

# Cerebellar learning of accurate predictive control for fast-reaching movements

Jacob Spoelstra<sup>1</sup>, Nicolas Schweighofer<sup>2</sup>, Michael A. Arbib<sup>3</sup>

<sup>1</sup> Department of Computer Science, University of Southern California, Los Angeles, CA 90089-2520, USA

<sup>2</sup> ERATO – JST, 2-2 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-02, Japan

<sup>3</sup> USC Brain Project, University of Southern California, Los Angeles, CA 90089-2520, USA

Received: 8 February 1999 / Accepted in revised form: 7 August 1999

**Abstract.** Long conduction delays in the nervous system prevent the accurate control of movements by feedback control alone. We present a new, biologically plausible cerebellar model to study how fast arm movements can be executed in spite of these delays. To provide a realistic test-bed of the cerebellar neural model, we embed the cerebellar network in a simulated biological motor system comprising a spinal cord model and a six-muscle two-dimensional arm model. We argue that if the trajectory errors are detected at the spinal cord level, memory traces in the cerebellum can solve the temporal mismatch problem between efferent motor commands and delayed error signals. Moreover, learning is made stable by the inclusion of the cerebello-nucleo-olivary loop in the model. It is shown that the cerebellar network implements a nonlinear predictive regulator by learning part of the inverse dynamics of the plant and spinal circuit. After learning, fast accurate reaching movements can be generated.

## 1 Introduction

What is the role of the cerebellum in the control of reaching movements? Bastian et al. (1996b) demonstrated that the cerebellum compensates for interaction torques that would otherwise push the arm off its desired equilibrium path during fast-reaching movements. While feedback control could, in principle, compensate for interaction torques, it is limited by both long delays in the nervous system and the dynamic properties of muscles and proprioceptors. In the present study, we show how the cerebellum can implement a feedforward, nonlinear predictive regulator by learning part of the inverse dynamics of the arm. After learning, accurate fast movements can be performed in spite of the long conduction delays.

The cerebellar cortex is often viewed as an array of perceptrons (Marr 1969; Albus 1971; Ito 1984). In this theory, the granule cells (GCs) – Purkinje cell (PC) synapses can be modified by climbing fiber (CF) inputs. The input to the cerebellum is characterized by its divergence from the mossy fibers (MF) to the GCs (we refer the reader to Ito 1984 for all the non-referenced data). The GCs are known to give, via their axons, the parallel fibers, excitatory projections to the PCs and to all the inhibitory interneurons (basket, stellate and Golgi cells) of the cerebellar cortex. The Golgi cells (GOs) feed back onto the GCs and have very powerful synapses with longlasting effects. As PCs have inhibitory action upon nuclear cells (collaterals of some MFs excite the nuclear cells), they modulate the signal flow from the nuclear cells. The PCs of the intermediate cerebellum project to the interpositus nucleus (IP), which itself projects to the motor cortex via the thalamus and to the spinal cord via the red nucleus. Thus, the intermediate cerebellum plays a major role in controlling ongoing movements. The climbing fibers, which are the axons of inferior olive cells (IOs), convey signals encoding error in the performance of the system in which the cerebellar subsystem is installed. CF signals induce long-term depression (LTD) in those PF-PC synapses that were activated in conjunction with the climbing fiber.

In the error feedback learning scheme of Miyamoto et al. (1988), a feedback controller acts both to control movement and to provide a teaching signal for the acquisition of an internal inverse dynamics neural model; the feedback error approximates the direction and magnitude of the necessary modifications to the internal neural model. During learning, the feedback torque is slowly replaced by a feedforward torque from the internal model, and movements gradually become more ballistic in nature. Schweighofer et al. (1998a) proposed a model for the role of the intermediate cerebellum in the control of voluntary movement based on feedback error learning. The model embeds a neural network based on known cerebellar circuitry to control a six-muscle two-link planar arm. Simulation results showed that this cerebellar model was able to learn to compensate for

interaction forces, as indicated by an improved tracking performance of desired trajectories after learning. However, the system required a large number of training repetitions and could only learn to control slow movements. This relatively poor performance was due to long delays in the afferent and efferent paths.

Indeed, if the feedback error is computed in long cerebrospinal loops (as in Miyamoto et al. 1988 or Schweighofer et al. 1998a), delays cause the controller to produce a feedback error given (for the simple case of a position feedback error) by:

$$\tau_{fb} = K[\theta_d(t) - \theta(t - \Delta T)] \quad (1)$$

Due to this temporal mismatch, the feedback error would lead to the acquisition of an incorrect inverse dynamics model. To solve this problem, we propose, in a manner reminiscent to Contreras-Vidal et al. (1997), that direct gamma drive to the spinal segments sets up an expectation of the desired state in muscle spindles. Ia afferents, which have been shown to excite IOs (Murphy et al. 1973), act as error detectors. Thus, by detecting errors at the low latency reflex feedback circuit,  $\theta_d$  and  $\theta$  become time aligned, and the training signal is now accurate, albeit delayed:

$$\tau_{fb} = K[\theta_d(t - \Delta T) - \theta(t - \Delta T)] \quad (2)$$

The delay is on the forward path and can be eliminated provided that  $\tau_{fb}$  can be generated at time  $(t - \Delta T)$  in a feedforward manner using the inverse model. Note that this scheme further eliminates the need to compute a weighted difference between cortical desired variables and the sensed state in the inferior olive, and solves the problem of coordinate system transformation and scaling. How can the inverse model be learned if the movements are very fast, i.e. if the feedback errors are severely delayed compared to the efferent commands? We argue that synaptic eligibility permits the inverse model to be learned despite the delayed (but accurate) training signals. The synapses participating in a computation are said to be eligible when a second messenger is released in the PC dendritic spine following parallel fiber activation. The location of the “tagged” synapses solves the spatial credit assignment problem and reproduces the specificity of LTD; if a parallel-fiber-PC synapse participates in synaptic transmission, it becomes eligible to be weakened by LTD if a CF signal is received somewhat later.

To provide a realistic test-bed of the cerebellar model, we embedded the system in a simulated biological motor system. The muscle-driven two-segment arm with spinal reflex circuitry (described in the Appendix) constrains the model in several ways: (1) the spinal delays limit the gain of the spinal feedback control; (2) the redundant muscles allow control of the stiffness of a joint, but add control complexity; (3) circuitry in the spinal cord provides damping. Moreover, although the reflex feedback loops give low-latency compensation to perturbations, they make the arm highly nonlinear when viewed from the higher level controllers. Our results show that the embedded cerebellar model allows stable and robust

control of fast-reaching movements if it combines detection of the trajectory errors at the spinal cord level and memory traces in the cerebellum.

## 2 The model

### 2.1 General architecture

Muscles are often viewed as damped springs with resting lengths determined as a function of the alpha motor command (Ozkaya and Nordin 1991). The result of this property is that a given vector of motor commands to a set of antagonistic muscles defines a point attractor for the limb in joint space. In the equilibrium-point theory (Bizzi et al. 1984, 1992; Feldman 1986), movements are effected by moving this equilibrium point from the starting point to the target position. This model does not seem to be sufficient because measured muscle stiffness values would lead to unrealistic trajectories. McIntyre and Bizzi (1993) have suggested that the spinal reflex path could serve as a low latency position and velocity feedback servo, but Schweighofer (1995) has shown that fast multi-joint movements still require some feedforward compensation to account for interaction torques.

We propose here a new model that bridges the gap between equilibrium-point control and inverse dynamics learning. The model comprises an inverse kinematic model controller (IKM), an inverse static model controller (ISM), and an inverse dynamics model controller (IDM). The IKM generates the required postural commands used to control slow movements and to get the system in the correct ball park by driving the muscle spindles through the gamma system. Because spindles are activated when a muscle is stretched longer than its desired length, which is determined by the gamma drive provided by the IKM, deviations from the desired position activate the spinal servo to provide an additional restoring force. As in the parallel hierarchical control scheme proposed by Katayama and Kawato (1991), the ISM learns the part of the inverse dynamics that depends only on posture, and the IDM provides the part of the inverse dynamics that depend on derivatives of joint angles. The ISM provides the alpha drive to set muscle tensions so that the limb has an equilibrium point at the instantaneous position specified by the trajectory generator. The IDM provides a second input to the alpha motoneurons that compensates for trajectory deviations from the equilibrium path.

