A Model of Cerebellar Learning for Control of Arm Movements Using Muscle Synergies[*]

Andrew H. Fagg[1] Nathan Sitkoff[1] Andrew G. Barto[1]
James C. Houk[2]

{fagg | sitkoff | barto}@cs.umass.edu
[1]Department of Computer Science
University of Massachusetts
Amherst, MA 01003-4610

houk@casbah.acns.nwu.edu
[2]Department of Physiology
Northwestern University School of Medicine
Chicago, IL 60611-3008

Abstract

Biological control systems have long been studied as possible inspiration for the construction of robotic controllers. The cerebellum is known to be involved in the production and learning of smooth, coordinated movements. In this paper, we present a model of cerebellar control of a muscle-actuated, two-link, planar arm. The model learns in a trial-and-error fashion to generate the appropriate sequence of motor signals that accurately bring the arm to a specified target. The motor signals produced by the cerebellum are specified in ***muscle synergy space****. When the cerebellum fails to bring the arm to the target, an extra-cerebellar module performs low-quality corrective movements, from which the cerebellum updates its program. In learning to perform the task, the cerebellum constructs an implicit inverse model of the plant. This model uses a combination of delayed sensory signals and recently-generated motor commands to compute the new output motor signal.*

1 Introduction

Modern robotics research is concerned with the control of complex plants. Such plants exhibit non-trivial dynamics and potentially long feedback delays. However, in order to be successful, many control techniques require accurate models of both the plant and the environment with which the plant interacts. In biology, both actuator and plant are in a constant state of flux, and behave in a complex and non-linear fashion. Delays of the sensory-motor loops are typically measured in hundreds of milliseconds. Millions of years of evolution have developed biological controllers that are very good at controlling these systems. These controllers do not rely on high-quality, pre-defined models of the plant. Rather, the control algorithm is tuned incrementally through experience with the environment. It is not surprising, then, that we should turn to biology for inspiration.

The primate cerebellum is known to be involved in the learning and production of coordinated movements [18]. In our earlier work, we proposed a biologically-inspired model of cerebellar learning for control of a one degree-of-freedom (DOF) mass-spring system [9], and a two-DOF, muscle-actuated arm [6]. This model does not rely on a high-quality, built-in model of the plant, but rather acquires an implicit model through experience.

The work is extended in this paper in two key directions. First, control signals generated by the learning controller are in a *muscle synergy space*. An individual muscle synergy activates a collection of the available muscles in a coordinated fashion. This language of motor outputs is more akin to what is available to motor areas of the brain, including the cerebellum [14].

Second, we introduce a simple, visual-based extra-cerebellar (EC) controller for producing corrective movements. When the learning cerebellar controller is unable to bring the arm to the specified target, the EC controller produces a short burst of muscle activity that moves the arm closer to the target. Due to the crude implementation of the EC module, however, the arm moves in short, quick hops, and does not necessarily move directly toward the target. Despite these properties, the cerebellar controller learns from these corrective movements, to generate the appropriate motor signals for bringing the arm to the target.

2 System Architecture

The system architecture, outlined in Figure 1, consists of two control modules (the *extra-cerebellar* (EC) and *cerebellar* modules), a spinal/muscle processing system, the arm plant, and a module responsible for estimating movement errors (representing the *inferior olive*).

[*]This work is supported by the National Institutes of Health (grant #NIH 1-50 MH 48185-02)

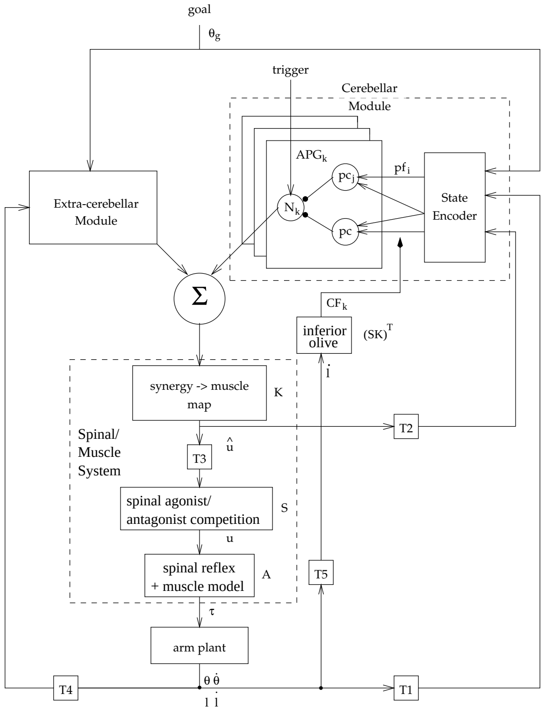

Figure 1: The System Architecture. Two control modules (cerebellar and extra-cerebellar systems) produce signals in muscle synergy space. These signals are transformed by the spinal/muscle system into torque commands that drive the arm plant. Movement errors are detected by the inferior olive, which are used to update the cerebellar control program.

