

Control of a Simulated Arm using a Novel Combination of Cerebellar Learning Mechanisms

Christopher Assad*, Mitra Hartmann*, Michael G. Paulin†

*Jet Propulsion Laboratory, California Institute of Technology, MS 303-300, 4800 Oak Grove Dr., Pasadena, CA, USA, 91109

† Department of Zoology and Centre for Neuroscience, University of Otago, New Zealand
Chris.Assad@jpl.nasa.gov, Mitra.Hartmann@jpl.nasa.gov, mike.paulin@stonebow.otago.ac.nz

Keywords: Cerebellum, Cerebellar learning, Dynamic state estimation, Sensorimotor control

Abstract

We present a model of cerebellar cortex that combines two types of learning: feedforward predictive association based on local Hebbian-type learning between granule cell ascending branch and parallel fiber inputs, and reinforcement learning with feedback error correction based on climbing fiber activity. The model is motivated by recent physiological and anatomical evidence and has more computational capacity than previous functional models of cerebellum. To demonstrate the model's utility, we simulated the control of a simple virtual arm. The model successfully learned to control the timing of release for the arm during a target-throwing task.

Introduction

We present a model of cerebellar cortex that performs dynamic state estimation and prediction. This function is similar to that described by Keeler [14] for Marr-Albus-Kanerva type cerebellar networks [16, 1, 13]. However, the learning mechanisms have been modified to reflect more recent neurophysiological and anatomical evidence. The model acts as a sparse distributed associative memory: information from sensory, proprioceptive and cortical inputs (such as motor commands) provides the context from which to learn to predict future states. Cerebellar output can then be used for several purposes: to adaptively filter future input, to improve detection of novel or unexpected events, to modulate motor outputs, or to provide feedback for motor learning.

Our goal is not to faithfully replicate the cerebellum but rather to build a model inspired by its circuitry and function. The model incorporates two types of learning potentially supported by the cerebellar cortex. Purkinje cells (PC) have long been known to receive inputs from two major sources: parallel fibers (PF) and climbing fibers (CF) (Fig. 1). Simultaneous activation of PF and CF input has been shown to cause long-term depression (LTD) at the PF-PC synapse [12]. Recently, however, evidence has been growing for a third major input to PCs [10] – granule cell axons make multiple synapses onto their overlying PCs as they ascend through the PC layer to the molecular layer (gray in Fig. 1). We have proposed that this ascending branch (AB) input induces postsynaptic activity that causes facilitation when coupled with local PF input [2]; i.e., AB-PF correlations lead to Hebbian learning including long-term potentiation (LTP) at the PC-PF synapse. This allows the learning in cerebellar cortex to be much more flexible through a combination of LTP and LTD. The model presented here combines AB-PF LTP for feedforward state prediction with CF LTD for feedback error correction and reinforcement learning.

We tested the model in a computer simulation to perform dynamic state estimation on a virtual dynamic arm, with the goal to learn to throw a ball at a target. Given a particular arm trajectory, the critical variable for an accurate throw is the timing of release [11]. Over multiple trials the model learned to track and predict the arm trajectory, to modulate the trajectory for different target heights, and to release the ball at an appropriate point within the trajectory.

Methods

The cerebellar model network and virtual dynamic arm were simulated in MATLAB (The Mathworks, Natick, MA) on a Power Mac G4.

Cerebellar Model

The cerebellar model consisted of 6 PC neurons with 4280 PF, 6 AB and 6 CF inputs (Fig. 2). All units were adaptive threshold spiking neurons, adapted from the model neurons described in detail by Nelson and Paulin [17]. 3680 of the PF inputs were excitatory, representing GC activity that sampled from the state variables on the mossy fibers. The receptive field of each GC was chosen as a radial basis function of two of the state variables shown in Figure 2. The receptive fields overlapped and covered the input space uniformly. The remaining 600 PF inputs were inhibitory, representing stellate cells present in the molecular layer with coarser receptive fields. Other inhibitory interneurons (Golgi and basket cells) were not modeled. Each PC received one AB input representing the summed input of many granule cells carrying information about the particular input variable to be learned. Correlations between the AB input and PF inputs caused Hebbian-like learning at the PF-PC synapse, resulting in feedforward predictive association during the throwing trial. After the trial ended, the throw result was evaluated and a binary error signal returned on the CF input for feedback error correction to the PC assigned to learn the release time. If the CF indicated an error, then all training for that trial was forgotten.