In the present model, the IKM is simply computed with standard forward kinematics equations. The ISM, whose muscle output is a simple function of two variables, is a lookup table that is generated off-line. As described in the following, the IDM is implemented as a neural cerebellar model that learns on-line using a biologically plausible mechanism. The spindle activity is used to train the IDM cerebellar controller (see Gomi and Kawato 1993 and Contreras-Vidal et al. 1997).

## 2.2 The cerebellar model

### 2.2.1 Cerebellar inputs

As reviewed in Schweighofer et al. (1998a,b), the intermediate cerebellum receives via MFs extensive inputs from cerebral cortices as well as feedback from Ia muscle spindles. In the present model, inputs from central origins contain representations of the descending postural commands, as well as the desired joint position, velocity and acceleration, and spindles inputs carry arm-state information.

Following Georgopoulos et al. (1986), variables are coded in the activity of a population of neurons with each neuron tuned to a specific value and its output defined as a cosine function of the difference between the value of the variable and the neuron's preferred value. We model a total of 680 MFs; after preliminary simulations, we found that if 340 conveyed desired kinematic variables (position, velocity and acceleration for each joint), 280 target and movement distance information and 60 fibers are Ia afferents from the six muscles, good performance could be obtained. These are not arbitrary numbers, but chosen so that variables could be coded using a large, but manageable arrays of neurons.

Alexander and Crutcher (1990, 1992) identified cells in the supplementary motor areas that fire selectively for different spatial targets while evidence has also been found for a coding of movement distance (Riehle and Requin 1989; Fu et al. 1993). Based on this evidence, the desired arm position was represented using two  $10 \times 10$  populations, receptive to the instantaneous desired arm configuration (Fig. 1A), and eventual target arm configuration (Fig. 1B). In both populations, each neuron is tuned to a specific combination of shoulder and elbow angles. In the model, the neurons are arranged so that the arm configuration is coded as a bump of activity on a grid with shoulder angle represented on the  $y$ -axis, and elbow angle on the  $x$ -axis.

In the velocity group, three populations of neurons are tuned to the current desired velocity (Fig. 1D), peak shoulder velocity (Fig. 1F) and peak elbow velocity (Fig. 1G), respectively. Since premotor neurons also show sensitivity for the direction of movement (Kalaska et al. 1989; Fortier et al. 1993; Fu et al. 1993), each neuron on a  $8 \times 15$  grid representing the current desired velocity input is tuned to a specific point in a polar representation of the signal, i.e. a neuron fires maximally for a specific combination of direction and amplitude of the signal that it is receptive to. Peak velocity signals were coded on  $3 \times 30$  grids using a simple population code where each neuron is broadly tuned to a specific value in the input range. To prevent discontinuity in the inputs, the peak velocity signal is modulated to ramp up before a movement begins and to fade as deceleration begins. As shown in Fig. 1E, the desired joint acceleration values are coded on a  $8 \times 15$  array using the same coding scheme as used for the desired velocity.

Although three of the model input signals, target location and the two peak velocity signals, are not required for inverse dynamics computation, we found that providing these signals improved performance. We note

that, although the cerebellar learning model implements part of an inverse dynamics model, it is not a parametric model and acquires the function as an associative memory. These redundant signals provide an additional context facilitating the recall of specific function points.

Position and velocity errors are sensed by the static and dynamic spindles. The output of each spindle (one static and one dynamic for each muscle) is represented as a  $1 \times 5$  array of fibers, with thresholds set so that a value is coded by both the number and activity of active fibers. Sample output is shown in Fig. 1C. Spindle responses are modeled as linear functions of muscle length and velocity respectively:

$$Ia_{\text{static}}^i = [K_s(I(\theta) - l_r(\gamma_s))]^+ \quad (3)$$

$$Ia_{\text{dynamic}}^i = [K_d(\dot{I}(\theta, \dot{\theta}) - \dot{l}_d(\gamma_d))]^+ \quad (4)$$

where constants  $K_s$  and  $K_d$  were set to 120 and 30, respectively,  $I(\theta)$  and  $\dot{I}(\theta, \dot{\theta})$  are the length and stretch velocity of the muscle as a function of the arm state;  $l_r(\gamma_s)$  is the neutral length of the spindle, modulated by the static gamma drive;  $\dot{l}_d(\gamma_d)$  is the neutral stretch velocity of the dynamic spindle, modulated by the dynamic gamma drive;

$$[x]^+ = \begin{cases} x & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases} \quad (5)$$

### 2.2.2 Cerebellum model

The neural implementation of the IDM is shown schematically in Fig. 2. We used the same cerebellar module described in Spoelstra and Arbib (1997) for learning visuomotor transformations when throwing while wearing wedge prism glasses.

All cells are modeled as leaky integrators with membrane potential defined by

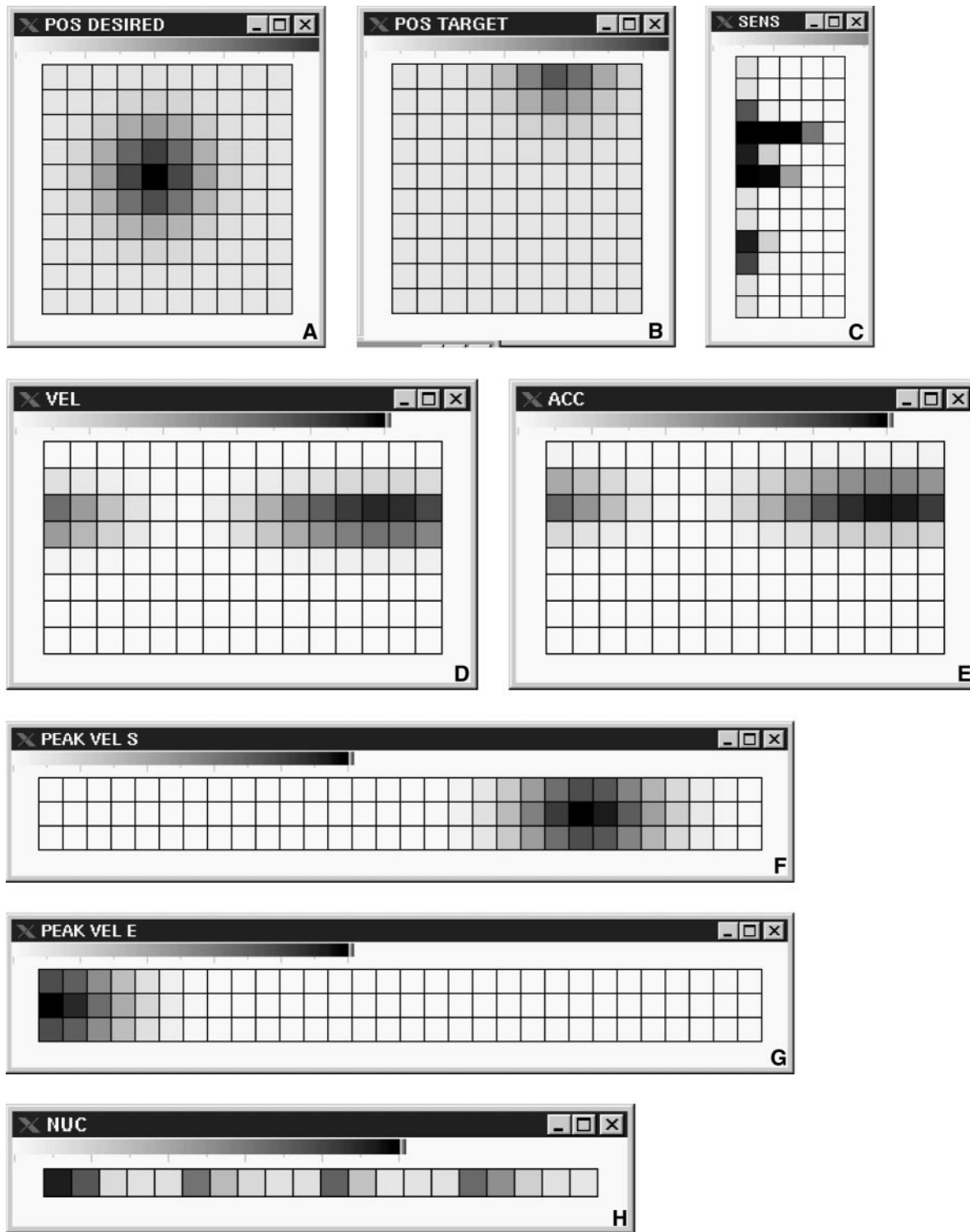
$$\tau \frac{dm}{dt} = -m + x, \quad (6)$$

