixed duration.

FIGURE 8



Time course for one CB module. Considering that the abstractness of the model, the numbers shown in the graphs mainly convey the information about how the activities of the neurons change over time. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

Studies have shown that the neurons here modeled are responsible for modulating both the velocity and duration of movements [11,13,15]. The duration and intensity of motor commands in the red nucleus have been correlated with the duration and velocity of movements [13]. It is also well known that the activity of red nucleus cells is in turn modulated by inhibitory input to CN from PCs [7]. Therefore, the duration of pauses in PC firing should be well correlated with the timing of the movements. To analyze how well the model captures these characteristic firing patterns, we performed a series of simulations where the duration and the magnitude of PC disinhibition to the loop were varied separately. The graph on the left in Figure 9, shows a linear relationship between PC pause duration and the duration of the motor command. This is consistent with physiological findings [11]. However, the relationship between the magnitude of PC disinhibition and the resulting intensity of the motor command is approximated by a step function (Figure 9, right panel). Instead, as discussed in the previous section, we should expect the PC to control a wide range of command intensities, which would correspond to a large range of different possible movement velocities.

3. SIMULATING ONE-DIMENSIONAL MOVEMENTS

The NM system is responsible for translating motor commands into the muscle forces that allow limbs to interact

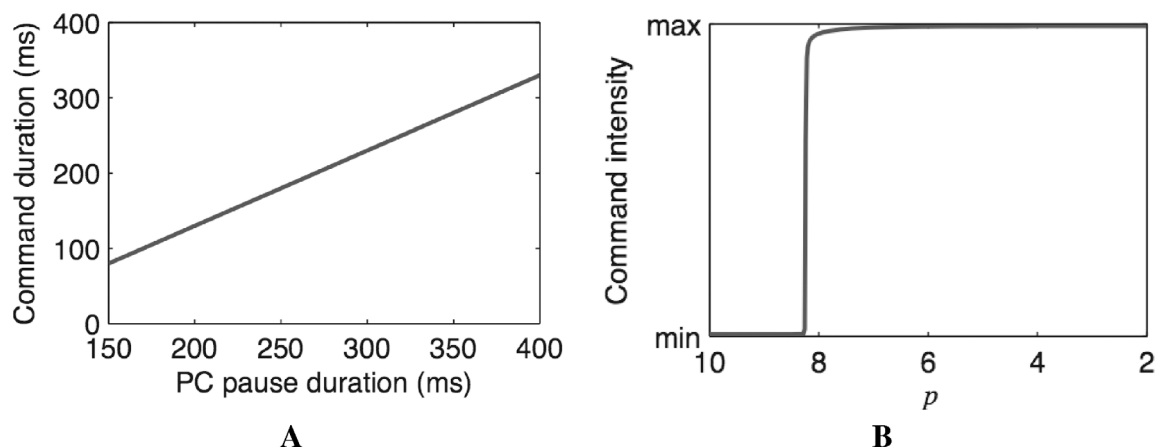
with the world. To facilitate interfacing with the above model of motor command generation, we will assume that each muscle in combination with its sensory feedback (stretch reflex) comprises an individual NM module. One dimensional movements typically involve the coordinated activity of a pair of muscles that act as antagonists. It is convenient to consider this pair of muscles as producing forces related to their own lengths, in a muscle-based coordinate system (l_s and f_s in Figure 10). These muscle forces then move the mass of the limb in relation to an external coordinate system.

3.1. Translation of Motor Commands into Movement by NMs Modules

While the intensity of motor commands recorded from the brain correlates well with the velocity of movement ([6,12,13] and Section 2), the relationship is highly nonlinear. This is expected since the NM system has prominent nonlinearities that have a strong influence on movement control [16]. Before considering these complexities, let us first consider the simple elastic, or spring-like, behavior of the NM system.

Muscles have spring-like properties and the sensory feedback that regulates their activation tends to linearize the stiffnesses K of the elastic behaviors [17]. Muscle forces are thus approximately equal to

FIGURE 9



Motor command characteristics of a CB module. (A) Command duration vs. PC pause duration (programming period), where PC value p is fixed to 5. (B) Command intensity vs. PC value p , where PC pause duration is fixed to 300 ms. PC value p is set to range from 2 to 10 in that p has to be greater than the lower threshold p_a which is 1.8 (see Remark 1 in Section 2.2). [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

$$f = K(1 - \lambda)$$

where the λ s represent the thresholds of the stretch reflexes. As muscles cannot push, f cannot be less than zero. Spring-like properties are the essence of the equilibrium point hypothesis of motor control [18–20].

Within the framework of the equilibrium point hypothesis there exists a threshold length λ for force production under static conditions. Under dynamic conditions, one must also include a velocity-dependent viscous component of force that dampens movements so as to minimize oscillations. Most NM models assume that damping is linearly related to velocity, in spite of the fact that length and velocity feedback are highly nonlinear in NM systems. Houk et al. [21] synthesized experimental findings from animal and human studies to formulate a highly nonlinear

model of the NM system. The present model, expressed in the equation below, amounts to a simplified version of the Houk–Fagg–Barto model.

$$f = K(1 - \lambda) + B(\dot{l})^{1/5}$$

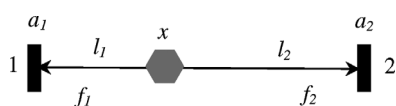
The second term on the right is a nonlinear damping force called fractional power damping (FPD). FPD is particularly effective in the control of rapid movements. Barto et al. used the FPD model in a computational study of the predictive control of movement by the CB [22].

