Distributed Representation of Limb Motor Programs in Arrays of Adjustable Pattern Generators

N. E. Berthier, S. P. Singh, and A. G. Barto
University of Massachusetts, Amherst

J. C. Houk
Northwestern University Medical School

Abstract

■ This paper describes the current state of our exploration of how motor program concepts may be related to neural mechanisms. We have proposed a model of sensorimotor networks with architectures inspired by the anatomy and physiology of the cerebellum and its interconnections with the red nucleus and the motor cortex. We proposed the concept of rubrocerebellar and corticocerebellar information processing modules that function as adjustable pattern generators (APGs) capable of the storage, recall, and execution of motor programs. The APG array model described in this paper extends the single APG model of Houk et al. (1990) to an array of APGs whose collective activity controls movement of a simple two degree-of-freedom simulated limb. Our objective was to examine the APG array theory in a simple computational framework with a plausible relationship to anatomy and physiology. Results of simulation experiments show that the APG array model is capable of learning how to control movement of the simulated limb by adjusting distributed motor programs. Although the model is based on many simplifying assumptions, and the simulated motor control task is much simpler than an actual reaching task, these results suggest that the APG array model may provide a useful step toward a more comprehensive understanding of how neural mechanisms may generate motor programs. ■

INTRODUCTION

The concept that movements are controlled by motor programs has evolved from a long line of studies in experimental psychology (cf. Schmidt, 1988). The basic idea is that movement commands are centrally specified and then executed in essentially an open-loop manner. In other words, the system is said to operate in a feedforward mode, as opposed to using sensory feedback from the periphery. In his text on motor control and learning, Schmidt (1988) points out that this view is oversimplified since sensory information is ordinarily utilized in several ways. It is used to select parameters of a motor program before the program is initiated, it can modify ongoing programs in certain limited ways, and it guides the adaptive process that mediates motor learning. With these qualifications in mind, the idea that the brain commands movements by recalling and executing "quasi-feedforward" motor programs remains a very useful concept in motor control. Unfortunately, progress toward understanding the neural mechanisms that might implement this type of control has been slow.

In an attempt to relate motor program concepts to neural mechanisms, we have recently begun to explore the properties of sensorimotor networks with architectures inspired by the anatomy and physiology of the cerebellum and its interconnections with the red nucleus and the motor cortex (Houk, 1989; Houk, Singh, Fisher, & Barto, 1990). It is widely accepted that these brain regions are important in the control of limb movements (Kuypers, 1981; Ito, 1984), although relatively little attention has been devoted to probing how the different regions might function together in a cooperative manner (reviewed in the Discussion). Starting from a foundation of known anatomical circuitry and the results of microelectrode recordings from neurons in these circuits, we proposed the concept of rubrocerebellar and corticocerebellar information processing modules that function as adjustable pattern generators (APGs) capable of the storage, recall, and execution of motor programs.

According to this theory, motor programs are stored in the cerebellar cortex in the weights of parallel fiber synapses onto Purkinje cells, this being consonant with recent evidence regarding synaptic plasticity in the cere-

bellum (Ito, 1989; Houk & Barto, 1992). Training signals based on sensory information are transmitted by climbing fibers to guide the adaptive adjustment of parallel fiber synapses. After training 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, some of the parameters of the recalled programs are determined by sensory feedback while others are determined centrally. After a program has been selected, APG modules wait idle until the arrival of a trigger signal that starts program execution, analogous to a cue in a reaction-time task. The trigger is envisioned as a transient sensory (or central) input that initiates positive feedback in recurrent loops that exist between the cerebellar nucleus, the red nucleus, and the motor cortex. Positive feedback is thought to provide the driving force for the generation of the program, while its accuracy in direction and amplitude is considered to be regulated by inhibition from cerebellar Purkinje cells. Because Purkinje cells are modeled as bistable devices, motor programs are initially executed in a feedforward manner. Purkinje cells are postulated to detect when the endpoint of the movement is nearly achieved, whereupon they switch to activated states that inhibit positive feedback and thus terminate program execution. Many features of this model relate quite naturally to the ideas about motor programs proposed by experimental psychologists.

In our previous work we simulated only a single APG module controlling unidirectional motion (Houk et al., 1990; Sinkjaer, 1990), or a nonadaptive model of planar motion (Eisenman, Keifer, & Houk, 1991). In the present report we deal more directly with the problem of multidimensional motion and the distributed nature of the neural signals that control such movements. We are thus able to relate our theory to the body of work on directional tuning in populations of motor cortical and cerebellar neurons by Georgopoulos and colleagues (Georgopoulos, Kalaska, & Massey, 1982; Georgopoulos, 1988; Fortier, Kalaska, & Smith, 1989).

We designed the APG array model described here to investigate how multiple APGs might operate together to control multiple degree-of-freedom arm movements. The model's purpose is to help us investigate strengths and weaknesses of the APG theory of motor control and learning in a framework that is as simple as possible while containing enough detail to permit computer simulation. Consequently, the model is purposefully abstract and necessarily speculative; it is an early tool aiding the formation of a theory requiring much additional development and modification.

Although our research is informed by control methodologies used in robotics and in other models of motor control, we have explicitly avoided letting conventional control principles dictate the form of the model. These principles rely so heavily on mathematical convenience that they can be misleading as guides to thinking about biological control. Animal limb movement is accomplished by distributed nonlinear control mechanisms that may have very little in common with centralized controllers based on linear feedback principles that play a dominant role in control theory. Although one of the long-term objectives of our research is to produce a competent controller of a complex dynamic limb, we are trying to see what progress can be made by following principles suggested by anatomy and physiology instead of control theory. We used this philosophy in developing the APG array model described here. The theoretical analysis we present arose from our attempt to understand the behavior of the resulting model. It is significant, therefore, that this analysis revealed relationships between our model and the error backpropagation learning algorithm for artificial neural networks (Le Cun, 1985; Parker, 1985; Rumelhart, Hinton, & Williams, 1986; Werbos, 1974) and the adaptive control method known as feedback error learning (Kawato, Furukawa, & Suzuki, 1987; Gomi & Kawato, 1990). These relationships—which emerged in our analysis of the completed model—suggest how the theoretical principles underlying these methods might be reconciled with anatomy and physiology.

THE APG ARRAY MODEL

Model Overview

Figure 1 shows the pattern generator network and the muscle-arm system that it controls. The model has three parts: a neural network that generates control signals, a muscle model that controls joint angle, and a planar, kinematic arm. The control network is an array of APGs as shown in the box on the left of the figure. The APGs

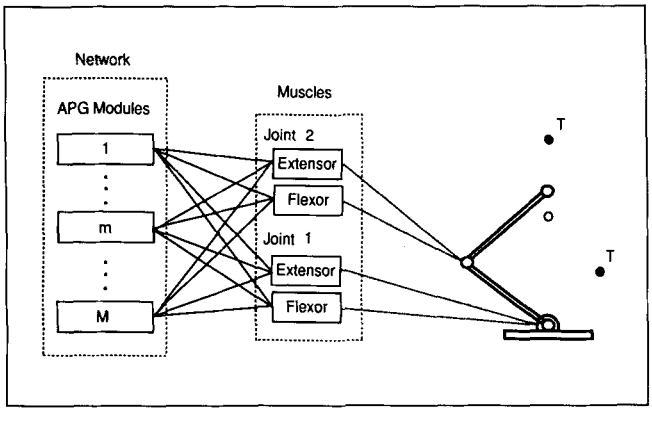

Figure 1. APG control of joint angles. A collection of APGs is connected to a simulated two degree-of-freedom, kinematic, planar arm with antagonistic muscles at each joint. The task is to move the arm in the plane from a central starting location to one of eight symmetrically placed targets. Activation of an APG causes a movement of the arm that is specific to that APG, and the magnitude of an APG's activity determines the velocity of that movement. The simultaneous activation of selected APGs determines the arm trajectory as a superposition of these movements.

generate signals that are fed to the limb musculature (center) to move the arm (right). Because here we are interested in the basic issue of how a collection of APGs might cooperatively control multiple degree-of-freedom movements, we use a very simplified model of the limb that ignores dynamics. The muscles convert APG activity to changes in muscle length, which determine the changes in the joint angles. Activation of an APG causes movement of the arm in a direction that is specific to that APG, and the magnitude of an APG's activity determines the velocity of that movement. The simultaneous activation of selected APGs determines the arm trajectory as the superposition of these movements. Learning processes adjust the subsets of APGs that are selected as well as characteristics of their activity in order to achieve desired movements.

Each APG (Fig. 2) consists of a positive feedback loop and a set of Purkinje cells (PCs). The positive feedback loop is a highly simplified model of a component of a complex cerebrocerebellar recurrent network. In this simplified model each APG has its own feedback loop, and the loops associated with different APGs do not interact. When triggered by sufficiently strong activation, the neurons in these loops fire repetitively in a self-sustaining manner. An APG's motor command is generated through the action of its PCs, which inhibit and modulate the buildup of activity in the feedback loop. PCs receive information via parallel fibers that specifies and constrains the desired movements. We hypothesize that the response of PCs to particular parallel fiber inputs is adaptively adjusted through the influence of climbing fiber activity signaling the occurrence of peripheral events. Climbing fibers provide "teaching" signals providing information allowing incremental improvement of the motor command (Houk & Barto, 1992). More detailed descriptions of APGs and relevant anatomy and physiology can be found in Houk (1989) and Houk et al. (1990).

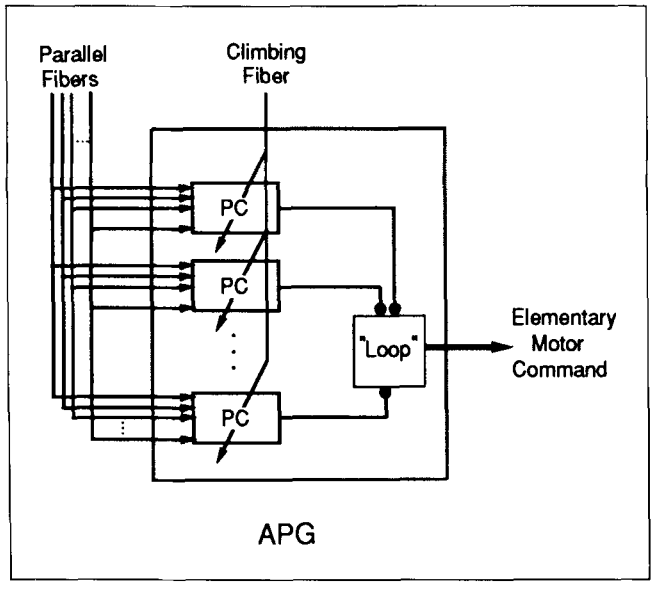

Figure 2. Functional diagram of a single APG. An APG receives information about targets and about limb position via parallel fibers. Climbing fibers convey training information that is hypothesized to alter the responsivity of PCs to parallel fibers. PCs inhibit the activity of neurons that participate in positive feedback loops to specify the output of the APG.

To investigate the behavior of the APG array model we simulated a movement task suggested by the task studied by Georgopoulos et al. (1982). This task has the advantage that neural and behavioral data are available to guide the development of the model. In the task's planar version, monkeys are trained to move from a central starting location to one of eight radially symmetric targets located on a board placed 15 degrees to the horizontal. Trials start with the monkey's hand in the central starting location when a LED at one of the target locations is illuminated. The monkey is then required to move to the target within a time criterion. As training proceeds over a period of months, the size of the targets is successively reduced. The ultimate criterion requires placement of the hand within 15 mm of the target.

Transformation of the Motor Command to Arm Movements

The cortico- and rubrospinal fibers that convey the motor command from the brain to the spinal cord cause arm movements primarily by their action on spinal interneurons. These interneurons influence several pools of motoneurons that innervate different muscles (Shinoda, Yokota, & Futami, 1981). As Georgopoulos (1988) suggested, each of the descending fibers might be thought of as activating a particular weighted combination of muscles causing a movement of the arm in a particular direction. The combined activity of multiple descending fibers produces the observed movement as the superimposed effects of the corresponding weighted muscle groups. Although this view of a movement as a combination of weighted muscle groups does not reflect the full complexity of the spinal system's underlying control of arm movement, it captures a basic feature of the descending connections to the motoneurons. In this respect, the APG array model is similar to synergy-based robot control schemes such as that proposed by Lane, Handelman, and Gelfand (1988).

The muscle–arm model used in the present paper is based on the preceding ideas but is very simplified because we wanted to investigate issues arising with the use of multiple APGs in as simple an example as possible. The model of the arm has two degrees of freedom and operates in a plane. We ignored the inertia and viscosity of the arm as well as changes in moment arms of the muscles with limb posture. APGs act via weighted, nonadaptive connections to incrementally change the rest length of linear springs representing the muscles (Fig. 1). According to this view, which is conceptually equiv-

alent to the approach of Massone and Bizzi (1989), a movement trajectory is a sequence of static postures that are determined by a sequence of spring rest lengths. APG–muscle connections are set so that if an APG were to act alone, it would move the arm in a direction that is different for each APG. Activation of each APG thus generates a specific movement of the arm, which here is a curved trajectory in the arm's workspace; the activation of multiple APGs determines the arm trajectory as the result of the simultaneous execution of movements generated by the individual APGs. The magnitude of an APG's activity determines the degree that its movement contributes to the trajectory.

The manner in which APG activity generates changes in arm position can be specified using the approach of Mussa-Ivaldi (1988). Suppose $u(t)$ is the activity of the APG array at time t. If there are M APGs, then $u(t)$ is a (column) vector of length M whose components are the activities of the M APGs. Let $\theta(t)$ denote the vector of joint angles at time t, and let $\Delta\theta(t)$ denote the change in joint angles from time t to $t + 1$.[1] Making the simplifying assumption that APG activity has an instantaneous and linear effect on changes in joint angles, we can write

$$\Delta\,\theta(t) = kAu(t) \qquad (1)$$

where A is the 2 row × M column matrix of weights that summarizes the influence of the M APGs on the two joints and k is a positive scaling factor. A weight can be either positive or negative depending on whether the APG flexes or extends the joint in question. (As explained in Appendix A, we defined joint angles using a convention from robotics in which a joint angle is zero when the adjacent arm links are aligned and increases as the joint moves counterclockwise; as a consequence the elbow joint angle increases with elbow flexion and decreases with extension.) The mth column of A, which we denote A_m, is the vector of weights giving how APG m influences the arm's joints. We loosely refer to these weights as connection weights to the joints, although they represent the descending network by which the APGs influence motor units. If the ith component of A_m is positive, we think of it as a connection weight to a flexor motor unit whose contraction increases the angle of joint i. On the other hand, if the ith component of A_m is negative, we think of the magnitude of this component as a positive connection weight to an extensor motor unit whose contraction decreases the angle of joint i. This means that the activity of an APG does not cause cocontraction of the flexors and extensors at any joint, a simplifying assumption in the current model.