where  $x$  is the current synaptic input. With the exception of IOs (for which we used the integrate-and-fire model discussed in more detail in Sect. 2.3), the output of each cell is a positive real number representing the instantaneous firing rate  $Y(m)$  and is derived from the membrane potential as

$$Y(m) = Y_{\text{max}} \frac{1}{1 + \exp[-s(m - o)]}, \quad (7)$$

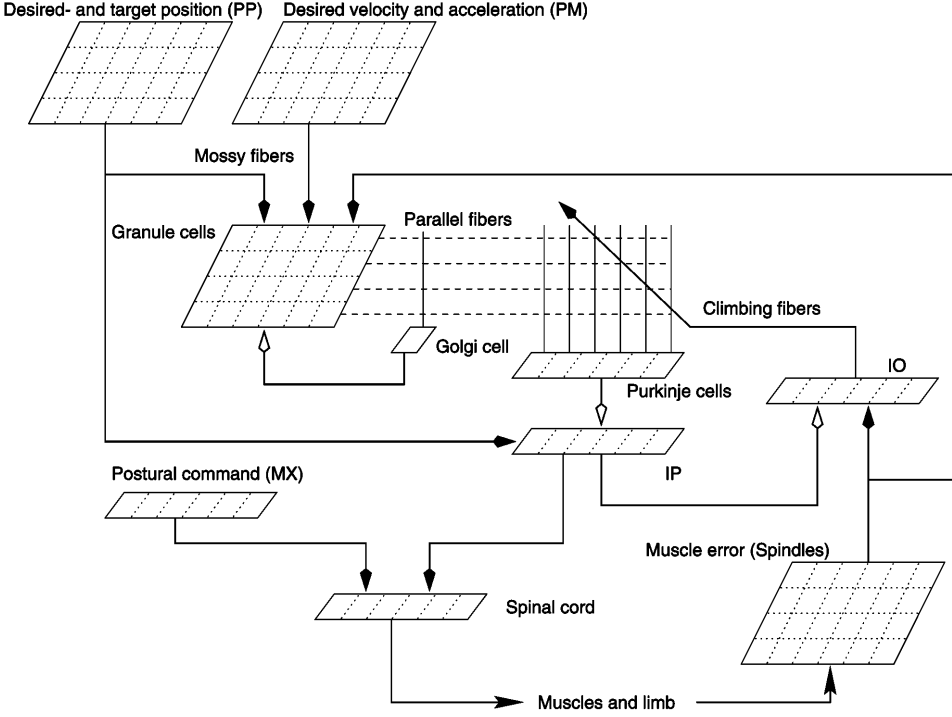
with  $o$  and  $s$  parameters that determine the baseline firing rate and linear range, respectively, of the cell. The parameters used for the different cells in the model are listed in Table 1.

Inputs arrive via MFs to the GCs whose axons bifurcate to form parallel fibers in the cerebellar cortex. Each of 1200 ( $30 \times 40$ ) GCs receives excitatory input ( $w_{\text{mf}} = 0.5$ ) from four randomly selected MFs (the set  $\mathcal{M}$ ) and an inhibitory input ( $w_{\text{go}} = -0.02$ ) from the GOs. The GCs make nonlinear combinations of the inputs to serve as expansion encoders as originally proposed by Albus (1971).



**Fig. 1A–H.** Cerebellar inputs and outputs. Desired position: **A** Instantaneous desired joint angles; **B** current target joint angles are coded on  $10 \times 10$  matrices. Sensory feedback: **C** for each of the six muscles, the output of the static- and dynamic spindle is coded in a  $1 \times 5$  array. Desired velocity: **D** instantaneous desired velocity and **E** acceleration are coded on  $8 \times 15$  matrices using a polar (direction and amplitude) tuning scheme as described in the text. Peak velocity:

**F** peak shoulder velocity and **G** peak elbow velocity are coded on  $3 \times 30$  matrices using simple amplitude tuning. Output representation: **H** nuclear cells code shoulder and elbow flexion and extension synergies with groups of five adjacent cells, using the same convention as for spindle output, i.e. output is equal to the sum of the activity of the five cells



**Fig. 2.** The neural implementation of the inverse dynamic model (IDM) with input and output connections. Brain areas and distinct cell types are implemented as either one-dimensional or two-dimensional layers of neurons. *PP* Posterior parietal cortex; *PM* premotor cortex; *IO* inferior olive; *IP* interpositus; *MX* motor cortex

**Table 1.** Simulation parameters for the different cerebellar neurons. GC, granule cell; GO, golgi cell; PC, Purkinje cell; IP, interpositus nucleus

	GC	GO	PC	IP
Time constant ( $\tau$ )	0.02	0.05	0.02	0.02
Slope ( $s$ )	8	0.5	0.005	0.08
Offset ( $o$ )	0.5	15	750	-50
Maximum rate ( $Y_{max}$ )	100	50	200	100

$$\tau_{gc} \frac{dm_{gc}}{dt} = -m_{gc} - w_{go}GO + \sum_{i \in \mathcal{M}} w_{mf}MF_i . \quad (8)$$

The single GC receives input from all the GCs ( $w_{gc} = 0.00083$ ) and inhibits the entire population in a negative feedback loop.

$$\tau_{go} \frac{dm_{go}}{dt} = -m_{go} + w_{gc}GC . \quad (9)$$

This implementation is in line with the hypothesis that the function of the Golgi-granule loop is to act as a gain control, both maintaining the total energy in the GC layer at a constant level and ensuring a sparse encoding.

In the model, 100 ( $20 \times 5$ ) PCs each receive input from a beam of GCs (the set  $\mathcal{P}$ ) comprising one third of the GC population. The weights  $w_{pf}$  are initialized with random numbers in the range [0:1]. Twenty stellate cells (one for each row of PCs) receive input ( $w_{pfs} = 0.0075$ ) from the same population of GCs as the PCs in their row. Each provides a fixed inhibitory connection to the PCs on its row ( $w_{sc} = -0.04$ )

$$\tau_{sc} \frac{dm_{sc}}{dt} = -m_{sc} + \sum_{i \in \mathcal{P}} w_{pfs}GC_i , \quad (10)$$

$$\tau_{pc} \frac{dm_{pc}}{dt} = -m_{pc} + w_{sc}SC + \sum_{i \in \mathcal{P}} w_{pf}GC_i . \quad (11)$$

As described by Bullock et al. (1994), the inhibition from the stellate cells effectively allows negative GC-PC weights (which gives the PCs greater discriminatory capabilities) while keeping the parallel fiber-Purkinje weights positive as dictated by biology; as long as the stellate cell operates in its linear region, the effective weight from each GC is  $w_{pf} + w_{sc}$ . Because  $w_{sc}$  is constant and negative, the total weight can be regulated to as low as  $w_{sc}$ . PCs also receive a single CF input from the inferior olive. We ignore the real-time effect of this input and use it exclusively for adapting PF-PC synaptic efficacies as described later.

In order to control limb stiffness through coactivation of agonist-antagonist muscle pairs, the descending torque signal for each joint consists of two parts: separate flexion and extension torque commands. In the model, there are 20 NCs. Ten code a shoulder torque, and ten code an elbow torque. For each of the shoulder and elbow torques, five cells code flexion synergy, while the other five cells code extension synergy. The same amplitude coding as was described for the spindle output is used (sample output is shown in Fig. 1H). The vector of 20 ( $4 \times 5$ ) IP cells each receive inhibition ( $w_{pc} = -0.4$ ) from a corresponding column of PCs.

$$\tau_{ip} \frac{dm_{ip}^j}{dt} = -m_{ip}^j + \sum_i w_{pc}PC_{ij} . \quad (12)$$

The parameters of IP cells are chosen to ensure a high spontaneous rate, so that PCs “carve out” the IP firing profile. NCs make inhibitory synapses with the same IOs that give rise to the CFs innervating the PCs from which the NC receives input, closing the loop. The torque-like output of the cerebellar module then has to be converted to commands in muscle space. We assume that this is handled in the brainstem as described in Appendix B.