The mechanism of motor command generation for two CB modules, controlling an agonist-antagonist pair of NM modules (see Figure 10), is very similar to that of one CB module. Let V_{m_i} and V_{n_i} ($i = 1, 2$) denote the membrane potentials of the MC and CN neurons in the CB modules. The model of two CB modules is given by

$$\begin{cases} \dot{V}_{m_i} = -V_{m_i} + w/(1 + e^{-V_{m_i}}) - 5 \\ \dot{V}_{n_i} = -V_{n_i} + w/(1 + e^{-V_{n_i}}) - p_i \end{cases} \quad (i = 1, 2)$$

where p_i is the PC value of the i th module. In this model of two CB modules, there is no interaction between the two modules; however, in our model for planar movements to be presented in the next section, the modules will interact with each other. Now supposing that the goal is for the limb to move leftward; then module 1 will be the agonist and module 2 the antagonist. To this end, we sup-

FIGURE 10



A simple limb model consisting of a pair of agonist-antagonist neuromuscular (NM) modules. Each module has a fixed anchor (bar) and is attached to a limb mass (hexagon). [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

pose that during the programming period, the PC values of the two modules change according to

$$\begin{cases} p_1 = \hat{p} - \alpha \\ p_2 = \hat{p} + \alpha \end{cases}$$

where \hat{p} gives the high spontaneous discharge of PC before and after the programming period, and α is a positive parameter.

As to the external stimulus, we suppose that only the agonist module (i.e., module 1) is stimulated; that is, the membrane potential V_m of module 1 will get a pulse-like jump (see Figure 11).

Initiating and controlling movement requires a change in the threshold λ , which is in turn determined by the motor command. Therefore, we need some mechanism to convert the motor command to the threshold λ . From the above force equation, we know that a large force requires a small λ , and we also know that a large force requires a strong motor command. To account for this we make the conversion as follows:

$$\lambda = \beta - R_m \quad (7)$$

where β is a positive parameter. When the modules pull the limb of mass M horizontally, the dynamics of the limb movement can be described by

$$M\ddot{x} = f_2 - f_1$$

where M is the mass of the limb, x the position of the limb, and f_i represents the pulling force from the i th module.

Let a_i be the anchor position of the i th module, and suppose that $a_2 > x > a_1$, then the muscle lengths are

$$\begin{aligned} l_1 &= x - a_1 \\ l_2 &= a_2 - x, \end{aligned}$$

and the pulling forces of the modules become

$$\begin{aligned} f_1 &= K(x - a_1 - \lambda_1) + B(\ddot{x})^{1/5} \\ f_2 &= K(a_2 - x - \lambda_2) - B(\ddot{x})^{1/5}. \end{aligned}$$

Finally, the limb movement dynamics can be described by

$$M\ddot{x} = K(-2x + a_1 + a_2 - R_{m1} + R_{m2}) - 2B(\ddot{x})^{1/5} \quad (8)$$

However, to be realistic, we require that both force f_i and one of its terms, the elastic force $K(l_i - \lambda_i)$, are always nonnegative. Considering this requirement, the revised dynamics will be more complicated than Eq. (8).

3.2. Simulation of Movements

Figure 11 shows the time course of the simulated limb movement generated by an agonist-antagonist pair of CB-NM modules. In the figure, the agonist module is denoted by 1, and the antagonist is by 2. The figure includes seven panels, which can be divided into three groups: CB group (PC, V_m , V_n , and R_m), NM group (λ and ‘force’), and movement group (‘velocity’ and ‘position’).