The plant used in our experiments is a simulated, two degree-of-freedom, planar arm (Figure 2). Actuation of the arm is performed by three pairs of opposing muscles. Two pairs of muscles individually actuate the shoulder and elbow joints, while the third pair actuates both joints simultaneously. Each pair of muscles pull against one-another. Activation of the *agonist* muscle results in a flexion of the joint(s); activation of the *antagonist* results in joint extension. A single muscle behaves as a non-linear visco-elastic (spring) element. Control of the arm is achieved by setting the rest length of each muscle.

The learning controller (the cerebellar module) is constructed from an array of *adjustable pattern generators* (APGs) [3, 4], each of which drives a single *muscle synergy.* These APGs combine the current target position with delayed sensory and motor efference signals to produce motor commands that are intended to bring the arm to the specified goal. In cases where the arm does not reach this goal, the static EC module is activated. This module produces a short, constant burst of activity in muscle synergy space that brings the arm closer to the goal.

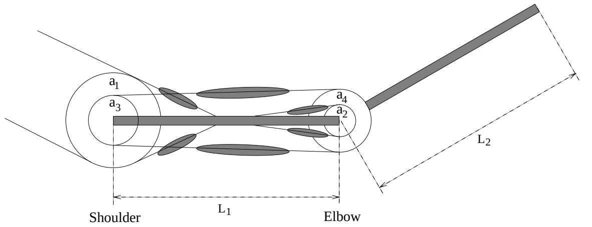

Figure 2: A muscle-actuated, 2 DOF, planar arm (Figure adapted from [11]).

The contributions of the two control modules are combined in the spinal/muscle system, which transforms muscle synergy signals into joint torques. Each *muscle synergy* [14] activates a subset of muscles. Motor commands for opposing muscles then reciprocally inhibit one-another (S). Finally, multiple muscles induce torques about individual joints, as described by transform A.

Errors in cerebellar-generated movements are assessed by the *inferior olive* (IO), and are measured in muscle synergy space. This is accomplished indirectly by observation of muscle length changes in response to the corrective motor commands generated by the EC module. The IO-generated signals are then used to adjust the APGs contained within the cerebellar controller.

2.1 Muscle Synergies

In the brain, neurons that send axons down the spinal cord (to motor neurons) do not typically affect single muscles. Instead, an individual cell often projects to a group of related muscles [14]. This makes the learning-to-control problem more difficult. Not only must the system decide which muscles to activate to move the limb appropriately, but it must also recruit the right combination of muscle synergies to achieve the desired muscle activation.

In the model described here, we make use of a total of eight muscle synergies. Each of these synergies activate a subset of the six arm muscles. We refer to the eight synergies as: S (shoulder agonist), $\bar{S}$ (shoulder antagonist), E (elbow agonist), $\bar{E}$ (antagonist), $S\bar{E}$ (shoulder agonist, elbow antagonist), $\bar{S}E$ (shoulder antagonist, elbow agonist), SEB (shoulder, elbow, and biarticulate agonist), and $\bar{S}\bar{E}\bar{B}$ (antagonists).

Figure 3 illustrates the behavior of each muscle synergy as a function of position in the workspace. Each "spider" corresponds to a single starting point. Indi-

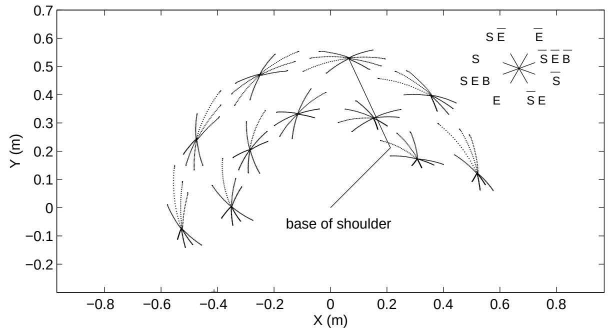

Figure 3: Behavior of the arm as different synergies are activated at several points in the workspace. Each "spider" traces the movement of the endpoint of the arm in reaction to activation of a single muscle synergy. The muscle synergy labels (upper right of Figure) correspond spatially to the upper, center spider.

vidual legs of a spider are the result of tracking the endpoint of the arm in response to activation of one muscle synergy for 50 ms. Movement in arbitrary directions can be obtained by appropriately mixing several muscle synergies. Note that as the shoulder rotates through its range, the spiders also rotate in a corresponding fashion. Hence, individual muscle synergies approximately encode movements in a cylindrical coordinate system.