That A does not vary with θ means that we have ignored the changes in moment arms of the muscles with limb posture. This simplification implies that each APG moves the arm in a particular direction in joint-angle space, called the *direction of action* of the APG. We prespecified the vectors A_m so that each had unit length and so that the directions of action of the APGs were equally spaced around the 360° of possible directions. Whereas the direction of action of an APG in joint-angle space is independent of the arm's current joint angles, the direction of action of an APG in the arm's workspace depends on these joint angles. Let $J(\theta)$ be the 2 × 2 Jacobian matrix of the transformation from joint-angle to Cartesian space when the arm is positioned with joint angles θ. Then the direction of action of APG m in Cartesian space is the direction of the vector $J(\theta)A_m$.

Adjustable Pattern Generators

Our model of an APG is based on the finding that cerebellar cortex is anatomically and functionally organized into parasaggital zones (Voogd & Bigaré, 1980; Oscarsson, 1980). Each APG is composed of a longitudinally oriented set of PCs that we assume receive the same parallel and climbing fiber input, as well as the same basket cell input. PCs inhibit cerebellar deep nuclear cells that form a link in a regeneratively active cerebrocerebellar loop (Allen & Tsukahara, 1974). This loop is treated as a positive feedback loop that remains active when triggered to provide a tonic level of activity that can be modulated by PC inhibition to form the motor command. The activity of loop cells is conveyed to spinal motor areas by corticospinal fibers. In this model we adopt a much simplified view of the loop in which we assume that the neurons within each nucleus or cortical area forming a link in the loop do not directly interact with each other. Consequently, the overall loop activity consists of the parallel activity of many separate loops, each of which is modulated by the PCs of a different APG. Further, we model each of these separate loops in highly aggregated form by representing only the overall activity level of each loop; the neurons making up the loop are not explicitly present in the model. Although a more refined model including loop cross-coupling and more complex loop dynamics will be essential for further development of the model, we did not want the present model to be so complex. Therefore, in the present model, by an APG we mean a set of PCs together with the positive feedback loop (an independent component of the overall loop) that the PCs modulate. The PCs of an APG share the same input signals.

The effect of synaptic inputs on the membrane potential of PCs cannot be modelled as a simple weighted sum. Small changes in input can lead to large changes in membrane potential because of high-threshold, slowly-inactivating calcium conductances (Llinas & Sugimori, 1980). In the intact cerebellum, PCs might have several possible stable levels that are the result of calcium conductances in different parts of the dendritic tree (Gutman, 1991). As an initial step in simulating these complexities, we model PCs as being in either "on-" or "off-states." We also assume that PCs show hysteresis, i.e., that PCs have different on- and off-thresholds. Figure 3A shows the state transition characteristics of a PC in the

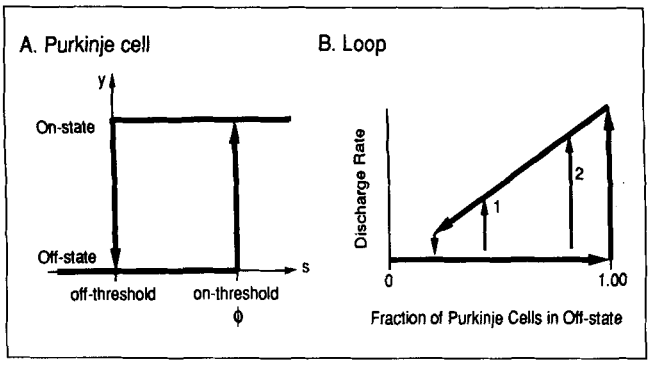

Figure 3. Operating characteristics assumed for PCs and for the cerebrocerebellar loops. (**A**) PC state, y, as a function of its total input, s. PCs are assumed to be bistable so that they are relatively insensitive to changes in their input except near the on-threshold, ϕ, where transitions from the off- to the on-state occur, and the off-threshold, where transitions from the on- to the off-state occur. (**B**) The cerebrocerebellar loops are also assumed to have off- and on-states, but with more complex features. When a loop is in its off-state, it has an output of zero independently of its PC input. Transitions to the on-state are caused by trigger signals, and after a transition, output (discharge rate) is proportional to the fraction of PCs in the APG that are in the off-state.

model. If the PC is in the off-state, the net input, s, must exceed the on-threshold, ϕ, in order to drive it to the on-state. If the PC is in the on-state, the net input must drop below the off-threshold to force the cell to the off-state. We assume that noise influences the transition to the off-state so that the lower the total input, the higher the probability of a transition to the off-state. A PC in the on-state is assumed to fire at a markedly higher rate than when it is in the off-state. For simplicity, here we assume that when a PC is in the off-state, it never fires, and when it is in the on-state, it fires at each time step.

The cerebrocerebellar loops are also assumed to have off- and on-states, but with more complex features (Fig. 3B). When a loop associated with an APG is in its off-state, it has an output of zero independently of its PC input. Transitions to the on-state are caused by trigger signals, and after a transition, output (discharge rate) depends on the fraction of PCs in the APG that are in the off-state.

The generation of motor commands occurs in three phases. In the first phase, we assume that all positive feedback loops are in the off-state, and teleceptive and proprioceptive parallel fibers and basket cells determine the states of the PCs. We call this first phase *selection*. In the second phase, loop activity is triggered by cortical activity. Once triggered, loop activity is self-sustaining because the loop cells have reciprocal positive connections. The triggering of loop activity causes the motor command to be "read out." The states of the PCs in the selection phase determine the speed and direction of the arm movement. As the movement is being performed, proprioceptive feedback from the arm gradually depolarizes the PCs and forces them to the on-state. When a large proportion of the PCs are in the on-state, PC inhibition reaches a critical value and terminates loop activity. This is the *execution* phase. In the third phase, the *correction* phase, corrective movements trigger climbing fiber activity that alters parallel fiber–PC connection weights.

APG Array Input–Output Behavior

Figure 4 shows the structure of a single APG in more detail. The output of an APG depends on its internal state as well as its input, where the state of an APG is determined by the states of all of its PCs as well as the state of its feedback loop. Suppose each APG consists of N PCs. For each APG m, $m = 1, \ldots, M$, each PC j, $j = 1, \ldots, N$, in APG m, and each time t, $s_j^m(t)$ and $y_j^m(t)$, respectively, denote the PC's total input and state at time t, where the latter is a binary variable indicating whether the PC is in the on-state [$y_j^m(t) = 1$] or the off-state [$y_j^m(t) = 0$]. The PCs in the same APG receive the same input signals via multiple excitatory parallel fibers, a single inhibitory basket cell input, and a single climbing fiber.

As the simplest starting point for the current studies, there are only nine parallel fiber inputs to each PC. We view these parallel fiber inputs as highly abstract representations of the much richer parallel fiber input that an APG would receive in a more detailed model. Eight of the parallel fibers are *target fibers* that represent the target for each trial, and these fibers synapse upon all

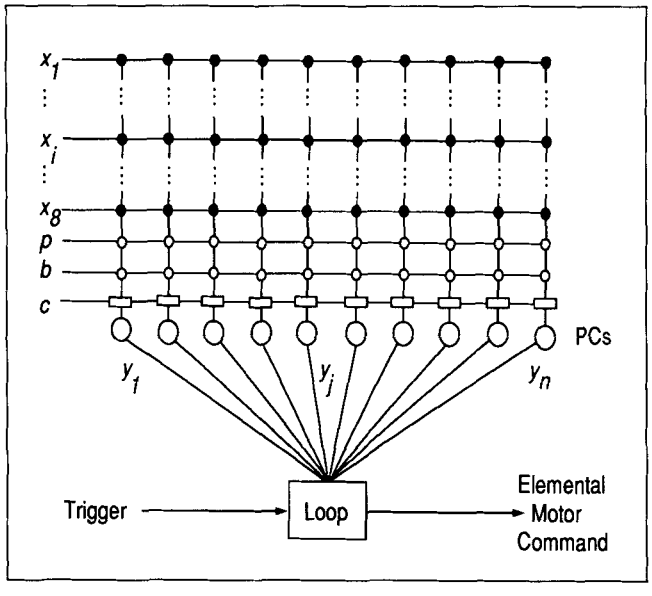

Figure 4. Structure of APG m. Each APG m consists of N PCs with states $y_1^m(t), \ldots, y_N^m(t)$ at time t and a positive feedback loop. Target related signals are $x_1(t), x_2(t), \ldots, x_8(t)$, proprioceptive feedback is $p_m(t)$, basket cell input is $b_m(t)$, and climbing fiber input is $c_m(t)$. Only the connection weights shown as filled circles are modified by climbing fiber input. The output of the APG is determined by the state of the feedback loop and the total amount of PC inhibition of the loop.

PCs of the model, irrespective of the APG in which they are located. We used the simplest possible represention of the targets. The presence of one of the eight targets is signaled by setting one of eight target fibers to one and the other seven to zero. That is, for $i = 1, \ldots, 8$, let $x_i(t)$ denote the activity of the ith target fiber at time t. If target k is present at time t, then $x_k(t) = 1$ and $x_i(t) = 0$ for $i = 1, \ldots, 8, i \neq k$. In this case, we call parallel fiber k the *active target fiber*. This simple coding means that there is neither interference nor generalization in the learning process by which the target signals come to be associated with movements. Substituting more sophisticated coding schemes, such as those used in the models of cerebellar cortex by Marr (1969) and Albus (1971), or by Zipser and Andersen (1988) in their study of parietal cortex, is relatively straightforward and is reserved for future elaborations of the model. The ninth parallel fiber to an APG transmits proprioceptive information about limb position, and this information is different for different APGs. Let $p_m(t)$ be the positive real number representing the activity of the parallel fiber conveying proprioceptive information to APG m at time t. It represents the aggregate influence of all proprioceptive parallel fibers to the APG. We describe how $p_m(t)$ is computed for each APG m in the next section.

The other inputs to each PC of APG m are basket cell and climbing fiber inputs, whose activities at time t we denote by $b_m(t)$ and $c_m(t)$. These inputs are binary, with 1 and 0 respectively indicating activity and inactivity. All the PCs of a single APG receive the same basket cell and climbing fiber input. In the current model, basket cell input inhibits PCs in the selection phase as described in the section on The Selection Phase. Climbing fiber input is the critical component of the correction phase described in the section on The Correction Phase.

At time t, the total input to PC_j of APG m is

$$s_j^m(t) = \sum_{i=1}^{8} w_{ij}^m(t)x_i(t) + w_p p_m(t) + w_b b_m(t) \qquad (2)$$

where $w_{ij}^m(t)$ is the connection weight at time t of the ith target fiber to PC j, w_p scales the aggregate proprioceptive feedback, and $w_b < 0$ is the weight of the inhibitory connection from the APG's basket cell. These latter two parameters are the same for all APGs and also remain constant over time. In the simulations presented in this article, their values were chosen so that $s_j^m(t) \geq -0.5$ for all t (see Appendix A for details). Because in the current model the weights associated with the target inputs change during learning, these weights require the time argument.

The state of PC j of APG m at time t depends on its current state as well as its total input $s_j^m(t)$. Figure 3A shows the hysteresis curve that characterizes this behavior. The state of a PC remains the same unless the following conditions for a state transition are met. A transition from the off-state to the on-state occurs whenever the total input equals or exceeds the on-threshold ϕ, i.e., whenever $s_j^m(t) \geq \phi$. Transitions from the on-state to the off-state, on the other hand, depend probabilistically on the current state and total input. In the current model, the probability that PC j of APG m turns off when it is on is

$$Pr\{y_j^m(t+1) = 0 | y_j^m(t) = 1\} = \begin{cases} -2s_j^m(t) & \text{if } s_j^m(t) < 0 \\ 0.05 & \text{otherwise} \end{cases} \qquad (3)$$

This implies that whenever a PC in the on-state is maximally hyperpolarized [$s_j^m(t) = -0.5$], it turns off with probability one. The probability of turning off decreases as the degree of hyperpolarization decreases. The probability of a PC in the on-state turning off when depolarized is 0.05 to allow a PC to occasionally turn off under these conditions.

The output of APG m at time t, which we denote $u_m(t)$, depends on the state of the positive feedback loop and the states of the N PCs in the APG. When the loop is in the off-state, the APG's output is zero independently of the activity of the PCs. When the loop is in an on-state, i.e., after it has been triggered, the APG's output is equal to the number of PCs that are in the off-state. This represents the influence of the total PC inhibition on the positive feedback loop: more PCs in the off-state implies less inhibition, which permits greater loop activity. Figure 3B shows two examples. In the case marked 1, relatively few PCs are off and hence, when triggered, the APG permits loop activity to reach only a modest level. In case 2, on the other hand, a large number of PCs are off so that the APG permits the loop to reach a high level of activity when triggered. The model as presented here represents only this basic relationship between loop activity and APG activity and does not include any details of the loop dynamics. Future work will involve disaggregating this very crude loop model following the preliminary work of Eisenman et al. (1991). We use $u(t)$ to denote the output at time t of the entire array of APGs; it is a vector of length M whose mth component is $u_m(t)$.

We now describe how the model generates proprioceptive signals; in the section on Climbing Fiber Responses we describe how it determines climbing fiber activity.