Arm Model

To test our learning methods, we simulated the dynamics of a single link arm (i.e., a catapult) given the task of throwing a ball at a target. The arm was modeled using the Denavit-Hartenberg representation of multi-link manipulators [7], with functions from a public domain Robotics Toolbox for MATLAB [6] to implement forward and inverse kinematics and dynamics. The virtual arm was 0.5 m long and had a mass of 1 kg. The desired arm trajectory (shown in Fig. 3A) and an appropriate torque function were given by equations of the form:

$$\theta = 90 + 90 \tanh(\alpha t)$$

$$\tau = mL^2 \ddot{\theta} / 3 + mgL \cos(\theta) / 2$$

where m is the mass of the arm, L the arm length, and time t ranges from -250 to 250 msec. The slope α of the arm angle function allows the torque gain and maximum endpoint velocity to be controlled with a single variable. Here we have neglected the mass of the object to be thrown, but it can be added into the equations with no loss of generality.

Training

Training consisted of multiple trials of swinging the arm through the chosen trajectory while the network performed Hebbian associative learning of spatial and temporal correlations among its inputs. For each trial, the release time was chosen from the spike probability distribution learned over previous trials by the release PC. At the moment of release, the ball was assumed to have the position and velocity of the end effector, and then to follow a ballistic parabolic path influenced only by gravity. A target 0.5 m in diameter was simulated at a distance of 5 m, centered at a height of 3 m (see Fig. 4). The accuracy of the throw was then evaluated. If the throw landed within a window around the target (initially quite large), the learning was accepted and the target window was shrunk for the subsequent trials. If the target window was missed, it was expanded by 10% and the CF for the release command PC provided a simple error signal indicating a miss. This caused the network to forget all learning from each trial with inaccurate results. Thus only learning during accurate throws was permitted to persist.

The throw evaluation also included another heuristic signal that indicated when the throw was too high. After each such trial the torque gain was simply lowered by 5%, both reducing the power and

moving the arm trajectory through a larger good region of state space (see Figure 3B). In this case the motor modulation is simple and does not need to be represented within the cerebellar circuit, but in more complex multi-link systems such changes in gain would also have to be learned.

Results

Because the single link dynamic arm has only one degree of freedom (DOF), the target-throwing problem can be readily visualized in a state space representation for angle and angular velocity. Figure 3B shows the region of state space (light gray) within which a release will result in the object hitting the target. The left heavy line in Fig. 3B indicates the original trajectory of the arm through this space. When training first starts, the release time is chosen randomly from a uniform distribution over the course of this trajectory, and so the throw results are randomly dispersed (Fig. 4A). As training progressed, good throws (ones which hit the target) caused learning that modified the distribution of release time probabilities for the ensuing trials, gradually concentrating the releases closer to the good region of state space (Fig. 4C,D). The final distribution of PF synaptic weights for the release PC (Fig. 3D) results in synaptic currents (Fig. 3C) into the PC that concentrate its output firing during the window of good states for this trajectory (Figs. 3E, 4B). As discussed in Methods, the arm trajectory was modified after several throws that went higher than the target, resulting in a lower torque gain so that the final trajectory passed through the widest extent of the good region of state space (right heavy line in Fig. 3B). This makes the throw less dependent on precise timing, (i.e. the throw accuracy is more robust), because the window of good release times increased from less than 4 msec to 16 msec. However, the changes in trajectory also slowed the overall learning because the network had to relearn the new trajectory whenever it shifted.