2.2 Cerebellar Module

In the brain, *mossy fibers* (MFs) carry plant state, motor efference, and other contextual signals into the cerebellum [5]. These fibers impinge on *granule cells*, whose axons give rise to *parallel fibers* (PFs). Through the combination of inputs from multiple classes of MFs and local inhibitory interneurons, the granule cells are thought to provide a sparse, expansive encoding of the incoming state information [1]. In our model, such an encoding is computed by the *state encoder* (Figure 1) using a *CMAC* (Cerebellar Model Arithmetic Computer) [1, 15]. This coding scheme makes use of multiple tilings over the state space. A single tiling partitions the space defined by several state variables (e.g. θ_e, $\dot{\theta}_e$, and $\hat{u}_1$) into discrete, non-overlapping rectangular volumes (tiles). When the system state falls into a particular tile, the tile is given an activation level of 1, and all others within the tiling are set to 0. Like Albus, we imagine that each tile corresponds to a single granule cell [1]. Due to the large number of state variables, it is impractical for a single tiling to effectively partition the entire space. Thus, a number of tilings are used in parallel, each making use of different subsets of state variables. Although this coding scheme is not biologically plausible, it captures the computational essence of the biology, while being inexpensive to simulate.

The large number of PFs converge on a much smaller set of *Purkinje cells* (PCs). The state of the PC j is computed as follows:

$$PC_j \quad = \quad g(\sum_i PF_i \times w_{ij}),$$

where PC_j is the cell's firing rate, and w_{ij} is the (variable) strength of the synapse from PF_i to PC_j. $g()$ is defined as the threshold-ramp function:

$$g(x; \rho, \bar{\rho}, \lambda, \bar{\lambda}) = \begin{cases} 0 & x < \lambda \\ \rho + (\bar{\rho} - \rho)\frac{x - \lambda}{\bar{\lambda} - \lambda} & \lambda \leq x \leq \bar{\lambda} \\ \bar{\rho} & otherwise, \end{cases}$$

where λ and $\bar{\lambda}$ define the lower and upper threshold of the ramp, and ρ and $\bar{\rho}$ define the output range of the ramp.

A set of Purkinje cells, in turn, provide inhibitory signals to a single cerebellar nuclear cell. The activation of the nuclear cell determines the output of the APG. The control signal output by APG_k is computed as follows:

$$u_{C_k} = 1 - \frac{1}{|L_k|} \sum_{j \in L_k} PC_j,$$

where L_k is the set of Purkinje cells that are within APG k.

2.3 Extra-Cerebellar Module

For a given movement, the cerebellar control module may generate a pattern of muscle activity that does not bring the arm to rest at the target position. In this case, the role of the hard-wired extra-cerebellar (EC) module is to produce one or more corrective movements. It is not necessary, however, that a perfect correction be produced. Rather, the minimal constraints are that 1) the corrective movement makes non-trivial progress toward the goal, and 2) the movement generally does not overshoot the goal.

In the model, corrective movements are generated by short, constant bursts of muscle activity, that are computed from the target error. The target error is measured in a Cartesian coordinate frame that is rooted at the endpoint of the arm, and is oriented such that the Y-axis passes through both the wrist and the center of shoulder rotation (Figure 4). In this scheme, errors along the Y-axis ($+e/-e$) translate directly into movements of the elbow, whereas errors along the X-axis yield movements of the shoulder. Although this

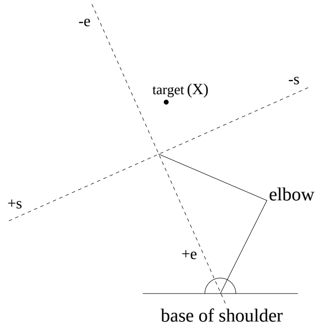

Figure 4: Extra-cerebellar computation of the corrective movement. The target error (X) is measured in a Cartesian coordinate frame that is rooted at the endpoint of the arm.

is a very crude mechanism for generating movements to the target, it satisfies both the constraints stated above. More importantly, as we shall see, these crude corrective movements provide sufficient information for the cerebellar module to learn an effective control policy, without the need for a high-quality, pre-defined model of the plant.