Proprioceptive Input

The cerebellum receives a substantial proprioceptive input that contributes to both the parallel fiber and the climbing fiber innervation of PCs (Bloedel & Courville, 1981). This section deals with the parallel fiber input, whereas proprioceptive signals in climbing fibers are discussed in the section on Climbing Fiber Responses. The proprioceptive parallel fiber input is both highly divergent and highly convergent. This ensures that PCs in all of the APG modules can potentially receive proprioceptive information from the entire limb. In our

present model, we assume that an APG's PCs are specifically wired up to respond to length changes of the muscles attached to the joints the APG influences. Eventually we hope to show that this is a natural consequence of developmental plasticity in the mossy fiber-parallel fiber pathway as shaped by training inputs from climbing fibers, but in the present model we assume that this process has already taken place. As a further simplification, we use a single parallel fiber to represent the combined, or aggregate, influence of all of the proprioceptive parallel fibers that would be present in a more detailed model. We defined this aggregate proprioceptive input so that it tends to depolarize the PCs in an APG as the arm moves in the direction of action of that APG. For example, an APG that has a positive connection to the two joints receives increasing aggregate proprioceptive input when θ_1 and θ_2 increase. An APG that has a negative connection to a joint receives increasing aggregate proprioceptive input when the joint angle decreases. To accomplish this, we defined each APG's aggregate proprioceptive input to be a particular weighted sum of signals giving information about the positions of the multiple joints influenced by the APG, where the weights are related to the weights by which the APG influences the joints. Athough these weights are probably the result of a learning process, we selected them by hand and held them fixed throughout the simulations.

To see how an aggregate proprioceptive input meeting these requirements could arise from muscle afferents and to explain how we selected these weights in the model, recall that the vector A_m represents how the activity of APG m influences the movements of the joints. Each component of A_m can be positive or negative, respectively indicating influence on flexors or extensors of a joint. If component i of A_m is positive, then the activity of APG m influences a flexor that increases the angle of joint i. A signal generated by an afferent sensing the length of an antagonist extensor would therefore increase in magnitude if APG m were to act alone. On the other hand, if component i of A_m is negative, then the activity of APG m influences an extensor that decreases the angle of joint i. The magnitude of a signal generated by an afferent sensing the length of an antagonist flexor would therefore increase if APG m were to act alone. This is consistent with the results of Capaday and Cooke (1981), who showed the importance of sensation from the antagonists in sensing joint angle. If we form a weighted sum of these afferent signals, where the weights are the magnitudes of the corresponding components of A_m, the result turns out to be an aggregate signal that would increase in magnitude if APG m were to operate alone to generate a movement.

Specifically, in our model we define the aggregate proprioceptive input at time t to the PCs of APG m, which we denote $p_m(t)$, as follows. Because extensor activity reduces a joint's angle from its maximum flexor of π radians (see Appendix A), afferent signals that increase with increasing extensor activity covary with the angles $\pi - \theta_i(t)$, $i = 1,2$. Consequently, define $\bar{\theta}(t)$ to be the vector whose ith component is $\pi - \theta_i(t)$ if the ith component of A_m is negative (i.e., APG m activates an extensor of joint i) and is $\theta_i(t)$ otherwise. Then

$$p_m(t) = |A_m|^T \bar{\theta}(t) \tag{4}$$

where $|A_m|$ is the vector of the absolute values of A_m's components, and the superscript T indicates the transpose of this vector. This means that $p_m(t)$ is a weighted sum, where the weights are all positive, of the joint angles comprising the vector $\bar{\theta}(t)$.

We can rewrite Eq. (4) as follows:

$$p_m(t) = A_m^T[\theta(t) - \Pi_m] \tag{5}$$

$$= A_m^T\theta(t) - A_m^T\Pi_m \tag{6}$$

where Π_m is the vector whose ith component is π if the ith component of A_m is negative and is zero otherwise. When written in this form, one can see that $p_m(t)$ would increase if APG m were to operate alone in generating a movement because A_m gives the direction of action of APG m. One can also see that the total proprioceptive input to the APG array is described by the matrix A^T, whose mth row is A_m minus a vector that is constant over time. We can think of this as the connection matrix of a network transmitting proprioceptive information to the APG array. The matrix gives the synaptic weights by which parallel fibers responsive to proprioceptive signals influence PCs. The network with connection matrix A^T has the same structure as the network by which the APG array influences joint movements (with connection matrix A), but transmits information from the arm to the APG array instead of from the APG array to the arm.

This definition of the aggregate proprioceptive input to an APG does not take into account the time delay that would be present in transmitting information about the arm's movement to PCs: $p_m(t)$ depends on $\theta(t)$ instead of $\theta(t - \tau)$, for some time delay τ. Although we did not employ a time delay in the simulations to be described, the behavior of the APG array model is not critically sensitive to this delay. This is a basic feature of the model that is explained below.

The Selection Phase

The selection phase begins with all positive feedback loops in the off-state, all PCs in the on-state, and parallel fiber input specifying information about the current target and providing proprioceptive signals generated by the current position of the arm. Selection then occurs when the basket cells fire, which provide the only inhibitory input to the PCs. Here we assume that all the basket cells fire at the same time, an assumption we will modify in future versions of the model. As a result, *some* of the PCs in each APG are turned off. Which PCs turn off depends on their total input [Eq. (2)] according to

the probabilistic state-transition rule given by Eq. (3). We say that a PC that has been set to the off-state by basket cell inhibition has been *selected*. After some PCs have been selected, basket cell activity ceases until the next selection phase. After selection of a subset of PCs, the positive feedback loops for all the APGs are triggered, which we did by simply changing the state of each positive feedback loop from the off-state to a state given by the total number of PCs selected in the corresponding APG. Triggering signals the start of the execution phase during which the motor command is generated. As described next, the subset of selected PCs determines the end-point and trajectory of the movement generated during the execution phase.

The Execution Phase

During the execution phase, all of the inputs to the PCs remain constant except for the input conveying the aggregate proprioceptive information. The aggregate proprioceptive input is determined by the arm position as previously explained. The aggregate proprioceptive input increases for most of the APGs involved in generating a movement. This is true because (1) the aggregate proprioceptive input to an APG increases when the arm moves under control of that APG acting alone [Eq. (5)], and (2) most of the APGs involved in a particular movement generate movement in similar directions. Movement stops when the aggregate proprioceptive input is sufficient to force a large proportion of the selected PCs to the on-state. As the selected PCs turn on, the level of inhibition to the positive feedback loop increases until the loop reverts to the off-state. This process is insensitive to delayed proprioceptive feedback because this feedback does not act continuously to control the motor command, as it would in a conventional feedback control system. Instead, proprioceptive feedback merely signals the conditions for terminating loop activity. This process requires only that the first time during a movement that the activation of a selected PC exceeds its on-threshold is the time when the motor command should terminate. Due to the learning process described below, this requires that the aggregate proprioceptive input to a selected PC is increasing when the command should terminate, but it does not require this input to have a specific magnitude or to follow a specific time course.

We can analyze the conditions required for a movement to stop when the target position is reached. During a movement when the kth target is presented, the total input to PC j of APG m is

$$s_j^m(t) = w_{kj}^m(t) + w_p p_m(t) \tag{7}$$

where $w_{kj}^m(t)$ is the connection weight from the kth target fiber to PC j. This is true because in Eq. (2), $x_k(t) = 1$, $x_i(t) = 0$ for $i \neq k$, and the basket cell input is zero during movements. If PC j has been selected, it makes the transition to the on-state when the total input, $s_j^m(t)$, exceeds the on-threshold, ϕ. Because the arm is supposed to stop when it reaches the target, the proprioceptive feedback when the arm is in a target posture should force each selected PC to the on-state. If the target is reached at time t_1, this means that $w_{kj}^m(t_1)$ must equal $\phi - w_p p_m(t_1)$. Let us call this weight w_m^*.[2]

But we have assumed that PC j of APG m has been selected. If the connection weight w_{kj}^m has the value w_m^* required to appropriately terminate a movement, how likely is it that PC j will be selected during the selection phase? Let t_0 denote the time at which selection occurs, i.e., at time t_0, $x_k(t_0) = 1$, $x_i(t_0) = 0$ for $i \neq k$, and the basket cell input, $b_m(t_0)$, is equal to one. Then from Eq. (2) we see that the total input to PC j of APG m at selection is

$$\begin{aligned} s_j^m(t_0) &= w_m^* + w_p p_m(t_0) + w_b \\ &= [\phi - w_p p_m(t_1)] + w_p p_m(t_0) + w_b \\ &= w_p[p_m(t_0) - p_m(t_1)] + \phi + w_b \end{aligned}$$

Using Eq. (5) to express proprioceptive feedback in terms of joint angles, we have that

$$\begin{aligned} s_j^m(t_0) &= w_p[A_m^{\mathrm{T}}(\theta(t_0) - \Pi_m) - A_m^{\mathrm{T}}(\theta(t_1) - \Pi_m)] \\ &\quad + \phi + w_b \\ &= w_p A_m^{\mathrm{T}}(\theta(t_0) - \theta(t_1)) + \phi + w_b. \end{aligned} \tag{8}$$

Because the selection probability of PC j is proportional to $-s_j^m(t_0)$ if $s_j^m(t_0) < 0$ [Eq. (3)], setting $w_b = -\phi$ implies that at selection when PC j is inhibited, its probability of being selected is proportional to the inner product of A_m and $\theta(t_0) - \theta(t_1)$. This latter vector points in joint-angle space from the point representing the target location to the point representing the initial joint angles. This has the following implications:

1. When the directions of action of the APGs are uniformly distributed in joint-angle space, as they are in the model, the expected trajectory of a movement is a straight line in joint-angle space from the starting position to the target. This is true because one can show that the expected movement in joint-angle space is proportional to $AA^{\mathrm{T}}[\theta(t_0) - \theta(t_1)]$, and AA^{T} turns out to be proportional to the identity matrix due to the fact that A is defined so that the directions of action of the APGs are uniformly distributed in joint-angle space. This is a special case of the result proved by Mussa-Ivaldi (1988), which holds for a wide range of such distributions.

2. The population of vectors of APG activity is cosine-shaped in joint-angle space, i.e., the length of the vector representing the activity of APG m is proportional to the cosine of the angle between the direction of action of APG m and the direction of the target in joint-angle space. This is true because the inner product by which selection occurs is proportional to the cosine of the angle between the APG's direction of action and the direction to the target in joint-angle space.

3. Presentation of targets that are farther from the starting posture in angle space lead to more rapid movements than closer targets. This is because A_m remains constant while the magnitude of $\theta(t_0) - \theta(t_1)$ increases with increasing target distance, resulting in a larger inner product in Eq. (8). This causes more PCs to be selected and hence the APGs command higher velocity movements.

Correction Phase

A third phase in the operation of this model is the correction phase. "Correction" refers to both corrective movements and to a resultant correction of connection weights. The corrective movements are assumed to be crude movements in the general direction of the target produced by extracerebellar circuits whenever the initial movement generated by the APG array fails to aquire the target. We also assume that climbing fibers sensitive to proprioceptive input are selectively responsive to these corrective arm movements, thus providing training information about the direction of an error. Connection weights of parallel fibers are incrementally adjusted as a result of this climbing fiber activity. These various assumptions are based on the physiological properties of climbing fibers and of their effects on PCs.

Generation of Corrective Movements

Reaching movements to switched targets have corrective components that operate at very short latencies, suggesting short pathways through the nervous system (Alstermark, Eide, Gorska, Lundberg, & Pettersson, 1984; Gielen & van Gisbergen, 1990; Pelisson, Prablanc, Goodale, & Jeannerod, 1986). Fast switching is abolished when the tectospinal pathway is interrupted, suggesting that this corrective system bypasses the corticospinal and rubrospinal pathways that initiate the first phase of the movement (Alstermark, Gorska, Lundberg, Pettersson, & Walkowska, 1987). Behavioral and computational studies by Flash and Henis (1991) also suggest that the corrective movement is generated by a system distinct from the corticospinal and rubrospinal systems, as the corrective component is simply added to the initial movement command. In the simulations described here, corrective movements were generated using a probabilistic scheme that resulted in a movement whose expected direction was in the target direction. The direction of each corrective movement was generated randomly according to a Gaussian distribution around the target direction. We set the variance of the Gaussian so that with probability 0.5 the direction of the corrective movement was within 45° of the correct direction. In the current simulations corrective movements were only generated when the movement was farther than 1.5 cm from the target, the radius of the target used in behavioral studies (e.g., Georgopoulos et al., 1982).

Climbing Fiber Responses to Corrective Movements

Our assumption that climbing fibers are selectively responsive to corrective movements is based on several lines of physiological evidence. Many climbing fibers are responsive to proprioceptive input (Gellman, Gibson, & Houk, 1985), and this responsiveness is selectively inhibited during self-initiated movements (Andersson & Armstrong 1987; Gellman et al., 1985). In contrast, climbing fibers responsive to tactile input appear to be inhibited at the end of a self-initiated movement (Gellman et al., 1985; Weiss, Houk, & Gibson, 1990). In the studies of Gilbert and Thach (1977), which supported a role for climbing fibers in motor learning, discharge of these fibers was especially evident when the primate subjects were making corrective limb movements. On the basis of these findings, we speculate that climbing fibers sensitive to proprioceptive input are selectively inhibited during the initial movement generated by the APG array, but regain their responsiveness during crude corrective movements generated by extracerebellar circuitry. Proprioceptive climbing fibers have preferred directions for stimulation but respond when the limb is manipulated in a variety of directions (Gellman et al., 1985). In the present simulations we use a cardioid-shaped receptive field for the climbing fibers following the use of cardioids to fit to retinal slip climbing fibers (Rosenberg & Ariel, 1990) and proprioceptively driven muscle responses (Flanders & Soechting, 1990). We further assume that the preferred directions of the climbing fibers are aligned with the directions of action of the APG modules they innervate, thus taking advantage of the precise anatomy of the climbing fiber input to the cerebellum (Oscarsson, 1980; Voogd & Bigaré, 1980).

Because the proprioceptive receptors underlying climbing fiber responsiveness lie in muscle or in joints, we specified their receptive fields in joint-angle space. Letting θ denote the vector of current joint angles, assume a corrective movement is generated according to the probabilistic scheme described above. Then the initial direction of the corrective movement in joint-angle space is

$$\Delta\theta \approx J^{-1}(\theta)\Delta X$$

where $\Delta\theta$ is the change in joint angles, $J^{-1}(\theta)$ is the inverse of the Jacobian of the arm's kinematic transformation for the current joint angles, and ΔX is a small magnitude vector in the direction of the corrective movement in Cartesian space.