### 2.3 Learning

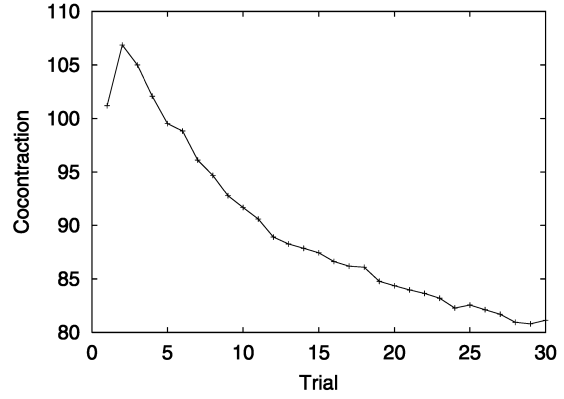
Mauk and Donegan (1997) showed that LTD of parallel fiber-Purkinje synapses, together with the known recurrent projections between cerebellar NCs and IOs (Kim et al. 1998), produce a stable input-following learning system. Similarly, to allow stable, realistic learning, we model the self-regulating cerebello-olivary loop that allows learning to be stable for a large range of learning parameters and avoids the need to resort to synaptic normalization for stable learning.

Each IO cell receives excitatory spindle afferents and a topographic inhibitory projection from an IP cell. In turn, it projects to the column of PCs connected to that IP cell where its effect is to modify  $w_{pf}$ , the strength of parallel fiber-Purkinje cell synapses. Activity of IO cell is computed as:

$$\tau_{io} \frac{dm_{io}^j}{dt} = -m_{io}^j - IP^j + \sum_i w_{Ia(S)}^i Ia_{static}^i + \frac{1}{2} w_{Ia(D)}^i Ia_{dynamic}^i \quad (13)$$

The weights  $w_{Ia(S)}^i$  and  $w_{Ia(D)}^i$  are set to +1 for muscles that contribute to torques in the same direction as the IDM microcomplex that the IO cell forms part of: -1 for antagonist muscles. This setup will train the system to provide an additional torque if a muscle is stretched more than expected, while at the same time penalizing cocontraction. A feature of learning in biological systems is that early in the process of adaptation a strategy of high co-contraction (giving higher limb stiffness and improving accuracy) is adopted, but with practice the level of co-contraction is reduced (Ghez 1991). As shown in Fig. 3, this feature is reproduced by the current cerebellar system; by subtracting the signals produced by the spindles in the antagonist muscle, the system will first try to relax an activated muscle before increasing the activation of the antagonist, so that co-contraction is minimized to produce effective reciprocal muscle activations.

IO cells are known to fire at very low rates – spontaneous firing occurs at about 2 Hz, while the maximum rate is a little over 10 Hz. This presents a challenge to the model, since the entire movement lasts only on the order of 600 ms, which would allow for a maximum of six error-correcting inputs/movement. Because such a biological limitation could not be ignored, we used an integrate-and-fire IO model; the output of IO cells are binary signals to indicate the presence or absence of an action potential (similar to the IO model developed in



**Fig. 3.** The effect of learning on coactivation of antagonist muscle pairs. The plot shows the sum of the coactivation of the three antagonist muscle pairs, averaged over the 16 movements as a function of learning trials. As in a biological system learning reduces the level of coactivation

Schweighofer et al. 1998a,b), rather than the real-valued rate output used for the other neurons. Formally,

$$\begin{aligned} &\text{if } m_{io} > V_{\text{threshold}} \\ &\text{then } m_{io} = m_{io} - V_{\text{rebound}}; \text{IO} = 1 \\ &\text{else IO} = 0 \end{aligned} ,$$

with  $\tau_{io} = 0.055$ ;  $V_{\text{threshold}} = -0.0001$ ; and  $V_{\text{rebound}} = 1$ .

The traditional view of cerebellar learning is that  $\Delta w_{pf} \propto -[GC][IO]$ . However, the Ia efferents (and therefore the IO error signals) are delayed, so when updating weights, we have to change them proportional to the GC activity some time earlier. We propose here that the concept of synaptic eligibility (Sutton and Barto 1981; Klopf 1982) solves this temporal credit assignment problem. The process of LTD has been shown to involve second messengers whose concentration can tag synapses eligible for modification (Houk et al. 1990). Schweighofer et al. (1996) suggested that the second messenger concentration might follow second-order dynamics to peak at a specified instant after parallel fiber activity, and that this could be used to solve the temporal credit assignment problem in saccade adaptation where the error information is available only after the control action. We used the same method and show that the cerebellum learns to associate current control actions with prior states and learns to act as a predictive feedforward controller. A second-order differential equation effectively keeps a memory trace of GC activity ( $e$ ) in each synapse, peaking approximately 100 ms after the event:

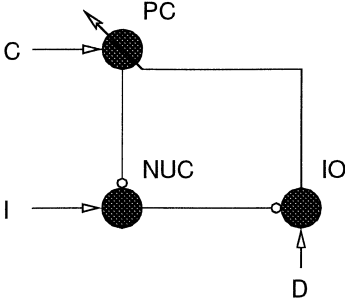
$$\tau_e \frac{de_1}{dt} = -e_1 + GC \quad , \quad (14)$$

$$\tau_e \frac{de}{dt} = -e + e_1 \quad , \quad (15)$$

with  $\tau_e = 0.1$ .

The learning rule used for each PF-PC synapse can be formalized as:

$$\Delta w(t) = -\alpha e(t)[y_{IO} - \beta] \quad , \quad (16)$$



**Fig. 4.** Simplified schematic of the cerebello-olivary loop. Nuclear cells (*NUC*) receive inhibition from Purkinje cells (*PC*) and inhibit the same inferior olive (*IO*) cells involved in the adaptation of said *PC*s. *C*, *I* and *D* denote “context”, “input” and “desired” (training signals), respectively. This setup adapts  $PC(C)$  until  $NUC(I) = D$

with  $\alpha = 0.0002$  the learning rate,  $y_{IO}$  the climbing fiber input,  $\beta = 2.0$  the baseline *IO* firing rate, and  $e(t)$  defined as above. From (16), it is clear that *IO* firing above baseline will produce LTD, while firing below baseline will produce LTP as originally proposed by Fujita (1982).

Figure 4 shows a simplified schematic of the cerebello-olivary loop. Assuming simple linear models for the neurons and a linear learning rule, we have

$$[PC] = Cw \quad (17)$$

$$[NUC] = 1 - [PC] \quad (18)$$

$$[IO] = D - [NUC] \quad (19)$$

with learning rule

$$\Delta w \propto -C[IO] \quad (20)$$

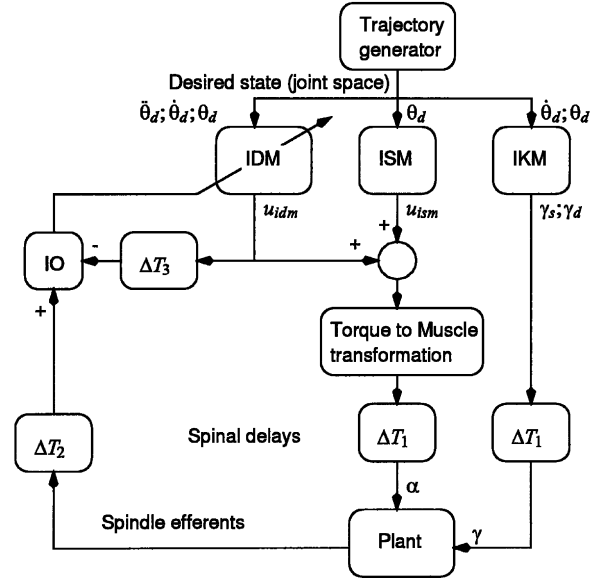
Where  $[PC]$ ,  $[NUC]$  and  $[IO]$  represent the firing rate of a Purkinje, nuclear and inferior olive cell, respectively;  $C$  represents a context provided by parallel fibers;  $w$  is the synaptic efficacy of parallel-fiber-*PC* synapses;  $D$  is a training signal activating the inferior olive.