3 Learning Algorithm

When the cerebellum does not bring the arm to the target, requiring the extra-cerebellar module to generate a corrective movement, the learning algorithm must determine which of the PF-to-PC synapses should be updated in order to improve the cerebellum's movement generation performance. This issue, referred to as the *credit assignment problem* consists of two components [8]. The *structural credit assignment problem* is that of determining *which* synapses should be updated. The *temporal credit assignment problem* is that of properly attributing error signals received at one instant to synaptic events that have taken place in the recent past.

3.1 Structural Credit Assignment

In animals, the inferior olive (IO) is known to be the source of climbing fiber (CF) inputs into the cerebellum, which are thought to carry some form of error information related to the current movement [10]. Studies in cat show that when a limb is passively moved by an external force, CFs fire in response to the proprioceptive inputs from the the limb [7]. In addition, the CFs are differentially responsive to the direction of limb movement. During active movements (such as stepping), some CF responses are inhibited, except at the very end of movement, where certain perturbations are likely to occur (e.g. due to misanticipation of the height of the step) [2].

In the model, CFs deliver error information to the cerebellum's APGs (there is exactly one CF for each APG). The CF behavior draws in two key ways from the behavior observed in cat. First, from the perspective of the cerebellar-generated movement, the corrective movement made by the EC module is seen essentially as passive (or unexpected) [4]. Thus, CF signals only deviate from their background firing rate when corrective movements are initiated. Second, the increase (or decrease) in CF firing probability is a function of the rate of stretch of the muscles. Each CF is maximally responsive to stretches in the muscles that are activated by the corresponding APG. In other words, the corrective movement generated by the EC system is spatially coded in the array of CFs (see Figure 5). Positive errors (movements along the same direction as an APG) result in an increase in the corresponding CF firing probability (PR_k), whereas negative errors (in the opposite direction as the APG) result in a decrease in firing probability.

3.2 Temporal Credit Assignment

The temporal credit assignment problem results from inherent delays in the system. First, there is a significant delay between the time when a PF becomes active and when the effects of its activation may be sensed by the IO. Second, the time between the activation of a PF and the generation of a corrective movement (which gives rise to CF signals) can vary by several hundred milliseconds.

We address these problems through the use of an *eligibility trace* on the synapse [13, 17]. This trace acts as a memory of recent synaptic activity. The memory marks the synapse as being *eligible* for potential update in the event that later training inputs arrive from the IO. In our model, we make use of a second-order eligibility trace ($\bar{e_{ij}}$) [12, 16], whose time-course is shown in Figure 6. This form of eligibility trace has the advantage that the memory of PF activation is temporally blurred over a long period of time. As we will see, this implies that CF error signals do not have to be carefully timed relative to PF activity in order for a useful program to be learned.

Finally, updates to the PF-to-PC synaptic strength require the coincidence of eligibility and corresponding CF input:

$$\Delta w_{ij} = -\alpha \; g(\bar{e}_{ij}) \; (CF_k - CF_{base}),$$

where CF_k is the state of the climbing fiber that projects to APG_k (which contains PC_j), α is the learning rate, and CF_{base} is the background firing rate of the climbing fiber. The binary state of CF_k is se-

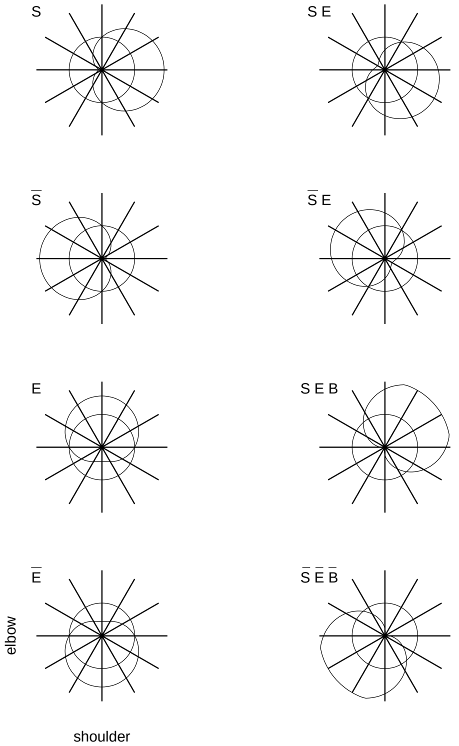

Figure 5: CF responses to movements in joint space. Each sub-plot corresponds to the CF of the indicated APG. Purely horizontal vectors correspond to shoulder movements, while vertical vectors correspond to movements of the elbow. For each possible direction in joint space (all combinations of shoulder and elbow movements), the distance from the origin represents the probability of a CF spike (PR_k). The circles indicate the base-line firing probability. Thus, for the APG that drives the "S" muscle synergy (upper-left panel), the CF responds most heavily to flexion of the shoulder joint (to the right of the origin), and is suppressed below baseline when an extension of the shoulder is sensed. Furthermore, the CF remains at baseline for pure movement of the elbow.