The probability of firing of the climbing fiber of APG m depends on the relationship between the direction of the corrective movement in joint-angle space and the direction of the movement caused by activity of APG m. Recall that the influence of APG m on joint movements is determined by the mth column, A_m, of the connection matrix A (see section on Transformation of the Motor

Command). Letting ω denote the angle between $\Delta\theta$ and A_m at a time t when a corrective movement occurs, we let the firing probability of the climbing fiber of APG m be

$$Pr\{c_m(t) = 1\} = \frac{1 + \cos(\omega)}{2} \quad (9)$$

At all times t when there is no corrective movement, $c_m(t) = 0$. This implies that when the initial direction of a corrective movement is exactly in the direction that the APG moves the arm ($\omega = 0$), the climbing fiber always fires, and when the initial direction of a corrective movement is directly opposite from the direction the APG moves the arm, it never fires ($\omega = \pi$). Climbing fiber firing probability varies smoothly between these extremes for other corrective movements. Figure 5 illustrates the procedure by which the probability of a climbing fiber firing is determined; Eq. (9) is plotted as a cardioid in polar coordinates.

The rule specifying climbing fiber responsiveness given by Eq. (9) can be implemented by a network similar to that previously described for generating the aggregate proprioceptive input to APGs used in the model. Although we did not explicitly implement this network in the simulations, its matrix of connection weights is A^T, suitably normalized to produce climbing fiber firing probabilities. This network has the same structure as the network described by the connection matrix A by which the APG array influences joint movement [Eq. (1)], but it transmits information in the opposite direction: from proprioceptors to the APG array. Consequently, the model uses a learning mechanism somewhat related to the well-known error backpropagation learning algorithm that is widely used to train layered artificial neural networks (Rumelhart et al., 1986). We discuss this in more detail in the Discussion section on Climbing Fiber Signals.

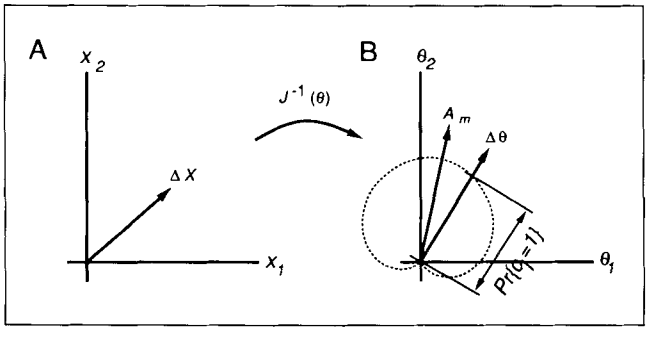

Figure 5. Determination of climbing fiber firing probability. **(A)** A unit vector, ΔX, that points in the direction of the corrective movement in Cartesian space. $J^{-1}(\theta)$ transforms ΔX from Cartesian to joint-angle space. **(B)** The resulting vector, $\Delta\theta$, as well as the vector, A_m, describing the influence of APG m on the joints. Plotting the climbing fiber's firing probability as a function of the angle, ω, between $\Delta\theta$ and A_m produces the cardioid shown as a dotted line.

Adjustment of PC Weights

The rule used to adjust the connection weights of the target fibers to PCs is based on the cellular properties that regulate plasticity of parallel fiber synapses on PCs (Crepel & Krupa, 1988; Ekerot, 1984; Ito, 1989; Linden, Dickinson, Smeyne, & Conner, 1991). These data indicate that plasticity in PCs is different than plasticity at most other sites in the nervous system. Whereas long-term potentiation (LTP) in hippocampal and other neurons is considered to depend on the concurrence of presynaptic input and postsynaptic activity (Collingridge & Singer, 1990), long-term depression (LTD) in PCs appears to depend on these two plus a third factor (Ekerot, 1984; Linden et al., 1991), all of which are incorporated in the learning rule used here (see Houk et al., 1990 and Houk & Barto, 1992 for more complete discussions). The three factors influencing long-term depression in the PCs of an APG m are (1) the presynaptic input, x_i, (2) the postsynaptic response, y_j^m of PC j, and (3) the climbing fiber input, c_m, to that Purkinje cell. The weights of the target fiber inputs to APG m are adjusted according to the following equation:

$$\Delta w_{ij}^m(t) = \alpha[1 - c_m(t)][1 - y_j^m(t)]x_i(t) - \beta c_m(t) y_j^m(t) x_i(t) \quad (10)$$

where $\Delta w_{ij}^m(t)$ is the change at time t of the strength of the synapse by which target parallel fiber i influences PC j of APG m, and α and β are constants determining the rates of facilitation and depression, respectively. The other connection weights from the proprioceptive parallel fibers to the PCs are held constant.

This learning rule is closely related to the rule described by Albus (1971) and the perceptron learning rule (Rosenblatt, 1962) if one assumes that the climbing fiber provides a signal that is the *opposite* of the signal that the PC should produce (Bartha, Thompson, & Gluck, 1991). If a PC's climbing fiber fires when the PC is in the on-state, all the weights with active inputs are decreased, making it less likely that the PC will fire to similar inputs. On the other hand, if the climbing fiber does not fire and a PC is off, then the weights are increased for active inputs, increasing the probability that the PC will fire.

SIMULATION EXPERIMENTS

We simulated the APG array model in an analog of the planar movement studied by Georgopoulos et al. (1982). This task requires a rhesus monkey to move from a starting point in the center of the workspace to one of eight radially symmetric targets. A trial starts with the monkey's hand over a central starting position when a LED at the location of one the targets is turned on. A correct response is defined as a movement made within a time criterion that ends within 1.5 cm of the target. Targets are presented serially. In our simulations we used the workspace dimensions of Georgopoulos et al. (1982),

and we set the link lengths of the simulated arm to those of a typical rhesus monkey (see Appendix A).

Our simulations employed 48 APGs, each consisting of 36 PCs. The on-threshold, ϕ, of all the PCs was set to 1. The basket cell weight, w_b, of each PC was set to -1, and the scale factor for the aggregate proprioceptive input, w_p, to each PC was set to 0.2. The learning rate parameters α and β were set to 0.0001 and 0.0011, respectively. All the target fiber connection weights were initially set to 1. Appendix A provides a detailed discussion of how the parameters of the current simulations were selected.

Each simulated trial started with the end-point of the arm in the center of the workspace. The selection phase was then simulated. With all the positive feedback loops set to the off-state and all of the PCs set to the on-state, one of the eight target parallel fiber inputs was set to one and the other seven were set to zero. The proprioceptive input to each APG was set according to the current position of the arm. The basket cells then fired to inhibit all the PCs, thus setting some of them to the off-state, i.e., selecting a subset of the PCs. The execution phase began with the triggering of the positive feedback loops, which we did simply by changing the state of each positive feedback loop from the off-state to a state determined by the total number of PCs selected in the corresponding APG. Arm movement then commenced, causing proprioceptive input to each selected APG to change over time. Depending on the adjustable connection weights, selected PCs reverted to the on-state when sufficiently excited by proprioceptive input. When 95% of all the PCs were in the on-state, PC inhibition of each loop was considered sufficient to stop arm movement, and a corrective movement was then generated. Climbing fibers then fired according to the scheme described above, weights were adjusted, and the trial was terminated. Targets were presented serially.

Figure 6 shows a typical learning curve for movements to a single target. It shows the distance from the movement endpoint to the target for 750 training trials. Initially, arm movements were very small and in random directions because of the large initial connection weights (1.0) of the target fibers. The large initial weights made it unlikely that many PCs would be selected because the total input to each was sufficiently positive to produce a low probability of making a transition to the off-state [Eq. (3)]. Repeated climbing fiber firing lowered these weights, eventually causing the arm to move in the direction of the target. This took about 300 trials. Once this point was reached the error distance decreased until the endpoint of the movement was within about one cm of the target. Performance then stabilized with arm movements consistently ending close to the target.

Figure 7A shows three representative trajectories to each of the eight targets after training. The starting point for each movement is the center of the workspace, and the target location is the center of the open square. The position of the arm at each time step is shown as a dot. Movement trajectories tended to be curved lines in Cartesian space and straight lines in joint-angle space from the starting position to the target. Curvature of movement trajectories varied between targets, a result that was expected because of the positional dependence of the transformation from joint-angle space to Cartesian space. Movements to some targets were learned with greater accuracy than to others. Further, because selection of PCs was stochastic, trajectories to a target varied from trial to trial. For example, the movements to the top-most target in Figure 7A showed significant variability.

In these simulations it was possible for a single APG acting alone to move the arm to each of the targets. However, after training, subsets of APGs were used for each target. For any given target about half of the APGs

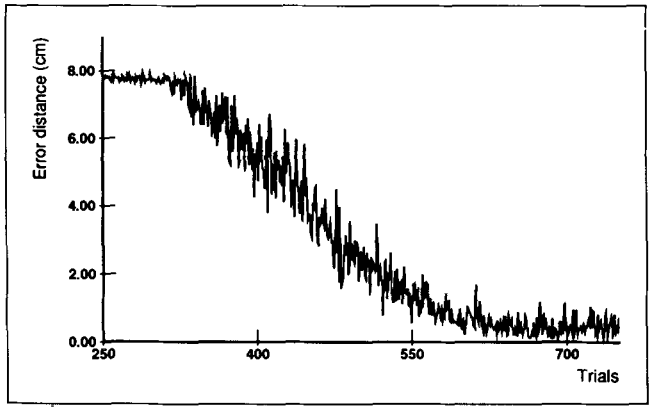

Figure 6. The distance in centimeters between the movement endpoint and the target plotted as a function of training trials for a representative training session. The distance from the starting point to the target was 8 cm.

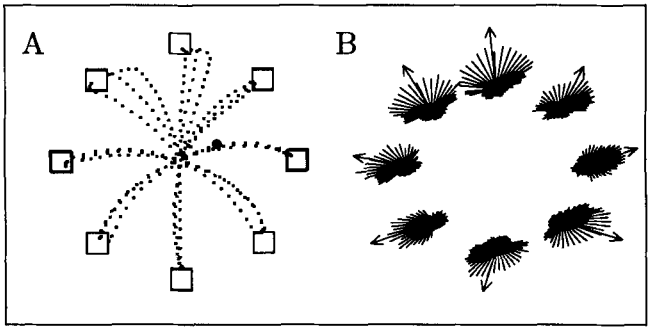

Figure 7. (**A**) Movement trajectories after training. The starting point for each movement is the center of the workspace, and the target location is the center of the open square. The position of the arm at each time step is shown as a dot. Three movements are shown to each target. (**B**) APG selection. APG selection for movements to a given target is illustrated by a vector plot at the position of the target. An individual APG is represented by a vector, the direction of which is equal to the direction of movement caused by that APG in Cartesian space. The vector length is proportional to the number of PCs that are selected during the selection phase. The arrow points in the direction of the vector sum.

were used. Figure 7B shows vector displays of APG selection following the method used by Georgopoulos (1988). Each cluster of lines refers to movements to the target in the corresponding position. The direction of a line in a cluster is the direction that one APG would cause the arm to move if it were to act alone. The length of a line is proportional to the average number of the APG's PCs that were selected for movements to that target. The line with the arrowhead shows the direction of the vector sum of the APGs' activities. Figure 7B shows that the motor command generated by the network was represented as a pattern of activity distributed over a set of APGs. Activity was distributed relatively broadly across the APGs, but the vector sum accurately pointed in the direction of the initial movement.

According to Eq. (8), the shape of the vector plots should be cardioids in joint-angle space. In Cartesian space, as in Figure 7B, the shapes of the vector clusters vary with the location of the target because of the nonlinear transformation from joint-angle to Cartesian space.

Because the cerebellum has been implicated in correcting movement length and in adapting to changes in the sensorimotor environment, we investigated the effect of shifts of target position after network training to simulate the effect of putting reversing prisms on the subject. In this series of simulations, a target to which the system had already learned to reach was shifted by reflecting its position about the ordinate. Figure 8 shows the results of this experiment. Figure 8A shows three superimposed movement trajectories to the target 45° to the right of the top-most target after initial training. The arm moves to the target in an approximately straight line. The target was then shifted to 45° to the left of the top-most target without changing the pattern of parallel fiber activity representing the target. Figure 8A also shows two representative movements that occurred while the model was learning to reach to the shifted target. One movement occurred on the 175th trial and the other on the 300th trial after the shift. Figure 8B shows three movement trajectories 500 trials after the shift.

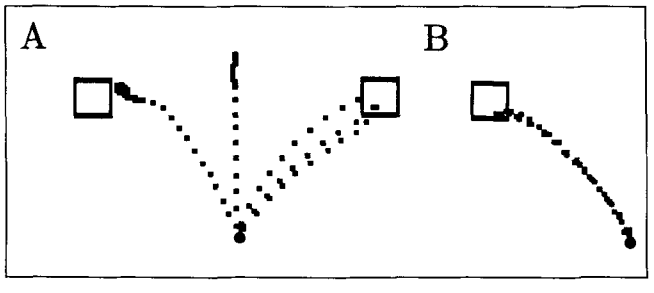

Figure 8. Simulated trajectories for shifting an already learned target. (**A**) Three representative trajectories after the initial target had been learned (target on right). Single representative trajectories are also shown for simulated reaches 175 and 300 trials after the target was shifted to the left. The starting point is the filled circle. (**B**) Three representative trajectories for movements 500 trials after the shift.

ANALYSIS OF NETWORK OPERATION

Trajectory and End-Point Generation

How the APG array model determines a movement trajectory can be understood through the examples shown in Figure 9. For this analysis, we assume that the values of the weights are constant within a trial and that all PCs in the same APG have the same weights. Because we have coded target information in such a simple way, the only critical connection weight in a trial is that of the active target fiber. Following a standard method of analysis for linear threshold units (e.g., Nilsson, 1965), one can understand the influence of this weight in terms of a decision surface in the space of possible input signals to an APG. During the execution phase, this space can be identified with the space of joint angles because the only time-varying input is the proprioceptive input generated by the moving arm. If we consider an arbitrary PC j in APG m, the value of the target fiber weight, w_{kj}^m, determines a decision surface dividing the space of possible joint angles into two disjoint regions: In one region, s_j^m is above the on-threshold; in the other region, s_j^m is below the on-threshold. It can be shown that this decision surface is a line perpendicular to the direction of action of the APG (see Appendix B). We call this dividing line the *decision line* for the APG. Changing the value of w_{kj}^m moves the decision line along the direction of action of the APG.