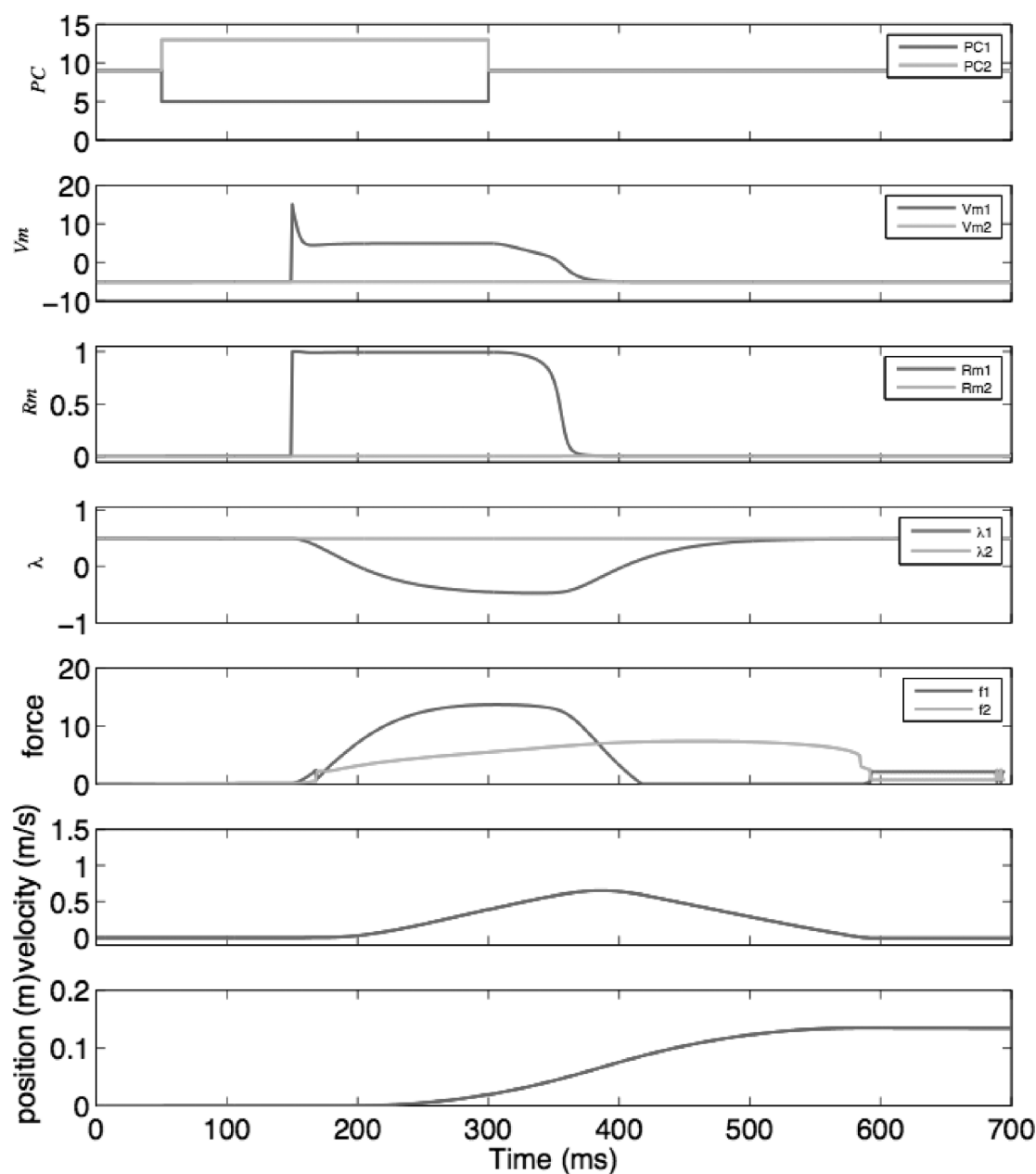
The CB group shows that the behavior of the agonist CB module 1 is the same as in the simulation of the previous section. After the initiation (the jump of V_m), CB module 1 can generate a motor command. The antagonist CB module 2 receives high inhibition from PC₂ during the programming period and no external stimuli to the MC neuron. As a result, it is kept from generating an antagonist motor command.

The NM group shows the thresholds (λ_1 and λ_2) and forces of the two NM modules. When receiving motor commands from CB modules, NM modules need to first translate the commands R_m into λ according to the inverse relation between λ and R_m [see Eq. (7)]. In our simulation, we apply a second-order low-pass filter to λ in order to accomplish a smoother transition. We suppose that at rest (i.e., at the beginning of the simulation) the muscle lengths of both NM modules are equal to their corresponding λ s, resulting in no forces. When the agonist NM module 1 receives an active motor command from its corresponding CB agonist module, its threshold (λ_1) starts to decrease. The decreased threshold then enables the agonist NM module to generate a force that pulls the limb. While the limb is being pulled towards the anchor position of the agonist module, the length of the module, l_1 , becomes shorter. At the same time, the length of the antagonist module 2, l_2 , becomes longer. As a result, when l_2 becomes larger than λ_2 , the antagonist module will also be able to generate a pull force, regardless of the fact that there is no active command from its corresponding CB module. As long as the agonist force f_1 is larger than the antagonist force f_2 , the velocity of the limb will increase. When the two forces cancel out, as illustrated by the intersection between the two force curves, the movement velocity will reach its maximum. After that, the velocity starts to decrease until the limb comes to rest.

3.3. Movement Characteristics of a Pair of CB-NM Modules

At the end of Section 2 (see Figure 9), we demonstrated that the CB module command duration increases with an increase in pause duration of PC inhibition (i.e., programming duration), and that the command intensity remains relatively constant as long as the PC value is below some threshold. Here we are interested in investigating how this

FIGURE 11



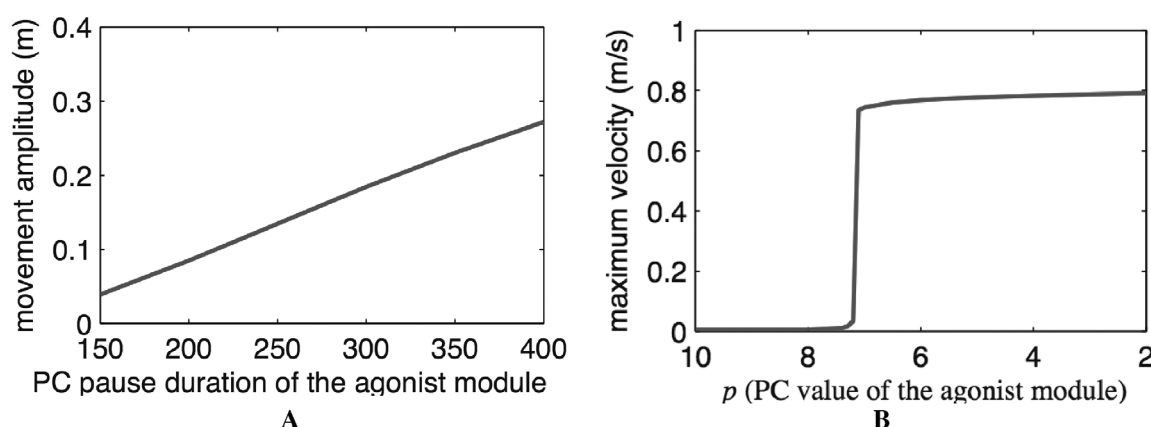
Simulation of one-dimensional limb movement generated by a pair of agonist–antagonist CB-NM modules. The limb movement group includes two panels: the time courses of the movement velocity and position. The bell-shaped profile of movement velocity shows the salient characteristics of rapid movements [23]. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

result translates into movement. For a pair of CB-NM modules we explore two relationships: (1) the relationship between movement amplitude and PC programming duration, and (2) the relationship between movement velocity and the PC value of the agonist module.