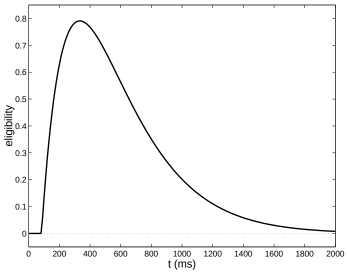

Figure 6: Eligibility trace $\bar{e}_{ij}$ in response to the transient activation of parallel fiber i (at $t = 0$). Unlike a first-order trace (as used in [7]), the synapse does not become eligible for update immediately following a PF event.

lected randomly as a function of its probability of firing (PR_k).

In cases where the corrective movement is aligned with APG_k, PR_k will be high. Thus, on average, the eligible synapses from the PFs to the PCs within APG_k will be decremented. The next time that the same movement is executed, the PCs of APG_k will be less active, resulting in a higher level of activity for APG_k. On the other hand, if the corrective movement is in exactly the opposite direction as APG_k, PR_k will be near 0, the synapse strengths will be incremented, and PC activity will increase, resulting in a decrease in APG activity. In essence, the trajectory that results will be a blend of the original and corrective movements. Over the course of repeated trials, the cerebellar module will ultimately be able to bring the arm to the target, and corrective movements will no longer be necessary.

4 Simulation Results

In this section, the behavior of the model is demonstrated during learning of a single point-to-point movement. Each of the eight APGs consists of eight PCs. The *state encoder* is made up of 120 separate tilings. Each tiling receives input from between 3 and 5 state variables, which are drawn from: θ, θ_g, *error* (i.e $\theta_g - \theta$), $\dot{\theta}$, and $\hat{u}$ (the motor efference copy). A total of 160,000 PFs result from this encoding.

The sensory afference delay to the state encoder (T1 in Figure 1) is 50 ms; motor efference delays (T2) range from 15 to 75 ms; the descending motor command delay (T3) is 50 ms; the sensory afference delay to the EC module (T4) is 50 ms; and inferior olive delay (T5) is 20 ms. A simulation timestep of 5 ms is used.

At the beginning of a trial, the arm is placed in an initial starting configuration (*start* in Figure 7A), and the target location is presented to the controller (*goal* in the Figure). The movement is then triggered, and proceeds under cerebellar control until the module turns off all descending motor commands, and the arm comes to rest. If the arm has stopped moving at a position that is not near the target, the EC system is triggered, which initiates a corrective arm movement. The trial terminates if the arm reaches the target position or the EC system has generated 9 bursts without successfully reaching the target. A new trial is then