Because we want to ensure that the selected APGs turn off when the target is reached, the decision lines of the

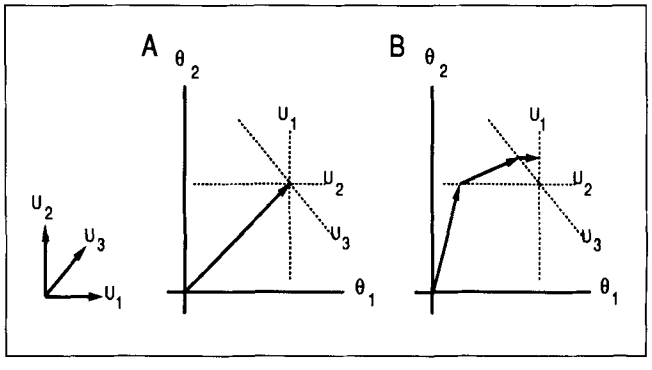

Figure 9. Arm trajectories in joint-angle space for movements generated by three APGs. The three vectors at the lower left show the directions of movement for the three APGs (numbered 1, 2, and 3) in joint-angle space. The dotted lines in **A** and **B** show the decision lines for the three APGs. The decision lines are perpendicular to the direction of movement in joint-angle space with the position determined by the value of the target fiber weight. The decision lines shown are at the correct location for a target reached when the joint angles are at the intersection of the lines. In **A**, a trajectory is shown from the starting to target positions where the PCs of the APGs were selected in equal proportion. The trajectory is a straight line that ends at the target. **B** shows a trajectory where twice as many PCs were selected for APG 2 as for APGs 1 and 3. The PCs of the APG turn on when the decision line is crossed, and a piecewise linear trajectory is seen that does not end at the target.

selected APGs should intersect at the point in joint-angle space at which the arm reaches the target position.[3] However, because selection is stochastic and because the APGs interact in producing the movement, the target fiber weights of the selected APGs do not determine the trajectory and the endpoint of the movement in a simple way. This can be seen by examining Figure 9A and B, which illustrate two different movements from a single starting location to a target with the target fiber weights held constant.

Figure 9 shows movements that are controlled by three APGs, numbered 1, 2, and 3, whose directions of action are shown at the lower left. In Figure 9A equal numbers of PCs were selected in the three APGs. In this example, the arm moves in a straight line in joint-angle space to the target and stops. This straight-line trajectory is similar to the previously discussed *expected* trajectory that results when our probabilistic selection rule is used. In contrast, Figure 9B shows a movement in which twice as many PCs were selected for APG 2 as for APGs 1 and 3. Because selection is probabilistic, selection of PCs in these proportions is possible and was seen in simulations. In this case, the trajectory is piecewise linear, with changes in direction when the arm crosses the decision lines and PCs turn on. The endpoint of the movement is not close to the target due to the asynchronous termination of the activities of the contributing APGs, a general problem discussed by Bullock and Grossberg (1988).

These examples show that both the trajectory and accuracy of the movement depend on the selection rule as well as the target fiber weights. Although the *expected* movement is a straight line to the target, the movement on any given trial can be inaccurate. Because of the Law of Large Numbers, the more PCs in each APG, the more closely the movement will resemble the expected movement.

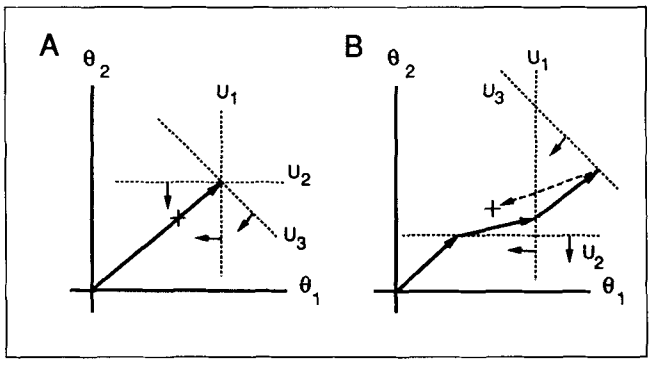

Figure 10. The effect of trajectory generation on changes in connection weights. This figure uses movements generated by three APGs having the same directions of action as those of Figure 9. (**A**) The PCs of the three pattern generators were selected in equal proportions, and the connection weights of the APGs specified three decision lines as shown. The position of the target is shown by the +. In this case the trajectory follows the thick arrow, and the arm stops at the intersection of the three decision lines. Because the movement has overshot the target, the expected direction of the corrective movement is back along the thick arrow. This corrective movement causes weight changes that move the decision lines in the directions shown by the small arrows. (**B**) A movement in which the PCs are selected in equal proportion and the synaptic weights specify the illustrated decision lines. The trajectory is piecewise linear and follows the thick arrows. The expected corrective movement (dashed arrow) causes the decision lines to move in the directions given by the small arrows, which for the case of APG 2 is in the wrong direction.

Learning

We can also analyze how the learning rule interacts with the movements in terms of the decision lines of the participating APGs. Figure 10 shows movements generated by three APGs having the same directions of action as those in Figure 9. Figure 10A shows a movement to a target whose position is indicated by the +. In this case the trajectory, shown by the thick arrow, stops at the intersection of the three decision lines. The expected direction of the corrective movement in this case is back along the arrow to the target. The weight update rule will cause the decision lines to move in the directions given by the small arrows. One can show that updating the weights in this example based on the *expected* corrective movement results in three new decision lines that intersect at a point on the line connecting the target location with the old point of intersection. This situation has the same logic as the one degree-of-freedom case published earlier (Houk et al., 1990).

However, other examples show that the learning rule does not always result in weight changes that are appropriate. Figure 9B illustrates that movements are sometimes dissociated from the values of target fiber weights. The weights used to generate the movements shown in Figure 9B are correct in that the expected movement is correct, but they would be altered by the learning rule in the case shown because this particular movement was not correct. One would hope that other inaccurate movements on subsequent trials would balance these inappropriate weight changes so that the target fiber weights remain near their correct values. For this balancing to occur, we found that the learning rate constants α and β must be small.

Figure 10B shows an example of a movement during which the performance and learning rules of the network interact to degrade performance. It shows a case where the decision lines do not have a common point of intersection. In this example, the trajectory of the arm is shown by the thick arrows. The expected direction of the corrective movement is shown by the dashed arrow. Weight updates based on this corrective movement cause the changes in the decision lines given by the small arrows. These directions are appropriate for APGs 1 and 3 but inappropriate for APG 2: movement accuracy will tend to decrease with training. In the current simulations

we avoided this problem by selecting parameter values that do not allow the weight values to become wildly inaccurate. For example, random initialization of the target fiber weights resulted in decision lines that caused trajectories similar to that shown in Figure 10. Often these trajectories led to cases from which the learning rule could not recover. When the target fiber weights were uniformly initialized to 1, the decision lines for the APGs intersected near the starting location in joint-angle space. Consequently, during the initial training trials, APGs turned off soon after the positive feedback loops were triggered. This largely avoided the problem illustrated in Figure 10, and the correct target fiber weights were consistently learned.

An aspect of network operation that is not illustrated in either Figure 9 or 10 is that the learning rule is suited for movements of particular numbers of time steps. This is because the increases in weights from the increment part of the learning rule [Eq. (10)] depend on the number of time steps that the PC is in the off-state. We view this dependency on movement duration as an undesirable property of the learning rule. However, this property is ameliorated by an aspect of the selection rule by which the number of PCs selected is proportional to the distance of the target. This implies that the network moves the arm faster to more distant targets, thereby tending to keep constant the number of time steps in a movement [see Eq. (8)].

DISCUSSION

We designed the APG array model to investigate how multiple corticocerebellar modules might operate together to control multiple degree-of-freedom arm movements. Our philosophy was to avoid letting control-theoretic issues dominate the model's formulation in favor of constraints from anatomy and physiology. Nevertheless, parallels between the model and behavioral concepts about motor programs and theoretically motivated computational methods emerged when we tried to understand why the model behaved as it did. Here we discuss desirable and undesirable features of the APG array model, emphasizing parallels with existing behavioral concepts and computational schemes.

Distributed Representation of Motor Programs

In the APG array model, APGs control arm movement in parallel so that the activity of all the modules taken together forms a distributed representation. The APG array is said to execute a distributed motor program when it produces a spatiotemporal pattern of activity in the cerebrocerebellar recurrent network, and the patterned outputs that are transmitted from the recurrent network to the spinal cord in descending fibers comprise a distributed motor command. In our current model, each element of the distributed motor command is controlled by an independent APG module. The output of that module calls for a component movement at a particular velocity and in a particular direction. The overall movement of the limb is the vector sum of these components. This section begins with a discussion of the interface between a distributed motor command and the limb, and then proceeds to look backward into the APG array to explore the nature of the distributed motor program that the model produces.

Interface with the Limb

At each instant in time, the command from each of the active APGs increments or decrements the joint angles of the arm. Although this is clearly an oversimplification in that it ignores the dynamics of the muscles and arm, it approximates a situation in which the instantaneous firing rates of the positive feedback loops of the APGs control the velocity of muscle shortening. Because the matrix, A, of weights determining how the APGs influence joint angles is constant, each APG moves the arm in a particular direction in joint-angle space, and the direction of movement in the Cartesian workspace changes with arm position. In a more realistic model, A would vary during movement in a way reflecting the variation of muscle length and moment arms.

The idea that the descending motor system operates in terms of joint or muscle space, instead of Cartesian space, is supported by the work of Caminiti, Johnson, and Urbano (1990) and Hollerbach and Atkeson (1987). Caminiti et al. (1990) found that preferred directions of individual motor cortical neurons in a three-dimensional Georgopoulos task changed with the area of the workspace in which the task was performed. The neurons were not coding a constant direction in the workspace but a direction that changed with the posture of the arm. Such a result is expected from a system that operates in terms of joint-angle or muscle space. Hollerbach and Atkeson (1987) showed that trajectories are not always straight lines in Cartesian space but are sometimes straight lines in joint-angle space. They suggest that planning is done in terms of a staggered joint interpolation strategy that they suggest can account for most experimentally observed trajectories.

In the present simulation, individual APGs show directional specificity similar to the cosine tuning rule found by Georgopoulos et al. (1982). This characteristic tuning arises because, when the correct weights are learned, selection of individual PCs is proportional to the inner product in joint-angle space between a vector pointing to the target from the starting location and a vector pointing in the direction of action of the particular APG. This results in cosine tuning in joint-angle space at

the time of selection because the inner product of two vectors is proportional to the cosine of the angle between them.

A point of difference with Georgopoulos' (1988) results is that the population vector in the present study can be described by cosines in joint-angle space, whereas those in Georgopoulos (1988) are described by cosines in Cartesian space. As Figure 7B shows, in some regions of the workspace the distribution of vectors representing APG activity remains roughly cosine shaped after the transformation to the workspace, and in other regions the distribution is significantly distorted. This discrepancy with the results of Georgopoulos (1988) could also arise from the fact that we used a two-joint, planar arm that moved its endpoint in a plane, whereas Georgopoulos' results came from monkeys that moved their hand position in a plane, but they actually moved their arms in three dimensions. Because the transformation from joint-angle to Cartesian space is different in these two arms, a real monkey's arm might allow for cosine population vectors in both Cartesian and joint-angle space in the regions that were investigated.

In a mathematical analysis of how spatial tuning might arise, Mussa-Ivaldi (1988) showed that experimentally observed cosine tuning would result whenever the cells in question code muscle-related variables. Furthermore, he showed that the alignment of the direction of movement and the direction of the population vector is a result of the cosine tuning and the distribution of the neurons' preferred directions. The present implementation falls under the purview of Mussa-Ivaldi's analysis because the APGs are cosine tuned with a uniform directional distribution (i.e., the vectors A_m point in equally spaced directions). This mathematical analysis is confirmed by the simulations of the present paper, which show that the vector sums at the time of selection point in the direction of the initial movement in Cartesian space. Georgopoulos (1991) has pointed out that only about a third of the motor cortical population has the muscle-related properties required for Mussa-Ivaldi's analysis, and this restriction applies also to the APG array model as currently conceived.

To summarize, the APG array forms a distributed motor representation that works in joint-angle or muscle space. While it seems most natural for the descending motor system to use a muscle-based representation, others have suggested that the descending system uses Cartesian coordinates. A system that worked in terms of Cartesian space has the drawback that it requires an inverse kinematic stage where a Cartesian space trajectory is transformed to joint-angle or muscle space. Such a stage would be computationally expensive and no such inverse kinematic step is contained in the current implementation. The recent results of Caminiti et al.(1990) and Hollerbach and Atkeson (1987) hold out the possibility that a joint-angle or muscle-based representation is actually used.

The Cerebrocerebellar Recurrent Network

The APG array model we have presented uses a highly simplified representation of the complex recurrent network that actually interconnects the cerebellum, motor cortex, and red nucleus (Allen & Tsukahara, 1974; Houk 1989). The cerebellar nuclei project by way of thalamus to the motor cortex and also directly to the red nucleus, and these projections show intricate patterns of topographic specificity as well as divergent features (Futami, Kano, Sento, & Shinoda, 1986; Shinoda, Futami, Mitoma, & Yokota, 1988) that would promote interactions between cerebrocerebellar modules. The motor cortex and red nucleus transmit motor commands to spinal motor neurons, segmental interneurons, and propriospinal neurons via corticospinal and rubrospinal pathways (Kuypers, 1981). As these descending tracts pass through the brainstem, they send prominent collaterals to pontine and medullary neurons that relay back into the cerebellum, thus forming multiple recurrent loops (Allen & Tsukahara, 1974). Positive feedback through these recurrent loops has been shown to support reverberatory activity (Tsukahara, Bando, Murakami, & Oda, 1983) and is considered to provide a fundamental driving force for the generation of motor commands (Houk, 1989; Keifer & Houk, 1991; Eisenman et al., 1991).