To answer the first question, we fix the PC values of the two CB modules to the same as in Figure 11 (i.e., 5 for the agonist module and 13 for the antagonist), but let the PC pause duration vary from 100 to 400 ms. The result is given in panel A of Figure 12. It shows that the

F12

FIGURE 12



Movement characteristics of a pair of CB-NM modules. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

movement amplitude is well controlled by the PC programming duration. The explanation of this result is that the longer the PC programming duration, the longer the command duration of the agonist CB module, and so the longer the effective force duration of the agonist NM module.

To answer the second question, we fix the PC pause duration to 300 ms, but let the PC value of the agonist CB module, p , change from 2 to 10. The result is given in panel B of Figure 12. It shows that the movement velocity is poorly controlled by the PC value of the agonist module: when the PC value is below some threshold, the movement velocity only increases a little bit as the PC value decreases. The explanation is that the response intensity, as illustrated in the right panel of Figure 9, remains nearly constant when the PC value is under the threshold. In summary, CB modules are able to control the amplitude of a movement well, but they control movement velocity poorly.

4. SIMULATING PLANAR MOVEMENTS

In this section, we build a toy limb plant model, controlled by an array of CB modules, to simulate the center-out task, a classic task used to study planar movements in different directions [24]. The task requires a subject (e.g., a rhesus monkey) to move rapidly from a starting point in the center of a workspace to one of eight radially symmetric targets [Figure 13(A)].

Our limb NM model has eight NM modules, labeled from 1 to 8 in Figure 13(B). As illustrated, each NM module has a fixed anchor at the one end, and is attached to a

shared, movable limb mass at the other end. Using the center starting point as reference, for each module we can define a direction that points to its anchor from the center; such a direction will be referred to as module direction or agonist direction. When receiving motor commands from CB modules, the NM modules will generate forces that act in the module direction on the limb mass to produce movement [Figure 13(C)].

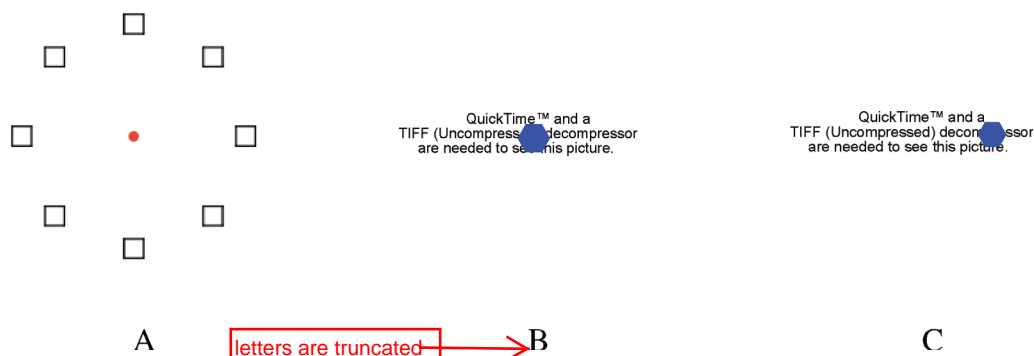
Unlike what we did in the previous section on simulating one-dimensional movements using Matlab, here we use NetLogo [25] to simulate planar movements. NetLogo is a free, cross-platform multi-agent modeling tool that enables users to easily and quickly build their models. It is particularly well suited for modeling complex systems consisting of hundreds or thousands of agents, thus making it possible to explore the connection between the micro-level behavior of individuals and the macro-level patterns that emerge from the interaction of many individuals. When CB-NM modules are modeled as agents that can adapt their behavior through feedbacks such as whether a desired target is reached or not, we may model very complicated movements while keeping the underlying CB-NM modules as simple as possible.

The interface of our NetLogo implementation is given in Figure 14 (see [26] for software). The interface includes three panels: view (left-top), control (middle-top), and plots (right and bottom). The view panel shows how the limb object moves under the forces of the NM modules; the control panel enables a user to set up various parameters; and the plot panel gives various time course plots.

F13

F14

FIGURE 13



(A) Illustration of the center-out task that requires a subject to move from the center (red dot) to one of the eight targets (squares). (B) A toy limb plant model that involves eight NM modules attached to the limb mass in the center (blue hexagon). (C) In a simulation of the center-out task, the limb mass moves to the right-most target under the forces of the eight modules.

We will explain in the next two subsections the mechanisms of planar motor commands generation and how the commands are transformed into planar movements.

4.1. Planar Motor Commands

The mechanism of motor command generation for eight CB modules is similar to the case of one or two CB modules: they are programmed by an array of PCs and initiated by a sensory stimulus. However, for multiple modules such as eight CB modules, we need to specify: (1) how the modules interact with each other; (2) how the PCs change their values during the programming period; and (3) how the motor command is initiated by a sensory stimulus.² In this article, the specification is based on the conceptual schema described by Houk and Mugnaini [3].