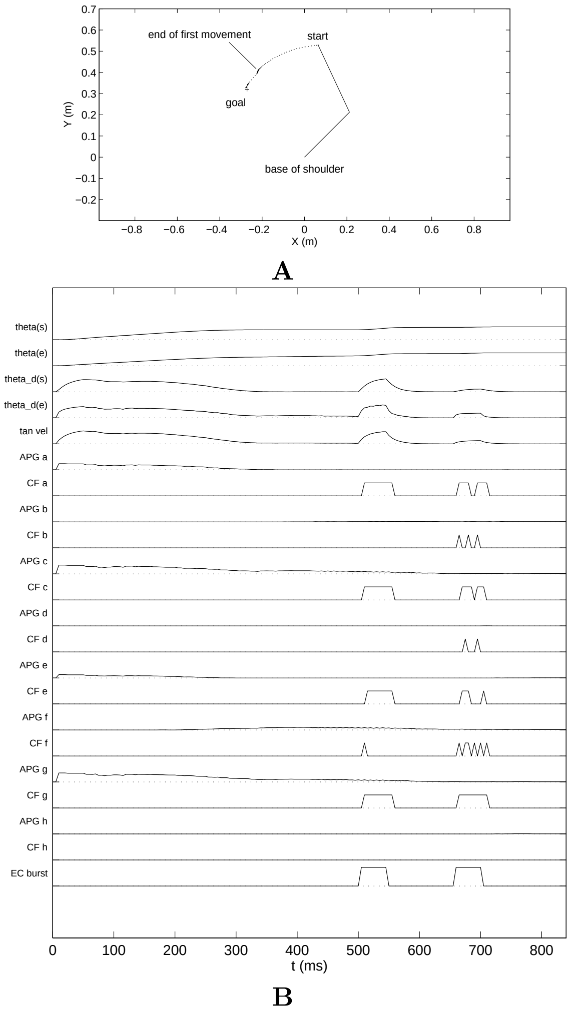

Figure 7: Plant/neural behavior on trial 8. **A** *Movement of the arm in Cartesian space. The initial and target positions are indicated, as is end-point of the first (cerebellar-generated) movement.* **B** *Traces of kinematic and internal variables.* **theta(s)** *and* **theta(e)** *correspond to the joint angles of the shoulder and elbow, respectively.* **theta_d(s|e)** *represent the joint velocities.* **tan vel** *is the tangent velocity of the wrist in Cartesian space. For each APG, two traces are shown: the APG activity and the corresponding CF activity. The APGs correspond to the following muscle synergies: S (a), $\bar{S}$(b), E and $\bar{E}$ (c and d), $S\bar{E}$ (e), $\bar{S}E$ (f), SEB (g), and $\bar{S}\bar{E}\bar{B}$ (h).* **EC_burst** *indicates when the EC system generated a corrective movement.*

initiated, with a resetting of the arm position and the selection of a new target position. In the experiment presented below, however, only a single start/target pair is used.

Figure 7 illustrates the behavior of the plant and the cerebellar control module during the eighth learning trial. The movement produced by the cerebellar module has a duration of 450 ms, and brings the arm most of the distance to the target. The primary contributors to this movement are the APGs that activate muscle synergies E and SEB ($APGc$ and $AGPg$ in Figure 7B). In addition, a minor role is played by the APGs that drive S and $S\bar{E}$ ($APGa$ and $APGe$).

Following the cerebellar-generated movement, two small movements are produced by the EC system. In the first of these corrective movements, the EC system recruits the APGs that drive E, S, $S\bar{E}$, and SEB. This is indicated indirectly by the high level of activity of the corresponding climbing fibers (*CFa*, *CFc*, *CFe*, and *CFg*), and by the low activity of the opposing CFs. During learning, these CF signals cause the original cerebellar-generated movement to be blended with the corrective movements. As seen in Figure 8B, this ultimately leads to a significant increase in the initial burst of three APGs (S, E, and SEB).

By trial 28, the cerebellar module has learned to generate an appropriate motor pattern for reaching the target (Figure 8), and no longer requires the aid of the EC module. APGs S, E, and SEB are the primary contributors to this final motor program. Comparing Figures 7A and 8A, note that even though the EC module generates a trajectory made up of two discrete hops, the final trajectory produced by the cerebellar module does not reflect this behavior. This happens because the cerebellum does not simply append corrective movements onto its original movement, but rather blends the EC-generated movement into its own.

The cerebellar controller is capable of learning single target movements within about 20-30 trials of experience. In addition, the cerebellum's ability to store and appropriately generalize over a set of targets has been tested. In these experiments, the start and target positions for a given trial are selected from a uniform distribution over an 80 × 80 degree range of the joint space. After 10,000 learning trials, the cerebellar controller produces movements with an average of 3.5mm error in the endpoint position (average movement distances are 29 cm). Furthermore, the controller hits the target on 98% of the trials (within 1 cm of the goal), and reaches to within 2 cm of the goal on 99% of the trials.