Rewriting (20) gives

$$\Delta w \propto -C(D - 1 + Cw) \quad (21)$$

The system is stable (Kenyon et al. 1998) and  $w$  will exponentially approach the equilibrium value of  $(1 - D)/C$ , to give  $[NUC] = D$ . In terms of the cerebellar system, this implies that training will drive the cerebellar nuclear cells to predict (as described below) any excitatory inferior olive afferent signals. Thus, in our model, input from the spindles determine the *IO* firing (13), which in turn determine the output of the *IP*s (16). The net effect is that the cerebellar model *IDM* produces a signal similar to the Ia spinal efferents (signaling muscle errors) which opposes the error and causes improved trajectories.

However, delays would cause this signal to be applied too late. Synaptic eligibility (described above) solves this temporal credit assignment problem by shifting the blame earlier in time so that the corrective actions are associated with an earlier system state,



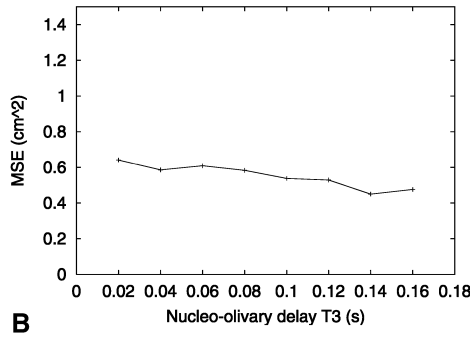
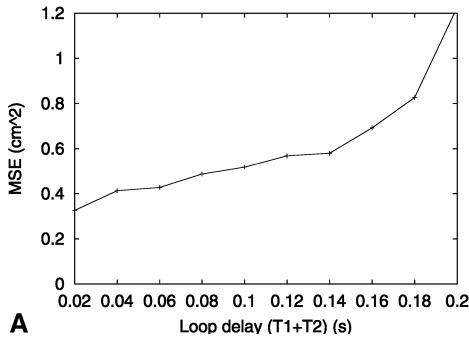
**Fig. 5.** Simplified schematic of the control system. The trajectory generator defines a minimum jerk kinematic trajectory, i.e. at each time step, the desired position, velocity and acceleration are computed for each joint so that the hand would move along a straight line towards the target with bell-shaped velocity profile. The inverse static model (*ISM*) provides alpha-muscle control input to define a static equilibrium point for the limb at the current desired position. The inverse kinematic model (*IKM*) generates static and dynamic gamma drive for the spindles to detect length and velocity errors at the muscles. The inverse dynamic model (*IDM*), implemented as a cerebellar model, uses the sensed muscle tension produced by the spinal reflex circuitry as teacher signal and learns to associate this with the system state (provided by the trajectory generator and muscle spindles as discussed in the text) to provide corrective control signals in a feedforward manner

allowing the *IDM* to provide adaptive, predictive control, as shown by Barto et al. (1999). A further effect of the synaptic eligibility trace is to smooth the control signal over the temporal evolution of the movement, favoring solutions that minimize motor command changes.

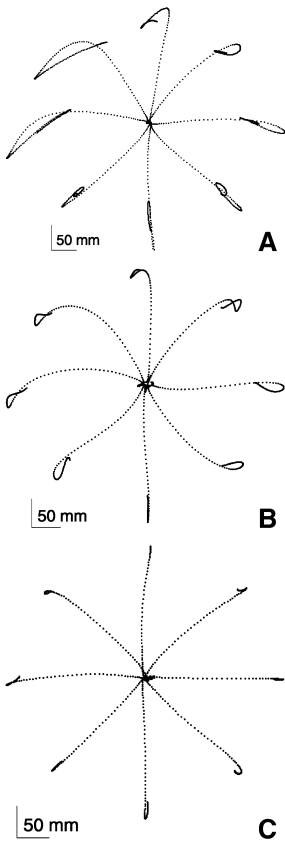
One temporal problem remains: the cerebellar output must be aligned with the error signal in the cerebello-olivary loop. Thus, we incorporate a delay  $\Delta T_3$  (set to 100 ms) as shown in Fig. 5 between the *NC*s and the *IO*. This delay matches approximately  $\Delta T_1 + \Delta T_2$ . Such a long-latency response has been shown to exist in cats by Ruigrok and Voogd (1995). Simulation experiments (as shown in Fig. 6B) suggest, however, that this delay need not precisely match the spinal loop delay. Learning performance varied by less than 10% for variations in  $\Delta T_3$  of 0.02–0.16 s.

### 3 Results

Simulation results that show the effectiveness of the system to learn accurate trajectory control for fast movements are presented in Fig. 7. Movements are made from a central position to eight radial targets at a rate of 0.6 s/movement with loop delays  $\Delta T_1 =$



**Fig. 6A,B.** Sensitivity to different delays. The eligibility time constant was kept constant while the system delays were varied. The small change in performance error indicates the robustness of the system. **A** Effect of varying spinal delays  $\Delta T_1 + \Delta T_2$ . **B** Effect of varying the nucleo-olivary delay  $\Delta T_3$



**Fig. 7A–C.** Tracking results. Movements are made from a central position to eight radial targets. Each segment of the pattern is completed in 0.6 s. **A** Tracking performance before learning, i.e. using only postural control; **B** the best tracking performance using a proportional-derivative (PD) controller shown for comparison; **C** performance of the cerebellar model after learning

$\Delta T_2 = 50$  ms, and  $\Delta T_3 = 100$  ms. The trajectory in Fig. 7A is produced using only the IKM and spinal reflex controllers by moving the equilibrium point along the desired trajectory. Coriolis, centripetal and inertial forces cause the limb to stray from the prescribed path.

For comparison, Fig. 7B show the best attempt using a proportional derivative (PD) controller. In this case, torques for each joint were generated using

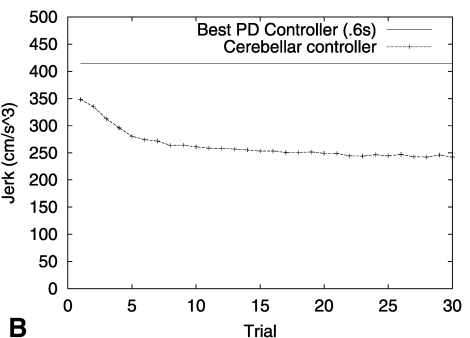
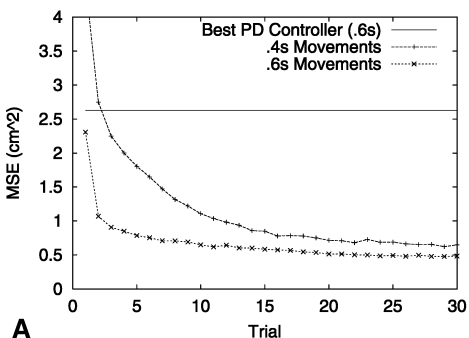
$$\tau = K_v(\dot{\theta}_d - \dot{\theta}) + K_p(\theta_d - \theta) \quad (22)$$

with  $\dot{\theta}_d$  and  $\theta_d$  desired joint velocity and position, respectively. Optimal gain values of  $K_v = 0.11$  and  $K_p = 0.01$  were experimentally determined. These torques were not directly applied to the joints, but simply used to replace the output of the cerebellar controller, so this controller had to deal with the same delays and nonlinearities of the spinal circuit. The delays in the system limit the gains, with the result that a rather large MSE remains.

The trajectory in Fig. 7C was produced by the full system after 30 learning trials; the trajectories are almost straight.

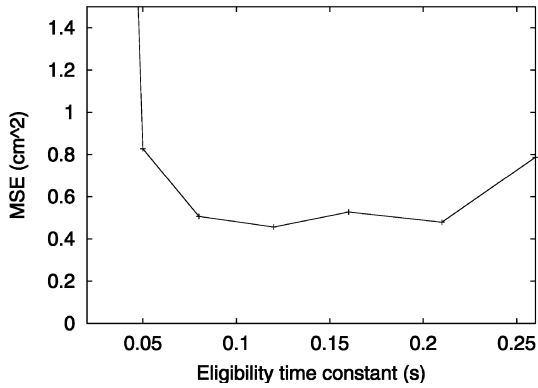
The results shown in Fig. 8A confirm that accurate trajectories can be generated in a small number visits to each target. The system also reduces the integral of jerk while learning, as can be seen in Fig. 8B.

The sensitivity of the MSE to changes in the eligibility profile is shown in Fig. 9. A value of 0.01 gives a profile with a sharp peak near zero, effectively eliminating the eligibility trace. It is clear that having a trace with a delayed peak improves performance, with the best performance for cases where the peak occurs at around 100 ms (which coincides with the value chosen for  $\Delta T_1 + \Delta T_2$ ).