Firstly, according to the Houk–Mugnaini schema, we suppose that each CB module is reciprocally connected with its adjacent modules. Let V_{m_i} and V_{n_i} ($i = 1, 2, \dots, 8$) denote the membrane potentials of the eight MC and eight CN neurons, respectively. Let w be the synaptic weight between a MC and a CN in the same CB module (referred to as self-weight), and v the weight between a MC (or CN) and the CN (or MC) of adjacent modules (referred to as neighbor-weight). The model of eight CB modules is given by³

$$\begin{cases} \dot{V}_{m_i} = -V_{m_i} + wf(V_{n_i}) + vf(V_{n_{i+1}}) + vf(V_{n_{i-1}}) - 5 \\ \dot{V}_{n_i} = -V_{n_i} + wf(V_{m_i}) + vf(V_{m_{i+1}}) + vf(V_{m_{i-1}}) - p_i \end{cases} \quad (i=1,2,\dots,8)$$

²For two CB modules, because of its simplicity, we can specify them by hand.

³In the equations, V_{m_9} (or $V_{m_0}, V_{n_9}, V_{n_0}$) is the same as V_{m_1} (or $V_{m_8}, V_{n_1}, V_{n_8}$).

Secondly, we suppose that during the programming period, the PC value of the i th module, p_i , changes according to

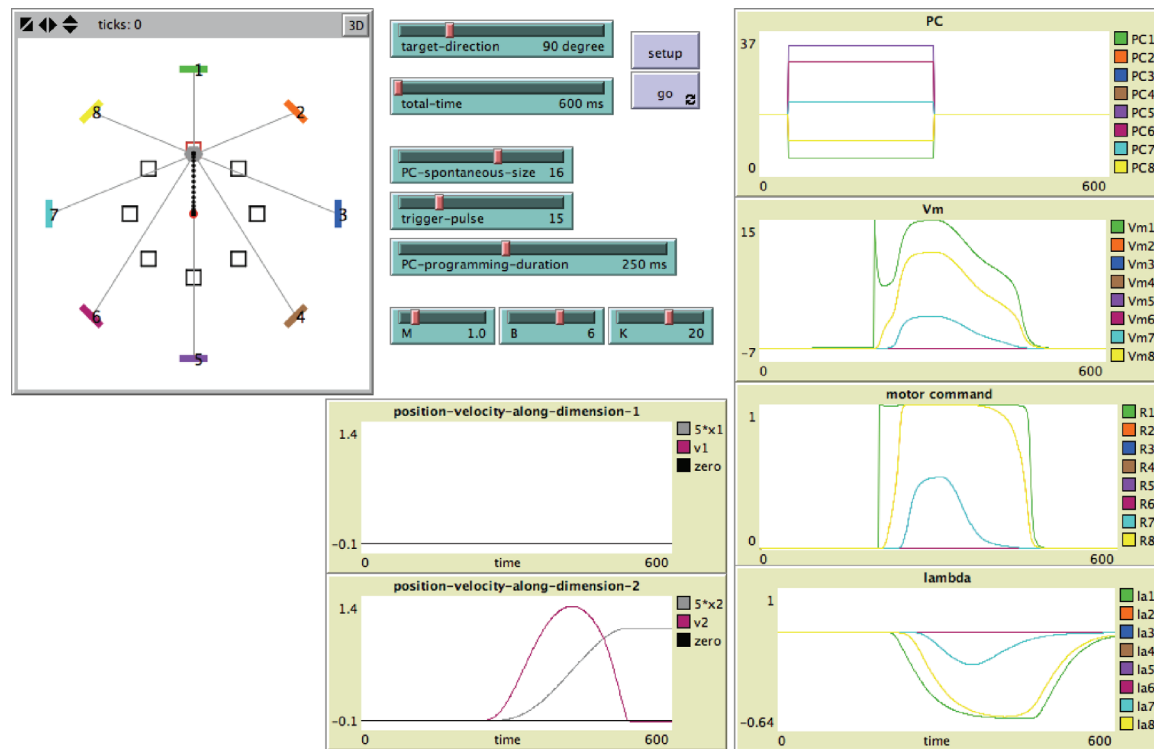
$$p_i = \hat{p}(\alpha - \beta \cos(\theta_i)).$$

In the equation, \hat{p} gives the high spontaneous discharge of PC before and after the programming period; it is the same for all modules. The number θ_i is the angle between the i th module direction (which is assumed to be the same as the i th NM module direction defined above), and the target direction which points from the starting center to the target. For example, if the target is located at the right-most position, then the angle between the target direction and the module 1 (upward) is $\pi/2$. The two parameters $\alpha > 0$ and $\beta > 0$ specify the range of PC values. In our simulation, they are $\alpha = 1.2$, and $\beta = 0.9$. The idea behind the PC value specification is obvious: the more aligned the target direction to a module direction, the lower the PC value, resulting in less inhibition, the more likely the motor command is activated.

Thirdly, when motor commands are initiated by an external sensory stimulus such as a visual cue of a target, we suppose that only the CB module that is the most aligned to the target direction is able to get stimulated; that is, the membrane potential V_m of the module will get a pulse-like jump. When two or more modules are equally aligned, then a random one gets stimulated.