In a brief note, Blomfield and Marr (1970) responded to the initial discovery of one of these recurrent pathways (Tsukahara, Korn, & Stone, 1968) with an amendment to Marr's (1969) theory of cerebellar function. They suggested that positive feedback in corticocerebellar loops might give temporal persistence to motor cortical discharge, as is assumed in the APG array model. They further suggested that intrinsic cortical mechanisms would tend to activate excess numbers of cortical neurons, and the role of cerebellar inhibition was seen as making corrections by turning off cortical neurons that had inappropriate contextual relations. Their implicit assumption is that different motor cortical neurons would be used to make a given type of movement, depending on the context of the situation. The recent studies of population activity in the motor cortex reviewed in the previous section instead suggest that virtually the whole population of cortical neurons is used in a graded, vectorial manner to control all voluntary limb movements, independently of their context (Georgopoulos, 1988). This is the tacit assumption followed in the APG array model.

In the simplified model of the recurrent network utilized here, each APG has its own positive feedback loop, and the loops associated with different APGs do not interact. Moreover, we did not explicitly model each positive feedback loop as a recurrent circuit; instead, we represented each loop by a single number giving its activity level. This abstraction of recurrent networks plays an important role in the model nevertheless because it mediates the transformation of inhibitory PC activity into

motor commands. However, more detailed representations of these recurrent networks will be needed in future models because their more complex behavior is essential to our view of both motor command generation and cerebrocerebellar interaction. For example, when the feedback loops of different APGs interact, the properties of the motor command will depend on relationships between the spatial patterns of APG activation and the triggering signals. In a preliminary simulation study, Eisenman et al. (1991) showed how these interactions could give rise to the rotation of the population vector that was observed by Georgopoulos, Lurito, Petrides, Schwartz, and Massey, (1989). These interactions also present a need to inhibit portions of the feedback loops by turning on PCs to prevent loop activity from spreading inappropriately. Thus, one would expect there to be a population of PCs whose activity increases during a movement, as observed in single unit studies (Armstrong, 1988; Fourtier et al., 1989), as well as the population included in the present model whose activity decreases.

Sensorimotor Transformations in the Cerebellar Cortex

The parallel fiber weights in the APG array model can be considered as distributed storage sites for motor programs that implement sensorimotor transformations. The parasagittally organized neuronal circuitry in the cerebellar cortex in essence utilizes these synaptic weights as a recipe for converting the initial position of the limb, sensed by proprioceptive feedback, and a static target, sensed by visual input, into a temporally extended motor command that moves the limb to the target. Once initiated, the motor command is then executed in a quasi-feedforward manner in that proprioceptive feedback is only used to terminate the motor command. The requirements this places on the form of the proprioceptive feedback are relatively weak: during a movement the aggregate proprioceptive input to a selected PC has to increase through a critical level for the first time when the command should terminate. An attractive feature of this quasi-feedforward mode of control is its capacity for using delayed proprioceptive feedback without sacrificing either accuracy or stability. As discussed elsewhere (Houk et al., 1990), a more conventional feedback controller would make errors and would display instability in the presence of the characteristic conduction and mechanical delays in this control system.

The assumption that motor programs are terminated by sensory feedback may seem inconsistent with deafferentation experiments showing that movements can be made in the absence of feedback from the limb (Taub & Berman, 1968). Actually, there is no conflict since, as explained elsewhere (Houk et al., 1990), we believe that Purkinje cells can also utilize efference copy signals as a substitute for sensory feedback. In a more complete version of our theory, movements would be terminated by an additive combination of sensory and efference copy information about the progress of a movement. Due to the nature of these signals, control of slow movements would be dominated by sensory feedback whereas fast movements would be controlled predominately by efference copy, in agreement with common opinion. The proposed additive combination of sensory and efference copy signals contrasts with most previous theories that generally follow von Holst's (1954) suggestion that sensory and efference copy signals are compared by subtraction in order to detect discordances.

Some aspects of the APG model are formally equivalent to the Bullock and Grossberg (1988) model of planned arm movements. In both their model and ours, the initial movement is the result of comparing a vector representing the target with a vector representing current arm position. This information is represented in muscle or joint-angle space. But the two models differ in how they use the target and position information during a movement. Bullock and Grossberg (1988) assume that a difference vector between the target vector and current posture vector is computed at each time during the movement. This difference vector is used to generate the motor output at each time. This is essentially a feedback control configuration and requires undelayed efference copy signals to operate effectively. In contrast, in the APG model the inner product of the target vector and the current posture vector determines the set of APGs selected to generate a motor program that, once triggered, is largely feedforward, with proprioception from the arm terminating the motor command. Further, while Bullock and Grossberg (1988) achieve a motor command with a bell-shaped velocity profile by means of a multiplicative "go" signal, our model produces motor commands that more closely resemble the neural activity that can be recorded from central neurons (Houk & Gibson, 1987; Houk et al., 1990). It is our view that spinal and peripheral mechanisms are responsible for creating the bell-shaped velocity profile. It should also be pointed out that theirs is a formal mathematical model, whereas ours is a distributed neural net that actually implements the computations.

Representations in Parallel Fibers

An important issue not addressed by the model as presented here is how the parallel fiber input to the cerebellum represents targets. We used a representation that is much too simple to represent the diverse, and possibly highly preprocessed, sensory information available about the target and the current state of the organism. This is an area where we expect future research to concentrate in an attempt to develop more realistic representations. Relevant to this development are the theories of cerebellar cortex developed by Marr (1969) and Albus (1971), which are notable for the representation schemes they propose for parallel fiber input to the cerebellum. These

theories postulate that mossy fiber signals are coded in terms of overlapping subsets of active parallel fibers. This allows a very large number of relevant patterns to be represented while enhancing their separability. Using more recent terminology, this is a coarse coding scheme yielding a specific kind of distributed representation (Hinton, 1984). Extending our model by using a form of coarse coding of target locations is relatively straightforward and would permit the model to generalize across movements directed to different targets.

Our current representation of proprioceptive parallel fiber input is also highly simplifed. A more realistic model would use a large set of delayed inputs deriving from a population of stretch receptors having different position and velocity sensitivities and taking origin from many different limb muscles, together with a set of undelayed inputs from various brainstem neurons conveying efference copy signals. Signals with these properties have been recorded from mossy fiber terminals (van Kan, Gibson, & Houk, 1993). In a more realistic model, the Purkinje cells would switch on when the constellation of inputs matches a "perceptual trace" as embodied in the learned synaptic strengths (see the next section). In the current model, instead of representating proprioceptive information with this complexity, we hand-crafted a single, undelayed aggregate position input to fulfill the requirements of each individual APG module. This simplification prevents the PCs in the model from exhibiting the more complex pattern recognition that would be possible with diverse proprioceptive and efference copy information, but it suffices in providing the basic requirement of coding the progress of a movement in a graded fashion. With respect to this requirement, we note that the models of Marr (1969) and Albus (1971) suggest representational schemes that are not suited for all aspects of parallel fiber input to the cerebellar cortex. These schemes are more appropriate for coding target information, where different subsets of active parallel fibers might differentiate between different targets, than for coding proprioceptive information, which may require graded signals on individual parallel fibers.

Parameterized Motor Programs

Motor programs could be stored in a lookup table as detailed lists of highly specific instructions. However, most investigators have favored the idea that memory is used more frugally to store generalized motor programs that are then parameterized in order to control specific actions. Certain features of the APG array model relate well to the ideas about parameterized motor programs discussed by Keele (1973) and the closely related schema ideas promoted by Schmidt (1988), whereas other features relate better to the closed-loop feedback theory advocated by Adams (1971, 1977). The selection phase of the APG array model provides a feasible neuronal mechanism for preparing a parameterized motor program in advance of movement on the basis of a motor schema stored in memory (in parallel fiber synapses). The execution phase is also consistent with the open-loop ideas associated with motor programming concepts, except that we explain the termination of the execution phase as being a consequence of proprioceptive feedback. In this respect, our model follows Adams' (1977) theory, although we believe that muscle spindle receptors are a more likely source of the sensory feedback than are the joint receptors promoted by Adams.

In the APG array model, the counterpart of a generalized motor program is a set of parallel fiber weights for proprioceptive and target inputs, which we can think of as being analogous to Adams' (1971) "perceptual trace." Given these weights, a particular constellation of parallel fiber inputs signifies that the desired endpoint of a movement is about to be reached, causing Purkinje cells to switch to their on-states. Once a "perceptual trace" corresponding to a desired endpoint is learned, the neuronal architecture and neurodynamics of the cerebellar network function in a manner that parameterizes the motor program.

Movement velocity is parameterized in the selection phase of the model's operation. As previously demonstrated, the velocity that is selected by turning off Purkinje cells is automatically scaled so as to depend on the distance between the initial position of the limb and the desired endpoint of the movement. In other words, velocity automatically increases as the amplitude of the movement increases. While this type of scaling is often observed in motor performance studies (Freund & Büdingen, 1978; Ojakangas & Ebner, 1991), velocity can also be varied in an independent manner. Furthermore, Schmidt (1988) reviews data supporting a mechanism whereby velocity scaling can be applied simultaneously to all elements of a motor program to slow down or speed up the entire movement, in a manner analogous to changing the speed of a phonograph record. Although we have not simulated this effect in the present report, it is not difficult to see how the intensity of the entire array of APG modules could be varied by a neuromodulatory substance that altered the "−2" coefficient in Eq. (3), thus scaling up or down the velocity of all elements of the motor program.

Movement duration is parameterized in the execution phase of model operation. To review the steps, the execution phase begins open-loop and continues until Purkinje cells recognize that the endpoint is about to be achieved. They then use feedback to switch to their on-states, thus inhibiting positive feedback in the cerebrocerebellar recurrent network. This terminates the execution phase of the motor program. In this manner, movement duration is a dependent variable that evolves from the course of the movement as opposed to being determined by some internal clock.

Movement amplitude is parameterized by the target

inputs in the model. Each APG module commands elemental motion in a particular direction, and targets corresponding to shorter distances along this direction come to have larger synaptic weights. As a consequence, the Purkinje cells turn on sooner, and the movement is terminated after a smaller amplitude of movement.

Motor Learning

The APG array faces two problems in generating motor commands. The first is to select a set of PCs that will cause the arm to move in the direction of the target; the second is to stop the movement at the target by setting the selected PCs to the on-state. Both of these problems are solved when the target fiber weight is correct as shown in the section on The Execution Phase [Eq. (8)].

Corrective Movements

We assume that the training information conveyed to the APGs is the result of crude corrective movements stimulating proprioceptive receptors. This sensory information is conveyed to the cerebellum by climbing fibers. Learning in the APG array model therefore requires the existence of a low-level system capable of generating movements to spatial targets with at least a ballpark level of accuracy. The lesion studies of Yu, Liu, and Chambers (1980) indicate that such a system exists in cats, and that it can acquire crude capabilities for limb control in response to instrumental conditioning. The observations of rudimentary visuomotor reaching in human infants by Hofsten (1982) also support the existence of a low-level system, since the cerebellum and corticospinal system are poorly developed in the human neonate. Other evidence indicates that when limb movements are not proceeding accurately toward their intended targets, corrective components of the movements are generated by an unconscious, automatic control system (Goodale, Pelisson, & Prablanc, 1986). This corrective system has been studied in human subjects (Gielen & van Gisbergen, 1990; Pelisson et al., 1986; Flash & Henis, 1991), monkeys (Georgopoulos, Kalaska, & Massey, 1981), and cats (Alstermark et al., 1984) by visually switching the targets of the reaching movements. Lesion studies in cats suggest that rapid, corrective movements are mediated by tectal and reticular inputs to a spinal processing stage, called the C3–C4 propriospinal system (Alstermark et al., 1987).

The same spinal system receives additional inputs from the motor cortex and red nucleus that operate with a longer reaction time. It is the latter pathways, corticospinal and rubrospinal, that are held responsible for generating the initial reaching movement. On the basis of feedforward information about the position of a visual target and feedback information about the initial position of the limb, the cerebellum, motor cortex, and red nucleus generate a motor command that is sent to the limb musculature via a neural processing stage in the spinal cord. We assume that collaterals from these commands function to gate off sensory transmission through the proprioceptive climbing fiber pathway, thus preventing sensory responses to the initial limb movement. As the initial movement proceeds, the spinal system receives proprioceptive feedback from the limb and feedforward information about target location from the gaze control system. The latter information is updated as a consequence of corrective eye movements that typically occur after an initial gaze shift toward a visual target (Becker & Fuchs, 1969). Updated gaze information causes the low-level processor to generate a corrective component that is superimposed on the original motor command (Gielen & van Gisbergen, 1990; Pelisson et al., 1986). Since climbing fiber pathways would not be gated off by this low-level corrective process, climbing fibers should fire to indicate the direction of the corrective movement.

Climbing Fiber Signals

We assume that the network by which climbing fiber activity is generated is specifically wired to provide appropriate training information to the APGs (Houk & Barto, 1992). The training signal provided by a climbing fiber is specialized for the recipient APG in that it provides directional information in joint-angle space that is relative to the direction in which that APG moves the arm. The fact that training information is provided in terms of joint-angle space greatly simplifies the problem of providing errors in the correct system of reference. For example, if the network used visual error information, the error information would have to be transformed to joint errors.

To implement this specificity between APGs and their climbing fiber input, the model uses a process sharing some features with the error backpropagation learning algorithm widely used to train layered artificial neural networks (Le Cun, 1985; Parker, 1985; Rumelhart et al., 1986; Werbos, 1974). Climbing fiber activity is generated by proprioceptive signals transmitted to APGs via a network with connection matrix A^T, where A is the matrix of connection weights by which the APGs influence the joint angles. This network has the same structure—but works in the opposite direction—as the network by which the APG array influences joint movement. Climbing fiber activity in the model is, therefore, the result of a simple kind of "backpropagation" process: the descending network (given by A) corresponds to the forward pass through the network, and the ascending network (given by A^T) corresponds to the process by which error information is sent backward through the forward network to provide correct errors to the neurons that generate the descending signal.

Because the APG array model does not involve multiple layers implementing nonlinear transformations, it is simpler than the general error backpropagation algo-

rithm and is most closely related to the use of error backpropagation for "learning with a distal teacher" as suggested by Jordan and Rumelhart (1991). This method addresses learning problems in which training information is available in a "distal" coordinate frame that is different from the "proximal" one in which the learning system generates output signals. Given a network model of the process by which proximal signals influence distal signals, called a *forward model*, propagating distal errors backward through this network translates distal errors into the proximal errors needed for learning. The backpropagation process computes the transpose of the Jacobian of the forward model, which in general depends on the current input to the forward model. Applying the transpose of the Jacobian to the distal error produces the incremental change in the input to the proximal-to-distal process that would decrease this error.