**Fig. 8A,B.** Learning performance for two movement speeds. The performance of the best proportional-derivative (PD) controller is shown for reference. **A** Tracking error (MSE in  $\text{cm}^2$ ) and **B** normalized integral of jerk as a function of learning trials





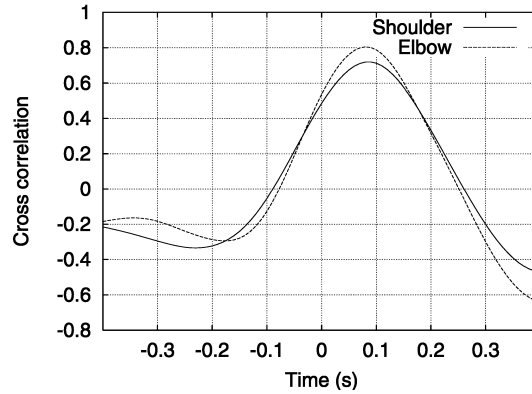
**Fig. 9.** Sensitivity of the eligibility profile. A value of 0.01 produces a sharp peak at 0, corresponding to no eligibility trace and caused the system to diverge. A value of 0.1 produces a trace that peaks around 100 ms

A key feature of the model is the ability to learn, despite long delays, by temporal alignment of error signals with the inputs at the time the outputs were generated. The eligibility trace does not produce a pure delay, but rather ‘smears’ the inputs over a time with dynamics that have time constants close to the actual delays in the control path. This has the advantage that the match does not have to be perfect. The results shown in Fig. 6A and B were obtained by varying the loop delays while fixing the eligibility time constant. Although the eligibility profile peaks at around 100 ms, performance degrades only slightly for a large mismatch in the actual delay.

Finally, to show that the cerebellar model does in fact learn predictive feedforward control, we computed the cross correlation between the training inputs arriving at the inferior olive and the outputs generated by the cerebellar model. As shown in Fig. 10, the correlations peak at values of 0.05–0.1 s, indicating that the cerebellar output leads the training signals by that time.

#### 4 Conclusion

In this paper, we have shown that a biologically inspired cerebellar network was able to learn accurate trajectory control in spite of long delays in the feedback error. We further proposed a new model that bridges the gap between equilibrium-point control and inverse dynamics learning. We assumed that a postural module generates the required postural commands to control slow movements and to get the system in the correct ball park. The trajectory deviations from the equilibrium path are detected at the muscle spindle level and are used to train an inverse dynamics model. The proposed architecture uses synaptic eligibility to time-align delayed error signals (from the spindles) with earlier system states in order to learn predictive feedforward control. In Schweighofer et al. (1996), synaptic eligibility was used for saccadic eye movements to learn a gain



**Fig. 10.** Cross correlation between the output produced by the cerebellar module and training signals arriving at the IO taken over one trail of 16 movements after learning. The peaks to the right of zero indicate that the output of the cerebellar module leads that of the training signal

adaptation from a delayed error. Here, we show that it is also useful for on-line learning of arm control.

How could our model be mapped on the central nervous system? Thach et al. (1982a,b, 1986) have shown that the cerebellum is involved in the differential control of alpha and gamma motoneurons. Moreover, four lines of evidence suggests that the cerebellum plays a role in controlling muscle spindles (Ito 1984): (1) signals of groups Ia, Ib and II muscle afferents reach the cerebellar cortex; (2) cerebellectomy affects the dynamic characteristics of the stretch reflex; (3) the stretch reflex exhibits a high degree of adaptability, but cerebellar patients are unable to adapt the long-latency reflex gain; (4) cerebellar stimulation affects the stretch reflex. Thus, we propose that the IKM is located in the cerebellum. The IKM cerebellar controller generates the required postural commands by driving the muscle spindles through the gamma system to control slow movements via the spinal reflex circuitry, thereby providing approximate control that facilitates the learning process. Fortier et al. (1993) studied the responses of motor cortical cells and cerebellar neurons during whole-arm movements. Motor cortical neurons were found to be more related to active maintenance of different arm postures and cerebellar neurons were found to be more related to phasic movements. Moreover, both populations exhibited shifts in activity during movements. Bastian et al. (1996a) found that cerebellar subjects performing fast-reaching movements often generate inappropriate muscle torques relative to the dynamic interaction torques. The inability to produce muscle torques that compensate for the dynamic interaction torques appears to be an important cause of the deficits shown by cerebellar subjects during reaching. From these studies, we predict that the motor cortex is responsible for the generation of static torques (ISM) and the cerebellum is responsible for the generation of dynamic torques (IDM). The ISM provides alpha drive to set muscle tensions so that the limb has an equilibrium point at the instantaneous position specified by the trajectory generator. The IDM provides a second input to

the alpha motoneurons that compensates for trajectory deviations from the equilibrium path. We further propose that the cerebellar IDM controller is trained by the errors between the specified muscle lengths (as specified by the IKM) and actual muscle lengths (via the inferior olive; Murphy et al. 1973).

Biological control of arm movements has attracted a lot of attention from modelers in the last 20 years. Here, we compare several previous models to the present one.

1. Cerebellar model articulation controller (CMAC) (Albus 1975a,b; Miller et al. 1990): Our implementation of a cerebellar model is far removed from a hash table used in the CMAC, but population coding of inputs give GCs local receptive fields in input space which encapsulates the principle of the CMAC architecture.

2. Adjustable pattern generator (APG) models (Berthier et al. 1993; Fagg et al. 1997): We do not use the positive feedback loops to build up nuclear activity, thus do not require the bistable PCs to shut down the activity. Instead, we learn a continuous, real-valued output in the cerebellar cortex and assume that MF activity, combined with a natural high baseline firing rate will provide the NC activity. Moreover, the present model includes arm dynamics.

3. Feedback-error learning (Miyamoto et al. 1988; Kawato and Gomi 1992; Gomi and Kawato 1993): The learning principle is similar, but we added explicit treatment of delays. In contrast to Schweighofer et al. (1998a) or the feedback error learning model (Miyamoto et al. 1988), we did not use the long-loop feedback controller as a means to drive learning. This solves the problem of delays that cause the feedback system to generate faulty training signals. Instead, we use the servo mechanism of the spinal reflex circuitry as a key element to allow kinematic control of slow movements for the default controller, and the short loop spinal feedback circuit to drive learning.

4. Virtual trajectory (Katayama and Kawato 1993): In the present model, we do not explicitly change the equilibrium trajectory by changing the desired kinematic path. The model just adds feedforward assistive torque.

5. Inverse dynamics cerebellar learning models (Schweighofer et al. 1998a,b): The current model evolved from these models, but takes into account a

number of desirable features: We added a postural control system utilizing spinal reflexes; spindle afferents are used as more plausible trajectory error detectors; the cerebello-olivary loop is exploited for stable learning in a way that drives the cerebellum to provide predictive torque compensation for accurate movements; and we explicitly address the problem of long delays.

6. VITE/FLETE (Contreras-Vidal et al. 1997): This model is closest to the present one, in terms of the basic principles of operation. In their model, cerebellar learning modifies velocity commands from the VITE system to produce feedforward signals to the FLETE system in order to improve trajectories. They use FLETE as the spinal controller, VITE to provide feedforward motor commands and drive the gamma system, and then also use spindle errors to drive IO. However, their cerebellar network has a single PC per muscle whose function it is to gate VITE velocity commands from agonist and antagonist channels on an IP cell. As in our model, the output of the IP cell is added to the motor command, but they also use it to modulate the spinal reflex circuit. Finally, in Contreras-Vidal et al. (1997), no delays are taken into account and the movements are unrealistically slow (several seconds).