The top three plots on the right panel of Figure 14 show the time course of the activities of PC, MC (V_m), and the motor command R_m in a simulation. The self-weight w and neighbor-weight v are set to $w = 10$ and $v = 5$. In the simulation, the target is the top-most one, module 1 is best aligned with the target direction, followed by modules

FIGURE 14



Interface of the center-out task simulation model. In the plots on the right, because of the symmetry along the direction of module 1 (or module 5), there are three pairs of overlapped time courses: modules 2 and 8, modules 3 and 7, and modules 4 and 6.

2 and 8, and so on. From the figure, we can see that the value of PC_1 drops the most during the programming period. When an external stimulus input arrives at 150 ms, the membrane potential of module 1, V_{m1} , gets a pulse-like jump, causing the membrane potentials of its neighboring modules (modules 2 and 8) to increase. After around another 50 ms, only the three most aligned modules (modules 1, 2, and 8) can reach the maximum of their motor command intensity.

4.2. Transformation of Motor Commands into Planar Movements

Our simulation involves eight modules, each being able to generate a muscle force to pull the limb. When they pull the limb of mass M in the planar space, the dynamics of the limb movement can be described by

$$M\ddot{\mathbf{x}} = \sum_i^8 -\frac{\partial l_i}{\partial \mathbf{x}} f_i$$

In the equation, $\mathbf{x} = (x_1, x_2)$ is the position of the limb, f_i the pulling force from the i th module, and l_i the length of the muscle of the i th module (see Figure 15).

Let $a_i = (a_{i1}, a_{i2})$ be the anchor position of the i th module, then the muscle length, l_i , is just the distance between the limb object position, \mathbf{x} , and the anchor position of the i th module, a_i . That is,

$$l_i(\mathbf{x}) = \sqrt{(x_1 - a_{i1})^2 + (x_2 - a_{i2})^2}$$

Given the muscle length of the i th module, l_i , and the threshold length, λ_i , the pulling force from the i th module is then given by $f_i = K(l_i - \lambda_i) + B(\dot{l}_i)^{1/5}$ along the direction of

$$-\frac{\partial l_i}{\partial \mathbf{x}} = \begin{pmatrix} (a_{i1} - x_1)/l_i \\ (a_{i2} - x_2)/l_i \end{pmatrix}.$$

To be realistic, we require that both the force f_i and one of its terms, the elastic force $K(l_i - \lambda_i)$, are always nonnegative.

The module velocity, \dot{l}_i , can be converted, in terms of (x_1, x_2) , to

$$\dot{l}_i(\mathbf{x}) = \frac{\partial l_i}{\partial x_1} \dot{x}_1 + \frac{\partial l_i}{\partial x_2} \dot{x}_2 = \frac{(x_1 - a_{i1})\dot{x}_1 + (x_2 - a_{i2})\dot{x}_2}{l_i}.$$

FIGURE 15

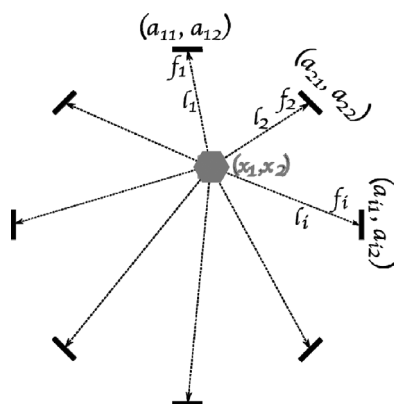


Illustration of the eight muscles pulling the limb mass. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

Thus, using the above limb movement dynamics description, the transformation of the motor commands to movements is completed.

4.3. Simulation of Planar Movements

The main purpose of our simulation is to investigate whether movement directions are well controlled or not. Figure 16 shows movement trajectories of reaching each of the eight targets. The starting point for each movement is at the center, and the target positions are at the centers of the open squares. The position of the limb mass at every 10 ms is shown as a dot. The result shows that the movement directions are nicely controlled by the motor commands from the eight CB modules. It also shows that trajectories are straight-line paths. This is expected because all targets and all module directions are symmetric.

5. DISCUSSION AND FUTURE DIRECTIONS

The model of a microscopic loop between the MC and the CN illustrated in Figure 2 has unique dynamics that mimic those of the biological system for motor command generation. Three features of motor command generation that have been challenging to comprehend from a neurophysiological perspective are matched by the nonlinear dynamics of microscopic modules: (1) motor commands can be programmed well in advance of the actual initiation of the command; (2) commands that may continue for variable durations can be initiated by a brief sensory input; and (3) the durations of commands are controlled in an independ-

ent manner by inhibitory input from PCs in the cerebellar cortex. Furthermore, the direction of planar movements is well controlled.

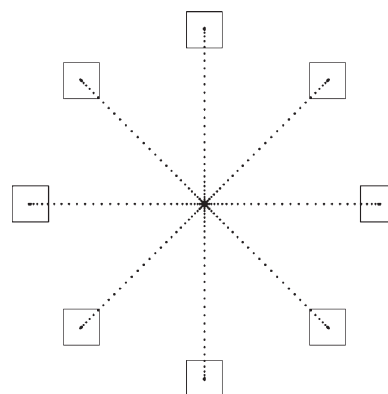
5.1. Limitations of the Present Model

One primary feature of motor commands that **was not mimicked by the present model** is the large range of intensities of discharge rate that has been observed biologically to control movement velocity [13]. We concluded in Sections 2 and 3 that this limited range of R_m results from our use of the conventional neural network activation function, the sigmoid function. That function not only limits the expressed range of MC membrane potential V_m , but it further flattens the resultant range of firing rates R_m . In [27], Kim and Wu have shown that other simple activation functions can overcome some of the limitations of the sigmoid function. In the future, we plan to incorporate the unique nonlinear properties of synapses mediated by NMDA receptors [9]. These nonlinearities are expected to have major useful influences on the nonlinear dynamics of microscopic modules.