Discussion

This modeling effort was a proof-of-concept demonstration of the utility of combining the two types of learning inspired by the cerebellar circuitry. Our model has not yet been optimized for learning conditions or speed, and was applied to a relatively simple system. The real power of the cerebellum model may only be realized if the model is efficiently scaled up to higher DOF problems. We hope to build on these types of models to help explain cerebellar function and for use in robotic applications.

Cerebellar learning

In 1969 David Marr published his revolutionary theory of cerebellar cortex [16], combining cerebellar physiology and anatomy with the machine learning methods of his day. Marr's foremost prediction was that PF synapses onto PCs would undergo Hebbian facilitation when presynaptic PF activity was coincident with postsynaptic PC depolarization induced by CF input. Subsequent physiological experiments, however, showed that simultaneous stimulation of PF and CF inputs causes LTD but not LTP [12]. Alternative Marr-like models therefore used LTD to confer learning capability to cerebellar cortex [1] and by 1989, LTD was widely regarded as the "memory element for cerebellar motor learning" [12].

We have proposed [2] that the AB input, ignored in most previous models of cerebellar function, can play the role envisioned by Marr. AB inputs may induce postsynaptic activity in the PC dendrite, that, when correlated with local PF input, leads to Hebbian learning or LTP at the PC-PF synapse. Llinas [15] first suggested that ABs might make functionally significant contacts with overlying PC dendrites. Bower and Woolston [5] then demonstrated that the ABs of neighboring granule cells provide a synchronous (and therefore relatively large) excitatory input to their overlying PC. Recent anatomical studies [9, 10] of the AB pathway indicate that AB-PC synapses are morphologically distinct from the PF-PC synapses. We believe that PF inputs adjacent to AB inputs are well situated to modulate or gate the AB response, creating ideal conditions for Hebbian-like facilitation or LTP. Neurophysiological

evidence for PF-PC LTP has been reported in studies of synaptic plasticity in cerebellar slice preparations (e.g., see [8]). Finally, learning to predict sensory input has already been demonstrated in cerebellar-like structures in fish [4]. In a similar manner the cerebellar circuitry can function as the associative memory Marr postulated, by learning patterns of sensory and motor inputs presented by the mossy fiber pathway as they are projected onto the AB and PF inputs. Notably, this is very complementary to the LTD result since the CF pathway is now free to assume other roles, including the representation of error signals. By assigning the AB and PFs to feedforward state prediction, and CF input to feedback error correction or change of states, our model can both account for more of the experimental evidence and has increased learning capacity compared to models which ignore the AB input pathway. As described in [2], this new learning hypothesis can be tested with *in vivo* experiments.

Dynamic state estimation and control

Given a mechanical system with particular dynamics and motor command inputs, three functions are useful to learn: (1) to estimate and predict trajectories through state space, i.e., modeling the system dynamics, (2) to learn the “good” region of state space in which an action decision (in our example, the release time) results in a desired goal (e.g., hitting the target), and (3) to modulate motor commands to redirect the trajectory to increase or optimize its intersection with the “good” regions of the state space. The cerebellum has been implicated in all three functions. Here we have been motivated by release timing in human throwing experiments (11), although this may be more aptly described as learning to release at the appropriate *state* during the trajectory, not the time. The set of good states can be reached along many different trajectories. In particular, variability and non-idealities such as noise in the sensed feedback or in actuator output will vary the timing over multiple trials. We are now investigating the model’s response to noise and multiple trajectories, as well as scaling up the system to higher dimensional multiple-link arms. The cerebellar circuit is given the task to generate the best estimate of the dynamic state variables by combining the current measurements and the implicit learned model of system dynamics, and to learn appropriate regions of the space to initiate actions.

Acknowledgements

The research described in this paper was performed at the Center for Integrated Space Microsystems, Jet Propulsion Laboratory, California Institute of Technology, and was sponsored by the National Aeronautics and Space Administration.