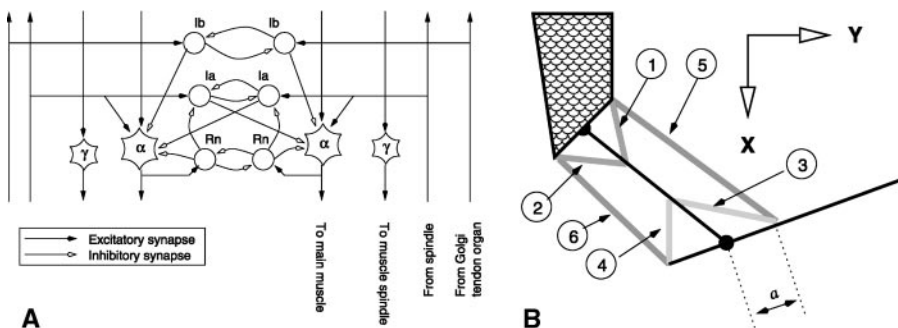
In summary, unlike previously published models, the model developed in the present paper permits the control of realistically fast and accurate arm movements, while retaining a high degree of biological plausibility. For robotic applications, we have shown that the cerebellar architecture is suitable for fast, on-line learning of a complex control function.

## Appendix A. Arm model

With an eye on robotic implementation, the plant simulated was a two-segment planar arm, actuated by six antagonistic McKibben artificial muscles (Chou and Hannaford 1996) as shown in Fig. A1B.

The muscle tension (as a function of motor command  $u$ ) is defined as:

$$T(u) = [K_g(K_p u - P_a)(l - l_0) + T_a]^+ , \quad (23)$$



**Fig. A1.** **A** The neural circuit used to implement a spinal segment (redrawn from Chou and Hannaford, 1997). Descending commands excite alpha- and gamma-motoneurons which drive the muscles and spindles, respectively. Feedback from spindles excite the alpha-motoneuron while inhibiting the antagonist muscle through Ia

interneurons. Muscle tension is sensed by Golgi tendon organs and provide feedback via Ib interneurons. **B** Schematic representation of the muscle attachments on the two-link arm. Six muscles are arranged in antagonist pairs. Two sets actuate single joints, while the third is a set of bi-articulate muscles

**Table A1.** Muscle constants. With the view of possible robotic implementation, these values were chosen to model McKibben artificial muscle pneumatic actuators, as reported by Chou and Hannaford (1996)

Muscle	Description	$l_0$	$a$
1	Shoulder flexor	0.26	0.04
2	Shoulder extensor	0.26	0.04
3	Elbow flexor	0.275	0.025
4	Elbow extensor	0.275	0.025
5	Biarticate flexor	0.237	0.028 and 0.035
6	Biarticate extensor	0.237	0.028 and 0.035

**Table A2.** Muscle parameters.  $l_0$  is the resting length of the muscle with no motor command. The value is chosen to be the maximum length such that, with a motor command of zero, the muscle would have a non-zero tension for all arm postures.  $a$  is the offset of the attachment point from the joint and determines the effective moment arm. The values were taken from Katayama and Kawato (1993)

Constant	Description	Value
$K_g$	Spring constant	0.466
$P_a$	Activation threshold	0.062
$K_p$	Input scaling constant	5000
$T_a$	Offset	-16.0

where  $K_g, K_p, P_a$  and  $T_a$  constants as defined in Table A1,  $l$  the length of the muscle as determined by the geometry of the limb and  $l_0$  the passive resting length of the muscle (See Table A2).

In addition, Ia-pathway kinematic feedback was provided from each muscle by static and dynamic spindle pairs; separate gamma drive inputs set a “desired” length and velocity, the spindle output is a clipped (positive only) function of the position or velocity error. Simulated Golgi tendon organs provide force feedback to the Ib interneurons. The output was modeled as a linear function of muscle tension:

$$G = K_g T, \quad (24)$$

With  $T$  the muscle tension and constant  $K_g$  set to 0.2 to ensure that receiving Ib neurons do not saturate.

## Appendix B. Spinal motor pattern generators

Realized torques are related to muscle tension as follows:

$$\tau = AT, \quad (25)$$

with  $T$  the vector of six muscle tensions,  $\tau = [\tau_s^f \tau_s^e \tau_e^f \tau_e^e]^T$  the flexion and extension torques and

$$A = \begin{bmatrix} a_1 \sin(\theta'_s) & 0 & 0 & 0 & a_5 \sin(\theta'_s) & 0 \\ 0 & a_2 \sin(\theta'_s) & 0 & 0 & 0 & a_6 \sin(\theta'_s) \\ 0 & 0 & a_3 \sin(\theta_e) & 0 & a_7 \sin(\theta_e) & 0 \\ 0 & 0 & 0 & a_4 \sin(\theta_e) & 0 & a_8 \sin(\theta_e) \end{bmatrix}$$

with  $\theta'_s = \theta_s + \pi/4$ .

$T$  was computed from  $\tau$  using the Moore-Penrose inverse of  $A$ :

$$T = A^\# \tau. \quad (26)$$

To simplify the angle dependencies of the moment, arms were ignored so that  $A^\#$  could be calculated only once and held constant. Direct application of this pseudo inverse could produce negative

muscle tensions. To prevent this, we note that the muscles are symmetrical around each joint and simply move negative values in the constant  $A^\#$  to the opposing muscle and change the sign to give the positive inverse  $A_p^\#$ .

Making the simplifying assumption that muscle tension is proportional to descending motor commands, motor synergies are computed as:

$$u_x = K_x A_p^\# \tau, \quad (27)$$

with  $K_x = 0.04$ .

## Appendix C. Spinal segment circuitry

A neural model of the spinal segment circuitry implemented the motor servo, with the circuit shown in Fig. A1A duplicated for each antagonist pair of muscles. We used a model that was implemented in DSP hardware by Chou and Hannaford (1997) for single joint posture control. As shown in Fig. A2A, the model incorporates alpha and gamma motoneurons, Renshaw cells, Ia and Ib interneurons.

We modeled only one neuron of each type for each muscle. All neurons are modeled as leaky integrators with a positive real-valued output representing the instantaneous firing rate of the neuron. The firing rate ( $f$ ) is computed from the membrane potential ( $p$ ) using the standard sigmoid function:

$$f(p) = F_{\max} \frac{1}{1 + \exp(-\alpha(p - \beta))} \quad (28)$$

with  $F_{\max}$  the maximum firing rate,  $\alpha$  determining the slope and  $\beta$  the offset of the sigmoid. The membrane potential is governed by the differential equation

$$\tau \frac{dp}{dt} = -p + I, \quad (29)$$

with  $I$  the current synaptic input. Table A3 shows the parameter values for each of the neuron types, while the connection weights are listed in Table A4.

Updating of the network is parallel, i.e. for every time-step the synaptic inputs for all neurons are computed, using current external inputs and firing rates computed the previous time-step, before membrane potentials and firing rates are updated. To

**Table A3.** Spinal circuit neuron parameters

Type	$F_{\max}$	$\alpha$	$\beta$	$\tau$
$\alpha$ -Motoneuron	1.0	10.0	0.5	0.02
$\gamma$ -Motoneuron	1.0	10.0	0.5	0.1
Ia interneuron	1.0	10.0	0.5	0.1
Ib interneuron	1.0	10.0	0.5	0.1
Renshaw cell	1.0	10.0	0.5	0.1

**Table A4.** Spinal circuit weights for network connections as shown in Fig. A1A

Source	Destination			
	$\alpha$	Ia	Ib	Rn
$\alpha$				1.0
Ia	-0.5	-0.5		
Ib	-0.5		-0.25	
Rn	-0.25	-0.25		-0.5
Spindle	0.5	0.25		
Golgi tendon			1.4	

demonstrate, external input to an alpha-motoneuron at time  $t_i$  will increase the firing rate of the neuron in the current simulation time-step, but Renshaw cell activity will only be affected at time  $t_{i+1}$  and the inhibitory effect on the alpha-motoneuron will only be seen at  $t_{i+2}$ .

*Acknowledgements.* This work was supported in part by the Human Brain Project (with funding from NIMH, NASA, and NIDA) under the P20 Program Project Grant HBP: 5-P20-52194 for work on "Neural Plasticity: Data and Computational Structures" (M.A. Arbib, Director). The authors thank Stefan Schaal and Etienne Burdet for providing many helpful comments and suggestions; also Blake Hannaford and his group at the University of Washington for explaining the intricacies of their muscle and spinal reflex models.