The loop between the MC and the basal ganglia (see Section 1) was ignored in the present model. This loop is particularly important in the embodiment of a particular motor command [28]. By embodiment we mean the selection and/or the initiation of a particular command either in response to a particular external stimulus (sensory event) or in response to an internal contingency (a desire to move). In this article, we assume that the loop through the basal ganglia has already selected the sensory inputs we simulated in Figures 8 and 11, thus allowing them to initiate the command.

Another limitation of the present work is that in the simulation some parameters were set by hand. For exam-

FIGURE 16



Trajectories of reaching the eight targets in the center-out task.

ple, the durations of PCs were set manually in order for different targets to be reached **nicely** though not perfectly—which can be seen from the movement trajectories shown in Figure 16. In future work we plan to enhance the model with the ability to learn to adjust the parameters so that the targets can be reached with improvement over the time.

5.2. Modularity and Agent-Based Modeling

Understanding the mind as a modular system dates back to the very beginnings of modern psychology and neuroscience and is still the predominant theoretical approach [4,29,30]. If modularity is the driving organizational principle of the brain, then we can specify a research question: given a particular computational goal of the brain, what are the sizes, structures, purposes and forms (anatomical or functional) of the modules involved? In this study we approached this question in a bottom-up fashion by modeling the generative processes that give rise to voluntary motor commands using the DPM framework espoused in [2]; this approach can lead to modular models that enable a deeper understanding of the neural computations underlying thinking (see Section 1 and [28]). Voluntary motor commands are achieved through the aggregate behavior of a large number of microscopic modules; each of these modules generates the individual limb motor commands that are sent from brain cells, in the MC and red nucleus, to the spinal cord. This array of microscopic modules forms a macroscopic module that is capable of directing complex movements of the arm. Macroscopic modules like this are the basic ingredients of DPMs [2].

Each of these microscopic modules can be **though** of as an agent with its own properties and actions. This is useful because the system as a whole can then be considered to be a multi-agent system. A multi-agent system is a set of autonomous agents each acting individually based on local information. Together these agents are capable of achieving global goals. Moreover, **the macroscopic modules can also be thought of as agents**, which **are** comprised of multitudes of microscopic agents. In other words, the macroscopic modules are multi-agent systems of microscopic agents, and these macroscopic modules are themselves part of a larger multi-agent system. Multi-agent systems have proved useful in examining a wide variety of natural phenomena from the behavior of ant colonies to collections of gas molecules to the organization of financial markets [31]. Our current understanding of the emergent behavior (behavior which emerges from the interactions within multi-agent systems) of decentral-

ized systems is in large part made possible by agent-based modeling. Agent-based models have not been previously used in neuroscience to understand the concerted activity of brain modules in the generation of motor commands. By taking advantage of the modularity of the neural system, we can leverage agent-based modeling to create powerful descriptions of the complex neural system. Such an approach provides an exciting and promising avenue for testing hypotheses about the modular organization and functioning of the motor system. Using a preliminary agent-based model, we have presented evidence that CB modules acting together provide an appropriate computational framework for generating the essential properties of motor behavior.

6. CONCLUSION

We have presented a computational model that is inspired by converging evidence regarding the anatomy and functioning of the neural substrate for generating voluntary limb movements. This evidence points to the existence of a modular neural system composed of cortical and subcortical brain regions that act in a concerted fashion to produce motor commands. However, because of the current limitations of experimental neuroscience, little is understood about how these highly nonlinear modules actually function. Here we have taken a theoretical approach based on nonlinear dynamics for testing extant hypotheses about how such a system controls voluntary movements. Our model of a single CB module generates simulated neural dynamics that are consistent with salient physiological evidence. Further, we have demonstrated using an agent-based modeling approach that a collection of CB modules is capable of reproducing motor behavior that is often studied in movement psychophysics. Taken together, these results provide a promising theoretical framework for understanding the fundamental neurobiological mechanisms that underlie the control of voluntary limb movements.

Acknowledgments

This research was made possible by grants NS44837 and P01-NS44383 from the National Institute of Neurological Disorders and Stroke. The authors thank Ilana Nisky for her initial implementation of the Matlab code for planar movements, and both Ilana and Amir Karniel for their discussions and helpful suggestions for improving this article.