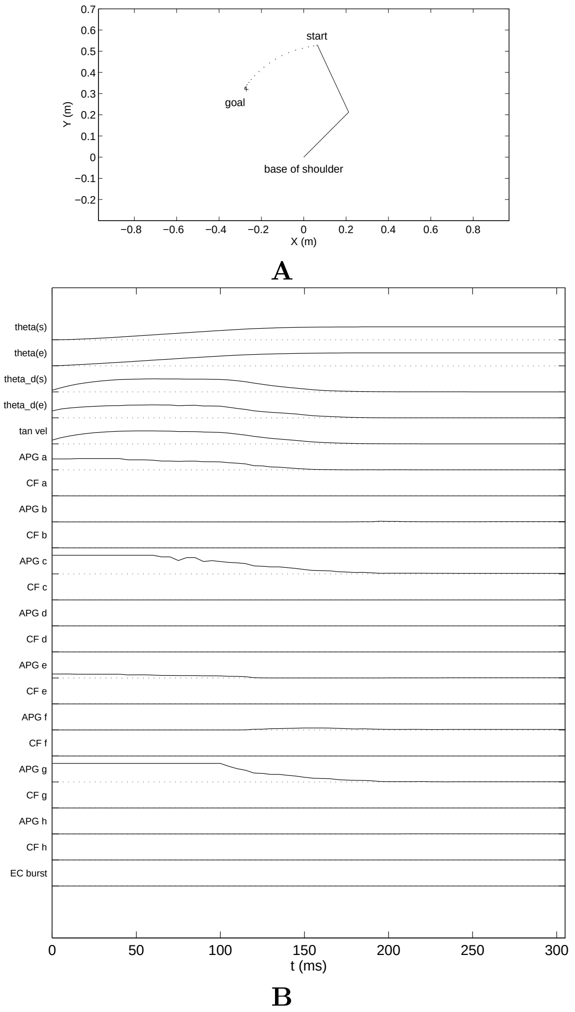

Figure 8: Final plant/neural behavior (trial 28). **A** *The cerebellar control module generates the entire movement to the target (in less than 200ms).* **B** *Trace labels are the same as in Figure 7. APGs S (a), E (c), and SEB (g) contribute the most to producing the movement. Note that no climbing fiber signals are generated in this case.*

5 Conclusions and Future Work

A biologically-inspired model of cerebellar control of a two-DOF planar arm has been presented. In the design of the control system, we have deviated in several ways from more traditional control methods. First, the output of our cerebellar controller is not torque to be applied to a set of joints. Rather, its output falls within a more abstract actuation space. On the engineering side, this allows us to consider multiple actuators affecting individual joints, and multiple joints being driven by a single actuator – without having to rely on a detailed model of these dependencies. From the biological perspective, this approach is also more satisfying. If one assumes that joint torques are the currency of output from the cerebellum, then it becomes necessary to assume that lower-level mechanisms are performing the complex transformation from joint torque space into muscle space. However, it is precisely the structure of the cerebellum that makes it very well suited to learning such a transformation. In taking advantage of this property, it is not necessary to directly represent joint torques at all.

Second, our model does not make use of *reference trajectories*, as has been assumed in other models of cerebellar control [11, 16]. More specifically, there is little physiological evidence to support the idea that another subsystem provides to the cerebellum a detailed kinematic plan – a plan which happens to satisfy certain *dynamic* optimality criteria. Rather, we see such a higher-level system as providing a target position and possibly some constraints about the desired path (e.g. a few via points). The actual trajectory taken by the arm, then, arises from the interaction between the controller and the natural dynamics of the arm.

The learning problems addressed by our cerebellar model fall into two categories: structural and temporal credit assignment. The structural credit assignment problem is solved 1) by marking active PF-to-PC synapses as eligible for future update, and 2) by using a spatially-coded CF activity pattern to select the correct subset of marked synapses. The CF activity pattern is a function of proprioceptive inputs that result from the generation of corrective movements. The error-corrective module (the EC controller) produces low quality corrective movements only when the cerebellum is unable to bring the arm to the target location.

The error-corrective teacher concept is one that we feel is fairly general. Consider a set of reflexes that are designed to respond to unexpected events (e.g. not reaching a goal position, unexpected muscle pulls, collisions with the environment, or slipping of a grasp). By "eavesdropping" on these reflexes (via proprioceptive or other sensory inputs), the IO can provide the cerebellum with a signed error-corrective signal. With this information, the cerebellum can learn to produce motor outputs similar to those generated by the reflexes, but in an anticipatory manner – correcting the movement before the reflex-generating conditions can occur. However, the cerebellum does not simply generate the equivalent of the reflex movement just prior to the reflex-triggering event. Rather, the reflexive motor signal is blended with motor signals that have been generated by the cerebellum over the previous several hundred milliseconds.

In the model, the temporal credit assignment problem results from two factors: 1) the significant hard-wired delay between the time that the cerebellum generates a motor command and the time that its "effects" are seen at the plant; and 2) the variable time delay between the activation of a PF, and the behavioral event that triggers a corrective movement (thus initiating a CF signal). These problems are addressed in this model through the use of a second-order eligibility trace, which maintains a temporally-smeared memory of recently active PF-to-PC synapses.

Finally, the model's ability to rapidly learn an appropriate motor trajectory is the result of the sparse, expansive encoding provided by the granule cell layer (which give rise to the PFs). Such an encoding ensures that two very different movements do not overlap significantly in their representation. This implies that learning for one movement does not tend to interfere with learning for the other.