## References

- Albus JS (1971) A theory of cerebellar function. *Math Biosci* 10:25–61
- Albus JS (1975a) Data storage in the cerebellar model articulation controller (CMAC). *Trans ASME (Am Soc Mech Eng) J Dynamic Syst Measurement Control* 97:228–233
- Albus JS (1975b) A new approach to manipulator control: the cerebellar model articulation controller (CMAC). *Trans ASME (Am Soc Mech Eng) J Dyn Sys Meas Control* 97:220–227
- Alexander GE, Crutcher MD (1990) Neural representations of the target (goal) of visually guided arm movements in three motor areas of the monkey. *J Neurophysiol* 64:164–178
- Alexander GE, Crutcher MD (1992) Control of goal-directed limb movements in primates: neurobiological evidence for parallel, distributed motor processing. In: Kien J, McCrohan CR, Winlow W (eds). *Neurobiology of motor programme selection: new approaches to the study of behavioural choice*. Manchester University Press, New York, pp 203–224
- Barto AG, Fagg AH, Sitkoff N, Houk JC (1999) A cerebellar model of timing and prediction in the control of reaching. *Neural Comput* 11:565–594
- Bastian AJ, Martin TA, Keating JG, Thach WT (1996a) Cerebellar-ataxia – abnormal control of interaction torques across multiple joints. *J Neurophysiol* 76:492–509
- Bastian AJ, Mink JW, Thach WT (1996b) A specific movement deficit due to focal lesion of the inferior cerebellar vermis in children. *Ann Neurol* 40:121–121
- Berthier NE, Singh SP, Barto AG, Houk JC (1993) Distributed representation of limb motor programs in arrays of adjustable pattern generators. *J Cogn Neurosci* 5:56–78
- Bizzi E, Accornero N, Chapple W, Hogan N (1984) Posture control and trajectory formation during arm movement. *J Neurosci* 4:2738–2744
- Bizzi E, Hogan N, Mussa-Ivaldi FA, Giszter S (1992) Does the nervous system use equilibrium-point control to guide single and multiple joint movements? *Beh Brain Sci* 15:603–613
- Bullock D, Fiala JC, Grossberg S (1994) A neural model of timed response learning in the cerebellum. *Neural Netw* 7:1101–1114
- Chou CP, Hannaford B (1996) Measurement and modeling of McKibben pneumatic artificial muscles. *IEEE Trans on Robot Autom* 12:90–102
- Chou CP, Hannaford B (1997) Study of human forearm posture maintenance with a physiologically-based robotic arm and spinal level neural controller. *Biol Cybern* 76:285–298
- Contreras-Vidal JL, Grossberg S, Bullock D (1997) Neural model of cerebellar learning for arm movement control: cortico-spinocerebellar dynamics. *Learn Memory* 3:475–502
- Fagg AH, Sitkoff N, Barto AG, Houk JC (1997) Cerebellar learning for control of a two-link arm in muscle space. *ICRA 97*, pp 2638–2644
- Feldman AG (1986) Once more on the equilibrium-point hypothesis ( $\lambda$  model) for motor control. *J Mot Behav* 18:17–54
- Fortier PA, Smith AM, Kalaska JF (1993) Comparison of cerebellar and motor cortex activity during reaching: directional tuning and response variability. *J Neurophysiol* 69:1136–1149
- Fu QG, Suarez JJ, Ebner TJ (1993) Neuronal specification of direction and distance during reaching movements in the superior precentral premotor area and primary motor cortex of monkeys. *J Neurophysiol* 70:2097–2116
- Fujita M (1982) Adaptive filter model of the cerebellum. *Biol Cybern* 45:195–206
- Georgopoulos AP, Schwartz AB, Kettner RE (1986) Neuronal population coding of movement direction. *Science* 233:1416–1419
- Ghez C (1991) Muscles: effectors of the motor systems. In: Kandel ER, Schwartz JH, Jessell TM (eds) *Principles of neural science*, 3rd edn, chapter 36. Appleton and Lange, Norwalk, Conn pp 548–563
- Gomi H, Kawato M (1993) Neural network control for a closed-loop system using feedback-error-learning. *Neural Netw* 6:933–946
- Houk JC, Singh SP, Fisher C, Barto AG (1990) An adaptive sensorimotor network inspired by the anatomy and physiology of the cerebellum. In: Miller WT, Sutton RS, Werbos PJ (eds) *Neural networks for control*. MIT Press, Cambridge, mass, pp 301–348
- Ito M (1984) *The cerebellum and neural control*. Raven Press, New York
- Kalaska JF, Cohen DAD, Hyde ML, Prud'homme M (1989) A comparison of movement direction-related versus load direction-related activity in primate motor cortex, using a two-dimensional reaching task. *J Neurosci* 9:2080–2102
- Katayama M, Kawato M (1991) A parallel-hierarchical neural network model for motor control of a musculo-skeletal system. *Sys Comput Jpn* 22: 95–105
- Katayama M, Kawato M (1993) Virtual trajectory and stiffness ellipse during multijoint arm movement predicted by neural inverse models. *Biol Cybern* 69:353–362
- Kawato M, Gomi H (1992) A computational model of 4 regions of the cerebellum based on feedback-error learning. *Biol Cybern* 68:95–103
- Kenyon GT, Medina JF, Mauk MD (1998) A mathematical-model of the cerebellar-olivary system. I. self-regulating equilibrium of climbing fiber activity. *J Comput Neurosci* 5:17–33
- Kim JJ, Krupa DJ, Thompson RF (1998) Inhibitory cerebello-olivary projections and blocking effect in classical-conditioning. *Science* 279:570–573
- Klopf H (1982) *The hedonistic neuron: a theory of memory, learning and intelligence*. Hemisphere, Washington, DC
- Marr D (1969) A theory of cerebellar cortex. *J Physiol (Lond)* 202:437–470
- Mauk MD, Donegan NH (1997) A model of Pavlovian eyelid conditioning based on the synaptic organization of the cerebellum. *Learn Memory* 4:130–158
- McIntyre J, Bizzi E (1993) Servo hypotheses for the biological control of movement. *J Mot Behav* 25:193–202
- Miller TWI, Glanz FH, Kraft LG (1990) CMAC: an associative neural network alternative to backpropagation. *Proc IEEE* 78:1561–1567
- Miyamoto H, Kawato M, Setoyama T, Suzuki R (1988) Feedback-error-learning neural network for trajectory control of a robotic manipulator. *Neural Netw* 1:251–265
- Murphy JT, MacKay WA, Johnson F (1973) Responses of cerebellar cortical neurons to dynamic proprioceptive inputs from forelimb muscles. *J Neurophysiol* 36:711–723
- Ozkaya N, Nordin M (1991) *Fundamentals of biomechanics: equilibrium, motion, and deformation*. Van Nostrand Reinhold, New York
- Riehle A, Requin J (1989) Monkey primary motor and premotor cortex: single cell activity related to prior information about direction and extent of an intended movement. *J Neurophysiol* 61:534–549
- Ruijgrok TJH, Voogd J (1995) Cerebellar influence on olivary excitability in the cat. *Eur J Neurosci* 7:679–693

- Schweighofer N (1995) Computational models of the cerebellum in the adaptive control of movements. PhD thesis, University of Southern California, Los Angeles
- Schweighofer N, Arbib MA, Dominey PF (1996) A model of the cerebellum in adaptive-control of saccadic gain. I. the model and its biological substrate. *Biol Cybern* 75:19–28
- Schweighofer N, Arbib MA, Kawato M (1998a) Role of the cerebellum in reaching movements. I. distributed inverse dynamics control. *Eur J Neurosci* 10:86–94
- Schweighofer N, Spelstra J, Arbib MA, Kawato M (1998b) Role of the cerebellum in reaching movements in humans. II. a neural model of the intermediate cerebellum. *Eur J Neurosci* 10:95–105
- Spelstra J, Arbib MA (1997) A computational model of the role of the cerebellum in adapting to throwing while wearing wedge prism glasses. *Proc 4th Joint Symposium on Neural Computation*, vol 7, Los Angeles, pp 201–208
- Sutton RS, Barto AG (1981) Toward a modern theory of adaptive networks: expectation and prediction. *Psychol Rev* 88:135–170
- Thach WT, Perry JG, Schieber MH (1982a) Cerebellar output: body maps and muscle spindles. *Exp Brain Res [Suppl]* 6:440–453
- Thach WT, Schieber MH, Gilbert PFC, Elble RJ (1982b) Cerebellar learning in limb movements. In: Woody CD, (ed) *Conditioning*. Plenum, New York, pp 401–416
- Thach WT, Schieber MH, Mink J, Kane S, Horne M (1986) Cerebellar relation to muscle spindles in hand tracking. *Prog Brain Res* 64:217–224