REFERENCES

1. Middleton, F.A.; Strick, P.L. Basal ganglia and cerebellar loops: Motor and cognitive circuits. *Brain Res Brain Res Rev* 2000, 31, 236–250.
2. Houk, J.C. Agents of the Mind. *Biol Cybernet* 2005, 92, 427–437.
3. Houk, J.C.; Mugnaini, E. Cerebellum. In: *Fundamental Neuroscience*; Squire, L.R.; Bloom, F.E.; Roberts, J.L., Eds.; Academic Press: New York, 2003.
4. Houk, J.C.; Wise, S.P. Distributed modular architectures linking basal ganglia, cerebellum, and cerebral cortex: Their role in planning and controlling action. *Cereb Cortex* 1995, 5, 95–110.
5. Berthier, N.E.; Singh, S.P.; Barto, A.G.; Houk, J.C. Distributed representation of limb motor programs in arrays of adjustable pattern generators. *J Cogn Neurosci* 1993, 5, 56–78.
6. Houk, J.C.; Keifer, J.; Barto, A.G. Distributed motor commands in the limb premotor network. *Trends Neurosci* 1993, 16, 27–33.
7. Keifer, J.; Houk, J.C. Motor function of the cerebellorubrospinal system. *Physiol Rev* 1994, 74, 509–542.
8. Hua, S.E.; Houk, J.C. Cerebellar guidance of premotor network development and sensorimotor learning. *Learn Mem* 1997, 4, 63–76.
9. Jiang, M.C.; Alheid, G.F.; Nunzi, M.G.; Houk, J.C. Cerebellar input to magnocellular neurons in the red nucleus of the mouse: Synaptic analysis in horizontal brain slices incorporating cerebello-rubral pathways. *Neuroscience* 2002, 110, 105–121.
10. Holdefer, R.N.; Houk, J.C.; Miller, L.E. Movement-Related Discharge in the Cerebellar Nuclei Persists After Local Injections of GABA_A Antagonists. *J Neurophysiol* 2005, 93, 35–43.
11. Miller, L.E.; Holdefer, R.N.; Houk, J.C. The role of the cerebellum in modulating voluntary limb movement commands. *Archives Italiennes de Biologie* 2002, 140, 175–182.
12. Lamarre, Y.; Busby, L.; Spidalieri, G. Fast ballistic arm movements triggered by visual, auditory, and somesthetic stimuli in the monkey. I. Activity of pericentral cortical neurons. *J Neurophysiol* 1983, 50, 1343–1358.
13. Gibson, A.R.; Houk, J.C.; Kohlerman, N.J. Relation between red nucleus discharge and movement parameters in trained macaque monkeys. *J Physiol (Lond)* 1985, 358, 551–570.
14. Strogatz, S.H. *Nonlinear Dynamics and Chaos*. Westview Press: Cambridge, 2000.
15. Ebner, T.J. A role for the cerebellum in the control of limb movement velocity. *Curr Opin Neurobiol* 1998, 8, 762–769.
16. Wu, C.H.; Houk, J.C.; Young, K.Y.; Miller, L. E. Nonlinear damping of limb motion. In: *Multiple Muscle Systems: Biomechanics and Movement Organization*; Winters, J. M.; Woo, S.L.-Y., Eds.; Springer-Verlag: New York, 1990; pp 214–235.
17. Houk, J.C.; Rymer, W.Z. Neural control of muscle length and tension. In: *Handbook of Physiology—The Nervous System II*; Brooks, V.B., Ed.; American Physiological Society: Bethesda, MD, 1981; pp 257–323.
18. Feldman, A.G. Once more on the equilibrium-point hypothesis (lambda model) for motor control. *J Motor Behav* 1986, 18, 17–54.
19. Shadmehr, R. Equilibrium point hypothesis. In: *Handbook of Brain Theory and Neural Networks*, 2nd ed.; MIT Press: Cambridge, MA, 2002; pp 409–412.
20. Bizzi, E.; Hogan, N.; Mussa-Ivaldi, F.A.; Gister, S. Does the nervous system use equilibrium-point control to guide single and multiple joint movements? *Behav Brain Sci* 1992, 15, 603–613.
21. Houk, J.C.; Fagg, A.H.; Barto, A.G. Fractional power damping model of joint motion. In: *Progress in Motor Control, Vol. 2: Structure–Function Relations in Voluntary Movements*; Latash, M., Ed.; 2002; pp 147–178.
22. Barto, A.G.; Fagg, A.H.; Sitkoff, N.; Houk, J.C. A cerebellar model of timing and prediction in the control of reaching. *Neural Comput* 1999, 11, 565–594.
23. Karniel, A.; Inbar, G.F. The use of a nonlinear muscle model in explaining the relationship between duration, amplitude and peak velocity of human rapid movements. *J Motor Behav* 1999, 31, 203–206.
24. Georgopoulos, A.P.; Kalaska, J.F.; Caminiti, R.; Massey, J.T. On the relations between the direction of two dimensional arm movements and cell discharge in Primate motor cortex. *J Neurosci* 1982, 11, 1527–1537.
25. Wilensky, U. NetLogo. Center for Connected Learning and Computer-Based Modeling Northwestern University, Evanston, IL, 1999. Available at <http://ccl.northwestern.edu/netlogo/>.
26. Wang, J.; et al. Reciprocity between cerebellum and cerebral cortex NetLogo simulation model. 2008. Available at www.isrl.uiuc.edu/~junwang4/share/complexity/.
27. Kim, A.; Wu, C.H. A study of dynamic behavior of a recurrent neural network for control. In: *Proceedings of the 30th IEEE Conference on Decision and Control*, 1991; pp 150–155.
28. Houk, J.C.; Bastianen, C.; Fansler, D.; Fishbach, A.; Fraser, D.; Reber, P.J.; Roy, S.A.; Simo, L.S. Action selection in subcortical loops through the basal ganglia and cerebellum. *Philos Trans R Soc Lond B Biol Sci* 2007, 362, 1573–1583.
29. Granger, R. Engines of the brain: The computational instruction set of human cognition. *AI Mag* 2006, 27, 15–32.
30. Fodor, J.A. *The Modularity of Mind: An Essay on Faculty Psychology*; MIT Press: Cambridge, MA, 1983.
31. Wilensky, U.; Rand, W. *An Introduction to Agent-Based Modeling: Modeling Natural, Social and Engineered Systems with NetLogo*. MIT Press, in press.