In the case of the APG array model, the distal coordinate frame is joint-angle space in which proprioceptive signals respond to corrective movements, whereas the proximal frame is that of APG array activity. The forward model is the descending motor network itself with connection matrix A. Because this is linear, the transpose of its Jacobian is just A^T. Consequently, the significance of a climbing fiber's response to a corrective movement for the recipient APG is as follows: it tells the APG that the movement has not proceeded far enough along that APG's direction of action. The APG should undergo an incremental adjustment to make it generate a motor command of longer duration. A complication in the present model is that there is no analogous signal—active only in response to a corrective movement—signifying that the duration of the motor command generated by an APG should be shortened. This has implications for the APG adjustment process discussed below.

Note that unlike the error backpropagation algorithm, whether used to train a layered network or for learning with a distal teacher, the APG array model uses separate ascending and descending networks instead of the physiologically implausible process of propagating activity backward through a network. Parker's (1985) version of the error backpropagation algorithm used a separate network in this way to provide training information to the hidden units of a layered network. He described an algorithm in which the weights of the ascending network are adjusted by an adaptive process so that they match the corresponding weights of the descending network. Similarly, some adaptive process must be postulated to account for the alignment between the descending motor network and the ascending network generating climbing fiber activity assumed in our model.

Houk and Barto (1992) suggested that this alignment might come about through trophic mechanisms stimulated by use-dependent alterations in synaptic efficacy. The influence of each APG on movement is adjusted so that if the APG were suitably adjusted through the faster cortical learning process, its activity would tend to diminish the firing probability of its climbing fiber. In the context of the present model, this hypothesis implies that the ascending network to the inferior olive, described by the matrix A^T, is established first, and that the descending network by which APGs influence motoneurons changes so that it is described by the matrix A. This is contrary to the scheme Parker (1985) proposed for error backpropagation in which the ascending network is adjusted through experience to align with the descending network. We have not yet simulated this mechanism to see if it could actually generate the kind of alignment we assume in the present model.

Adjusting APGs

With the training information available in the model, the learning rule used in the present simulations to adjust APGs was capable of finding the correct weights for each of the eight movements involved in the task. However, this required careful choice of parameters and initial conditions. The model's learning process is not robust enough to learn an arbitrary collection of movements in different parts of the workspace. Some of these difficulties were also present in the single APG simulations described by Houk et al. (1990), whereas others are unique to the multiple APG case studied here. In the first category is the sensitivity of the learning process to the relative magnitudes of the learning rate parameters α and β and to the magnitudes of the initial weights. Not all movements can be learned successfully with the same values for these parameters and initial weights. These problems are discussed by Houk et al. (1990). In the second category are the problems illustrated in Figure 10. Movements are made by a group of APGs that once selected act independently to move the arm. In the current model the training signal depends only on the endpoint of the movement, which is itself largely determined by the last few APGs to turn on. This can lead to situations where weights are changed in the incorrect direction, making the learning process unstable. We were able to control this situation in the present simulations by carefully selecting parameters and initial weights, but this strategy will not be effective for more general tasks.

Another shortcoming of the learning process is that it is not capable of adjusting the weights by which the proprioceptive parallel fibers influence PCs. We represented the aggregate proprioceptive input to each APG m by the quantity $p_m(t)$, a specific weighted sum of proprioceptive signals responsive to changes in the joint angles. These weights are the connection weights by which multiple parallel fibers influence the PCs of APG m, and we set them to appropriate values before the start of learning as previously described. Ideally, however, we would like the learning process that adjusts the target fiber connection weights to simultaneously adjust the proprioceptive weights to yield these weight values, or other values having the desired property that each PC

tends to depolarize as the arm moves in the direction of action of that PC's APG. Simulations show that the learning process used in the current model does not accomplish this when it simultaneously adjusts as few as one target fiber weight and one proprioceptive fiber weight. Further, the selection procedure described in the present paper depends on the fact that there is a unique set of correct weights. When both proprioceptive and target fiber weights are adjusted, the solution vector is no longer unique, and neither cosine tuning of population vectors nor reasonably correct trajectories necessarily result.

Although we believe that some features of the learning process used in this paper will survive future development, additional research is needed to describe a learning process that is capable of solving these problems while remaining consistent with anatomical and physiological constraints. This will probably require revising both the learning rule [Eq. (10)] and the process defining the training information. To understand the issues involved, it is useful to contrast the learning process used in the APG array model with one that we know, on theoretical grounds, will operate effectively. Consider a scheme similar to the one we have described except that the synaptic weights of the target fibers are adjusted only when a corrective movement occurs. Further, suppose that in response to each corrective movement the climbing fiber to each APG indicates that the movement was too short or too long along the direction of action of that APG. Finally, suppose the APG adjusts incrementally in response to climbing fiber input to either increase or decrease the duration of the motor command it generates. This error-correction procedure is a straightforward example of learning with a distal teacher and has known theoretical properties (Jordan & Rumelhart, 1991). However, this scheme leaves the APG a black box and places demands on the signaling capabilities of climbing fibers that they may not be able to achieve. Future research will require reconciling computational properties such as these with anatomical and physiological constraints.

Feedback-Error-Learning

It is also relevant to relate our model with that of Kawato (1990) based on *feedback-error-learning*. He postulated that the lateral cerebellum acts as a feedforward controller whose output combines with the output of an extracerebellar feedback controller. As a result of learning, the cerebellum becomes an inverse model of the system being controlled, in this case, the limb. As learning proceeds, the error-correction action of the feedback controller is replaced by feedforward preemption of errors by the commands generated by the inverse model. In Kawato's model, the training signals that climbing fibers transmit to PCs are equal to the output of the feedback controller. Learning occurs as synaptic connections in the cerebellar cortex are adjusted so as to reduce the output of the feedback controller to zero. This feedback-error-learning method has been applied to robot control systems (Kawato et al., 1987; Lane et al., 1988) and has been shown to have desirable convergence properties (Gomi & Kawato, 1990).

The APG array model has some of the features of feedback-error-learning. The extracerebellar mechanism generating corrective movements in the APG array model plays a role similar to that played by the feedback controller in feedback-error-learning. It both corrects movements and provides training information to the cerebellar cortex. The objective of learning in both cases is the same: to reduce the contribution of the corrective component. However, whereas this occurs continuously in feedback-error-learning, in the APG array model it occurs intermittently: the gating of climbing fiber activity by APGs provides error information only when corrective movements are made, thus making the APG array model consistent with the observed low firing rates of climbing fibers. Another difference between feedback-error-learning and the APG array model is the manner in which they implement an inverse model of the limb. In feedback-error-learning a layered neural network is trained to form the inverse model in the form of a function from a target specification to an appropriate motor command. In the APG array model, on the other hand, computing a motor command involves a dynamic process that uses proprioceptive feedback. Despite these differences, we believe that among the more control-theoretic models of the cerebellum, the feedback-error-learning model comes the closest to representing the kind of learning control implemented by the APG array model.

CONCLUSION

This paper describes the current state of our exploration of how motor program concepts may be related to neural mechanisms. We have proposed a model of sensorimotor networks with architectures inspired by the anatomy and physiology of the cerebellum and its interconnections with the red nucleus and the motor cortex. We proposed the concept of rubrocerebellar and corticocerebellar information processing modules that function as APGs capable of the storage, recall, and execution of motor programs. The APG array model described in this paper extends the single APG model of Houk et al. (1990) to an array of APGs whose collective activity controls movement of a simple two degree-of-freedom simulated limb. Our objective was to examine the APG array theory in a simple computational framework with a plausible relationship to anatomy and physiology. Results of simulation experiments show that the APG array model is capable of learning how to control movement of the simulated limb by adjusting distributed motor programs. Although the model is based on many simplifying assumptions, and the simulated motor control task is much simpler than an actual reaching task, these results suggest that

the APG array model may provide a useful step toward a more comprehensive understanding of how neural mechanisms may generate motor programs.

APPENDIX A

Parameter Selection

We used a planar kinematic arm with two joints in the present simulations. The link lengths were obtained from measurements of the upper limb of a rhesus monkey. The shoulder-to-elbow and elbow-to-wrist lengths were found to be 12.2 and 10.8 cm, respectively. Figure 11 shows a sketch of the arm. We used the convention from robotics of measuring joint angles as increasing for counterclockwise rotations. θ_1 was constrained to remain between $-\pi/4$ and π radians, and θ_2 was constrained to remain between 0 and π radians.

To obtain straight-line trajectories in joint-angle space and population vectors that were cosine shaped, ϕ must equal $-w_b$ [see Eq. (8)]. To meet this requirement we set ϕ equal to 1, and w_b equal to -1.

For the arm to stop at target k, w_{kj}^m, the weight of the kth target fiber to PC j of APG m, must be positive and equal to $w_m^* = \phi - w_p p_m(t_1)$ as previously discussed. It is necessary to set w_p, the proprioceptive scale factor, to enable the learning rule to meet this requirement. To determine a suitable value for w_p, we computed the maximum value of $p_m(t)$ at the locations of the eight targets for all of the APGs. We found that $p_m(t)$ was always between 0 and 4. Because $\phi = 1$, setting w_p equal to 0.2 ensures that w_m^* is always larger than 0.2.

The value of k in Eq. (1) determines how fast the arm moves at each time step in the simulation. In the current simulations, each component of $u(t)$, the vector of APG activities, ranged from 0 to 36 (the number of PCs in each APG). We set k to 0.0006 so that according to Eq. (1) $\Delta\theta(t)$ ranged from 0 to 0.0216.

The value of s_j^m depends on the weights of the network and the proprioceptive feedback at different locations in the workspace. Empirical studies showed that the value of s_j^m was never less than -0.2 with the parameters used in the current simulations. This was because the weight of each target parallel fiber was initialized to 1 and the training procedure slowly decreased each of these weights to their correct values.

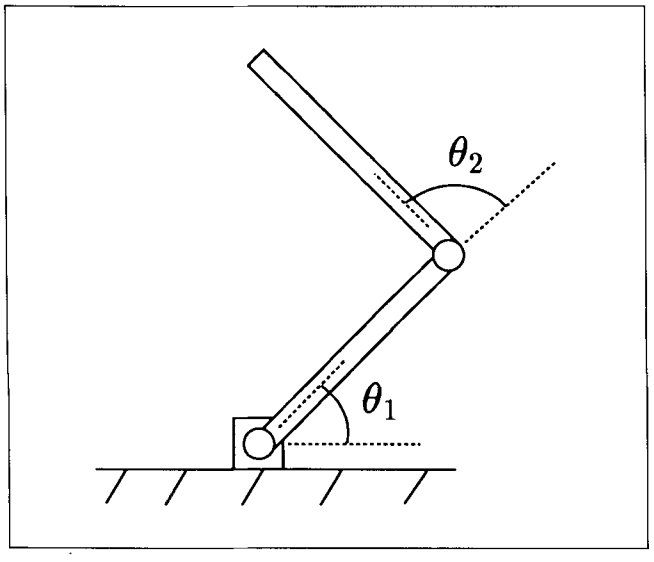

Figure 11. Sketch of the simulated arm. The arm was a two-joint, planar, kinematic arm. θ_1 and θ_2 were measured as shown with counterclockwise rotations of the joints increasing joint angle.

APPENDIX B

Decision Lines

To see that the decision line is perpendicular to the direction of action of APG m, note that PC j of APG m turns on when the proprioceptive feedback, $p_m(t)$, satisfies $w_p p_m(t) + w_{kj}^m(t) = \phi$. Using Eq. (6) to express $p_m(t)$ in terms of $\theta(t)$, we obtain

$$(\phi - w_{kj}^m)/w_p + A_m^T \Pi_m = A_{m1}\theta_1(t) + A_{m2}\theta_2(t)$$

where A_{m1} and A_{m2} are the weights by which APG m influences the first and second joints of the arm. The left-hand side of this equation is a scalar constant, so this is an equation of a line in joint-angle space with slope $-A_{m1}/A_{m2}$, which is the slope of a line perpendicular to the direction of action of APG m.

Acknowledgments

This research was supported by ONR N00014-88-K-0339, NIMH Center Grant P50 MH48185, and a grant from the McDonnell-Pew Foundation for Cognitive Neuroscience supported by the James S. McDonnell Foundation and the Pew Charitable Trusts.

Reprint requests should be sent to James C. Houk, Northwestern University Medical School, Department of Physiology, 303 East Chicago Avenue, Ward Building 5-319, Chicago, Illinois 60611-3008.

Notes

1. The model is a discrete-time model with time steps $t = 1,2, \ldots$. To relate the model's behavior to physiological and behavioral data, the discrete time steps have to be associated with time intervals of specific durations.
2. If the aggregate proprioceptive input were delayed by τ time units, as it would be in a more realistic model, then w_m^* would equal $\phi - w_p p_m(t_1 - \tau)$. Equation (8) becomes $s_j^m(t_0) = w_p A_m^T[\theta(t_0) - \theta(t_1 - \tau)] + \phi + w_b$. The three implications still hold because $\theta(t_0) - \theta(t_1 - \tau)$ is in the same direction, but smaller in magnitude than $\theta(t_0) - \theta(t_1)$.
3. In general, of course, there may be many points in joint-angle space for which the arm reaches the target position. Here we purposefully avoided the issue of excess degrees-of-freedom by using a simple arm in a limited workspace. However, future elaborations of the APG array model must address this issue.

REFERENCES

Adams, J. A. (1971). A closed-loop theory of motor learning. *Journal of Motor Behavior, 3,* 111–149.

Adams, J. A. (1977). Feedback theory of how joint receptors regulate the timing and positioning of a limb. *Psychological Review, 84,* 504–523.

Albus, J. S. (1971). A theory of cerebellar function. *Mathematical Biosciences, 10,* 25–61.

Allen, G. I., & Tsukahara, N. (1974). Cerebrocerebellar communication systems. *Physiological Review, 54,* 957–1006.

Alstermark, B., Eide, E., Gorska, T., Lundberg, A., & Pettersson L.-G. (1984). Visually guided switching of forelimb target reaching in cats. *Acta Neurobiologiae Experimentalis, 120,* 151–153.