In continuing work, we are examining how the model may be adapted to produce more human-like trajectories. The model is currently only constrained in its behavior to reach the target location at the end of the movement. Rather, we would like to integrate other forms of feedback, such that the cerebellar module is constrained in the types of trajectories that it learns to generate (this issue has been addressed by others [11, 16]).

References

[1] J. S. Albus. A theory of cerebellar function. *Mathematical Biosciences*, 10:25–61, 1971.

[2] G. Andersson, D. M. Armstrong, S. A. Edgley, and M. Lidierth. Impulse activities of cerebellar neurones during locomotion in the cat. In J. S. King, editor, *New Concepts in Cerebellar Neurobiology*, volume 22, pages 349–370. Alan R. Liss, Inc, New York, NY, 1987.

[3] A. G. Barto, J. T. Buckingham, and J. C. Houk. A predictive switching model of cerebellar movement control. In D. S. Touretsky, M. C. Mozer, and M. E. Hasselmo, editors, *Advances in Neural Information Processing Systems*, volume 8. MIT Press, Cambridge, Massachusetts, 1995.

[4] N. E. Berthier, S. P. Singh, A. G. Barto, and J. C. Houk. Distributed representations of limb motor programs in arrays of adjustable pattern generators. *Journal of Cognitive Neuroscience*, 5:56–78, 1993.

[5] J. R. Bloedel and J. Courville. Cerebellar afferent systems. In J. Brookhart, V. Mountcastle, V. Brooks, and S. Geiger, editors, *Handbook of Physiology, Sect. 1. The Nervous System. Motor Control*. American Physiological Society, Bethesda, MD, 1981.

[6] A. H. Fagg, N. Sitkoff, A. G. Barto, and J. C. Houk. Cerebellar learning for control of a two-link arm in muscle space. In *Proceedings of the IEEE Conference on Robotics and Automation*. Omnipress, 1997.

[7] R. Gellman, A. R. Gibson, and J. C. Houk. Inferior olivary neurons in the awake cat: Detection of contact and passive body displacement. *Journal of Neurophysiology*, 2:185–198, 1985.

[8] J. C. Houk and A. G. Barto. Distributed sensorimotor learning. In G. Stelmach and J. Requin, editors, *Tutorials in Motor Behavior*, volume 2, pages 71–100. Elsevier, Amsterdam, 1992.

[9] J. C. Houk, J. T. Buckingham, and A. G. Barto. Models of the cerebellum and motor learning. *Behavioral and Brain Sciences*, 19:368–383, 1996.

[10] M. Ito. A new physiological concept on cerebellum. *Rev Neurol*, 1990.

[11] M. Katayamma and M. Kawato. Virtual trajectory and stiffness ellipse during multijoint arm movement predicted by neural inverse models. *Biological Cybernetics*, 69:353–362, 1993.

[12] R. E. Kettner, S. Mahamud, H.-C. Leung, N. Sitkoff, J. C. Houk, B. W. Peterson, and A. G. Barto. Prediction of complex two-dimensional trajectories by a cerebellar model of smooth pursuit eye movement. *Journal of Neurophysiology*, 1997.

[13] A. H. Klopf. Brain function and adaptive systems - a heterostatic theory. In *Proceedings of the International Conference on Systems, Man, and Cybernetics*, 1974.

[14] L. E. Miller and J. C. Houk. Motor coordinates in primate red nucleus: Preferential relation to muscle activation versus kinematic variables. *Journal of Physiology - London*, 488(2):533–548, 1995.

[15] W. T. Miller, F. H. Glanz, and L. G. Kraft. CMAC: An associative neural network alternative to backpropagation. In C. Lau, editor, *Neural Networks. Theoretical Foundations and Analysis*, pages 233–240. IEEE Press, New York, NY, 1992.

[16] N. Schweighofer. *Computational Models of the Cerebellum in the Adaptive Control of Movements*. PhD thesis, University of Southern California, 1995.

[17] R. S. Sutton and A. G. Barto. Time-derivative models of pavlovian reinforcement. In M. Gabriel and J. Moore, editors, *Learning and Computational Neuroscience: Foundations of Adaptive Networks*, pages 497–537. MIT Press, 1990.

[18] W. T. Thach, H. Goodkin, and J. Keating. The cerebellum and the adaptive coordination of movement. *Annual Review of Neuroscience*, pages 403–442, 1992.