Alstermark, B., Gorska, T., Lundberg, A., Pettersson, L.-G., & Walkowska, M. (1987). Effect of different spinal cord lesions on visually guided switching of target-reaching in cats. *Neuroscience Research, 5,* 63–67.

Andersson, G., Armstrong, D. M. (1987). Complex spikes in Purkinje cells in the lateral vermis (b zone) of the cat cerebellum during locomotion. *Journal of Physiology (London), 385,* 107–134.

Armstrong, D. M. (1988). The supraspinal control of mammalian locomotion. *Journal of Physiology (London), 405,* 1–37.

Bartha, G. T., Thompson, R. F., & Gluck, M. A. (1991). Sensorimotor learning and the cerebellum. In M. Arbib & J. Ewert (Eds.), *Visual structures and integrated functions.* Berlin: Springer-Verlag.

Becker, W., & Fuchs, A. F. (1969). Further properties of the human saccadic system: Eye movements and correction saccades with and without visual fixation points. *Vision Research, 19,* 1247–1258.

Bloedel, J. R., & Courville, J. (1981). Cerebellar afferent systems. In *Handbook of physiology* (Section I, Vol. II, Part 2, pp. 735–829). Bethesda MD: American Physiological Society.

Blomfield, S., & Marr, D. (1970). How the cerebellum may be used. *Nature (London), 227,* 1224–1228.

Bullock, D., & Grossberg, S. (1988). Neural dynamics of planned arm movements: Emergent invariants and speed-accuracy properties during trajectory formation. *Psychological Review, 95,* 49–90.

Caminiti, R., Johnson, P. B., & Urbano, A. (1990). Making arm movements within different parts of space: Dynamic aspects in the primate motor cortex. *Journal of Neuroscience, 10,* 2039–2058.

Capaday, C., & Cooke, J. D. (1981). The effects of muscle vibration on the attainment of intended final position during voluntary human arm movements. *Experimental Brain Research, 42,* 228–230.

Collingridge, G. L., & Singer, W. (1990). Excitatory amino acid receptors and synaptic plasticity. *Trends in Pharmacological Science, 11,* 290–296.

Crepel, F. C., & Krupa, M. (1988). Activation of protein kinase C induces a long-term depression of glutamate sensitivity of cerebellar Purkinje cells. An in vitro study. *Brain Research, 458,* 397–401.

Eisenman, L. N., Keifer, J., & Houk, J. C. (1991). Positive feedback in the cerebro-cerebellar recurrent network may explain rotation of population vectors. In F. Eeckman (Ed.), *Analysis and modelling of neural systems* (pp. 371–376). Kluwer Academic Publishers.

Ekerot, C.-F. (1984). Climbing fibre actions of Purkinje cells: Plateau potentials and long-lasting depression of parallel fibre responses. In J. Bloedel et al. (Eds.), *Cerebellar functions* (pp. 268–274). New York: Springer-Verlag.

Flanders, M., & Soechting, J. F. (1990). Arm muscle activation for static forces in three-dimensional space. *Journal of Neurophysiology, 64,* 1818–1837.

Flash, T., & Henis, E. (1991). Arm trajectory modifications during reaching towards visual targets. *Journal of Cognitive Neuroscience, 3,* 220–230.

Fortier, P. A., Kalaska, J. F., & Smith, A. M. (1989). Cerebellar neuronal activity related to whole-arm reaching movements in the monkey. *Journal of Neurophysiology, 62,* 198–211.

Freund, H. J., & Büdingen, H. J. (1978). The relationship between speed and amplitude of the fastest voluntary contraction of human arm muscles. *Experimental Brain Research, 31,* 1–12.

Futami, T., Kano, M., & Sento, S., & Shinoda, Y. (1986). Synaptic organization of the cerebello-thalamo-cerebral pathway in the cat. III. Cerebellar input to corticofugal neurons destined for different subcortical nuclei in areas 4 and 6. *Neuroscience Research, 3,* 321–344.

Gellman, R., Gibson, A. R., & Houk, J. C. (1985). Inferior olivary neurons in the awake cat: Detection of contact and passive body displacement. *Journal of Neurophysiology, 54,* 40–60.

Georgopoulos, A. P. (1988). Neural integration of movement: role of motor cortex in reaching. *FASEB Journal, 2,* 2849–2857.

Georgopoulos, A. P. (1991). Higher order motor control. *Annual Review of Neuroscience, 14,* 361–377.

Georgopoulos, A. P., Kalaska, J. F., Caminiti, R., & Massey, J. T. (1982). On the relations between the direction of two-dimensional arm movements and cell discharge in primate motor cortex. *Journal of Neuroscience, 11,* 1527–1537.

Georgopoulos, A. P., Kalaska, J. F., & Massey, J. T. (1981). Spatial trajectories and reaction times of aimed movements: Effects of practice, uncertainty and change in target location. *Journal of Neurophysiology, 46,* 725–743.

Georgopoulos, A. P., Lurito, J. T., Petrides, M., Schwartz, A. B., & Massey, J. T. (1989). Mental rotation of the neuronal population vector. *Science, 243,* 234–236.

Gibson, A. R., Houk, J. C., & Kohlerman, N. J. (1985). Relation between red nucleus discharge and movement parameters in trained macaque monkeys. *Journal of Physiology (London), 358,* 551–570.

Gielen, C. C. A. M., & Gisbergen van, J. A. M. (1990). The visual guidance of saccades and fast aiming movements. *News in Physiological Science, 5,* 58–63.

Gilbert, P. F. C., & Thach, W. T. (1977). Purkinje cell activity during motor learning. *Brain Research, 128,* 309–328.

Gomi, H., & Kawato, M. (1990). Learning control for a closed loop system using *Feedback Error-Learning. Proceedings of the 29th Conference on Decision and Control,* Honolulu, 3289–3294.

Goodale, M. A., Pelisson, D., & Prablanc, C. (1986). Large adjustments in visually guided reaching do not depend on vision of the hand or perception of target displacement. *Nature (London), 320,* 748–750.

Gutman, A. M. (1991). Bistability of dendrites. *International Journal of Neural Systems, 1,* 291–304.

Hinton, G. E. (1984). Distributed representations. Technical Report number CMU-CS-84-157, Department of Computer Science, Carnegie Mellon University, Pittsburgh, PA.

Hofsten von, C. (1982). Eye-hand coordination in the newborn. *Developmental Psychology, 18,* 450–461.

Hollerbach, J. M., & Atkeson, C. G. (1987). Deducing planning variables from experimental arm trajectories: Pitfalls and possibilities. *Biological Cybernetics, 56,* 279–292.

Holst von, E. (1954). Relations between the central nervous system and the peripheral organs. *British Journal of Animal Behavior, 2,* 89–94.

Houk, J. C. (1989). Cooperative control of limb movements

by the motor cortex, brainstem and cerebellum. In R. M. J. Cotterill (Ed.), *Models of brain function* (pp. 309–325). Cambridge: Cambridge University Press.

Houk, J. C., & Barto, A. G. (1992). Distributed sensorimotor learning. In G. E. Stelmach & J. Requin (Eds.), *Tutorials in Motor Behavior,* II, (pp. 77–100). North-Holland: Elsevier Science Publishers.

Houk, J. C., & Gibson, A. R. (1987). Sensorimotor processing through the cerebellum. In J. S. King (Ed.), *New concepts in cerebellar neurobiology* (pp. 387–416). New York: Alan Liss.

Houk, J. C., Singh, S. P., Fisher, C., & Barto, A. G. (1990). An adaptive sensorimotor network inspired by the anatomy and physiology of the cerebellum. In W. T. Miller, R. S. Sutton, & P. J. Werbos (Eds.), *Neural networks for control* (pp. 301–348). Cambridge, MA: MIT Press.

Ito, M. (1984). *The cerebellum and neural control.* New York: Raven Press.

Ito, M. (1989). Long-term depression. *Annual Review of Neuroscience, 12,* 85–102.

Jordan, M. I., & Rumelhart, D. E. (1991). Forward models: Supervised learning with a distal teacher. Occasional Paper #40, MIT Center for Cognitive Science.

Kawato, M. (1990). Computational schemes and neural network models for formation and control of multijoint arm trajectory. In W. T. Miller, R. S. Sutton, & P. J. Werbos (Eds.), *Neural networks for control* (pp. 197–228). Cambridge MA: MIT Press.

Kawato, M., Furukawa, K., & Suzuki, R. (1987). A hierarchical neural network model for control and learning of voluntary movement. *Biological Cybernetics, 57,* 169–185.

Keele, S. W. (1973). *Attention and Human Performance.* Goodyear Pacific Palisades, California.

Keifer, J., & Houk, J. C. (1991). Role of excitatory amino acids in mediating burst discharge of red nucleus neurons in the in vitro turtle brainstem-cerebellum. *Journal of Neurophysiology, 64,* 454–467.

Kuypers, H. G. J. M. (1981). Anatomy of the descending pathways. In V. B. Brooks (Ed.), *Handbook of physiology* (Section I, Vol. II, Part 1, pp. 597–666). Bethesda, MD: American Physiological Society.

Lane, S. H., Handelman, D. A., & Gelfand, J. J. (1988). Can robots learn like people do? *Proceedings of the SPIE Conference on Applications of Artificial Neural Networks,* Orlando FL.

Le Cun, Y. (1985). Une procedure d'apprentissage pour researu a sequil assymetrique [A Learning procedure for asymmetric threshold network.] *Proceedings of Cognitiva, 85,* 599–604.

Linden, D. J., Dickinson, M. H., Smeyne, M., & Connor, J. A. (1991). A long-term depression of AMPA currents in cultured cerebellar Purkinje neurons. *Neuron, 7,* 81–89.

Llinas, R., & Sugimori, M. (1980). Electrophysiological properties of *in vitro* Purkinje cell dendrites in mammalian cerebellar slices. *Journal of Physiology (London), 305,* 197–213.

Marr, D. (1969). A theory of cerebellar cortex. *Journal of Physiology (London), 202,* 437–470.

Massone, L., & Bizzi, E. (1989). A neural network model for limb trajectory formation. *Biological Cybernetics, 61,* 417–425.

Mussa-Ivaldi, F. A. (1988). Do neurons in the motor cortex encode movement direction? An alternative hypothesis. *Neuroscience Letters, 91,* 106–111.

Nilsson, N. J. (1965). *Learning machines.* New York: McGraw-Hill.

Ojakangas, C. L., & Ebner, T. J. (1991). Scaling of the metrics of visually-guided arm movements during motor learning in primates. *Experimental Brain Research, 85,* 314–323.

Oscarsson (1980). Functional organization of olivary projection to the cerebellar anterior lobe. In J. Courville, C. de Montigney, & Y. Lamarre (Eds.), *The olivary nucleus. Anatomy and physiology* (pp. 207–234). New York: Raven Press.

Parker, D. B. (1985). Learning-logic. Technical Report TR-47, Massachusetts Institute of Technology, Cambridge, MA.

Pelisson, D., Prablanc, C., Goodale, M. A., & Jeannerod, M. (1986). Visual control of reaching movements without vision of the limb: II Evidence for a fast unconscious process correcting the trajectory of the hand to the final position of a target step. *Experimental Brain Research, 62,* 303–311.

Rosenberg, A. F., & Ariel, M. (1990). Visual-response properties of neurons in turtle basal optic nucleus in vitro. *Journal of Neurophysiology, 63,* 1033–1045.

Rosenblatt, R. (1962). *Principles of neurodynamics.* New York: Spartan Books.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning internal representations by error propagation. In D. E. Rumelhart, & J. L. McClelland (Eds.), *Parallel distributed processing. Explorations in the microstructure of cognition,* Vol. 1: *Foundations.* Cambridge, MA: Bradford Books/MIT Press.

Schmidt, R. A. (1988). *Motor control and motor learning.* Champaign, IL: Human Kinetics.

Shinoda, Y., Futami, T., Mitoma, H., & Yokota, J. (1988). Morphology of single neurones in the cerebello-rubrospinal system. *Behavioral Brain Research, 28,* 59–64.

Shinoda, Y., Yokota, J. I., & Futami, T. (1981). Divergent projections of individual corticospinal axons to motoneurons of multiple muscles in the monkey. *Neuroscience Letters, 23,* 7–12.

Sinkjaer, T., Wu, C. H., Barto, A., & Houk, J. C. (1990). Cerebellum control of endpoint position—a simulation model. *Proceedings of IJCNN* (Vol. II, pp. 705–710). Ann Arbor, MI: IEEE Neural Networks Council.

Taub, E., & Berman, A. J. (1968). Movement and learning in the absence of sensory feedback. In S. J. Freedman (Ed.), *The neuropsychology of spatially oriented behavior.* Homewood, IL: Dorsey Press.

Tsukahara, N., Bando, T., Murakami, F., & Oda, Y. (1983). Properties of cerebello-precerebellar reverberating circuits. *Brain Research, 274,* 249–259.

Tsukahara, N., Korn, H., & Stone, J. (1968). Pontine relay from cerebral cortex to cerebellar cortex and nucleus interpositus. *Brain Research, 10,* 448–453.

van Kan, P. L. E., Gibson, A. R., & Houk, J. C. (1993). Movement-related inputs to intermediate cerebellum of the monkey. Submitted to *Journal of Neurophysiology,* in press.

Voogd, J., & Bigaré, F. (1980). Topographical distribution of olivary and corticonuclear fibers in the cerebellum. A review. In J. Courville, C. de Montingny, & Y. Lamarre (Eds.), *The inferior olivary nucleus* (pp. 207–234). New York: Raven Press.

Weiss, C., Houk, J. C., & Gibson, A. R. (1990). Inhibition of sensory responses of cat inferior olive neurons produced by stimulation of red nucleus. *Journal of Neurophysiology, 64,* 1170–1185.

Werbos, P. J. (1974). Beyond regression: New tools for prediction and analysis in the behavioral sciences. Ph.D. Thesis, Harvard University.

Yu, J., Liu, C. N., & Chambers, W. W. (1980). The effects of subcortical sensory and motor lesions on conditioned tactile placing in cats with pyramid section. *Acta Neurobiologiae Experimentalis, 40,* 479–488.

Zipser, D., & Andersen, R. A. (1988). A back-propagation programmed network that simulates response properties of a subset of posterior parietal neurons. *Nature (London), 331,* 679–684.