References

- [1] J. Albus, A theory of cerebellar function, *Mathematical Biosci.* 10 (1971) 25-61.
- [2] C. Assad, An hypothesis for a novel learning mechanism in cerebellar cortex, *Autonomous Robots* 11 (2001) 285-290.
- [3] C.C. Bell, P. Cordo, S. Harnad, eds., *Controversies in neuroscience IV: Motor learning and synaptic plasticity in the cerebellum*, *Behav. Brain Sciences* 19 (1996).
- [4] C.C. Bell, D. Bodznick, J. Montgomery, J. Bastian, The generation and subtraction of sensory expectations within cerebellum-like structures, *Brain Behav. Evol.* 50 suppl. 1 (1997) 17-31.
- [5] J.M. Bower, D. Woolston, Congruence of spatial organization of tactile projections to granule cell and Purkinje cell layers of the cerebellar hemispheres of the albino rat: vertical organization of cerebellar cortex, *J. Neurophys.* 49 (1983) 745-766.
- [6] P. I. Corke, *Robotics Toolbox for MATLAB*, <http://www.cat.csiro.au/cmst/staff/pic/robot>
- [7] J.J. Craig, *Introduction to Robotics: Mechanics and Control* (Addison-Wesley, Reading, MA, 1986).
- [8] F. Crepel, D. Jaillard, Pairing of pre- and postsynaptic activities in cerebellar Purkinje cells induces long-term changes in synaptic efficacy *in vitro*, *J. Physiol.* 432(1991) 123-141.

[9] G. Gundappa-Sulur, J.M. Bower, Differences in ultramorphology and dendritic termination sites of synapses associated with the ascending and parallel fiber segments of granule cell axons in the cerebellar cortex of the albino rat, Soc. Neurosci. Abs., Vol. 16 (1990) 371.3.

[10] G. Gundappa-Sulur, E. DeSchutter, J.M. Bower, Ascending granule cell axon: An important component of cerebellar cortical circuitry, J. Comp. Neurol. 408 (1999) 580-596.

[11] J. Hore, S. Watts, J. Martin, B. Miller, Timing of finger opening and ball release in fast and accurate overarm throws, Exp. Brain Res. 103 (1995) 277-286.

[12] M. Ito, Long-term depression, Ann. Rev. Neurosci. 12 (1989) 85-102.

[13] P. Kanerva, Sparse Distributed Memory (MIT Press, Cambridge, 1988).

[14] J.D. Keeler, A dynamical system view of cerebellar function, Physica D 42 (1990) 396-410.

[15] R. Llinas, Radial connectivity in the cerebellar cortex: a novel view regarding the functional organization of the molecular layer, in: The Cerebellum, New Vistas (Springer, 1982).

[16] D. Marr, A theory of cerebellar cortex, J. Physiol. 202 (1969) 437-470.

[17] M.E. Nelson, M.G. Paulin MG, Neural simulations of adaptive refference suppression in the elasmobranch electrosensory system, J. Comp. Physiol. A 177 (1995) 723-736.

Figures

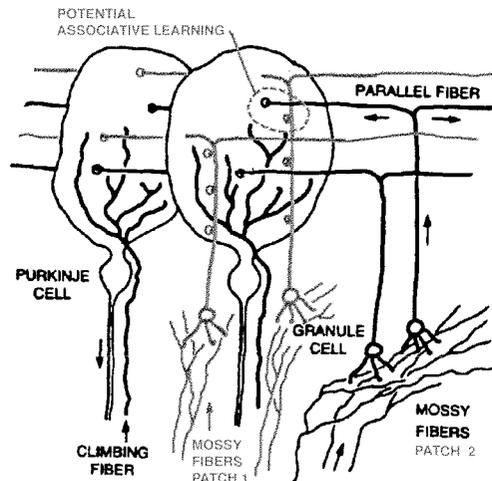


Figure 1. The cerebellar functional microcircuit, adapted from [13] with the addition of ascending branch inputs from the granule cell axons (gray).

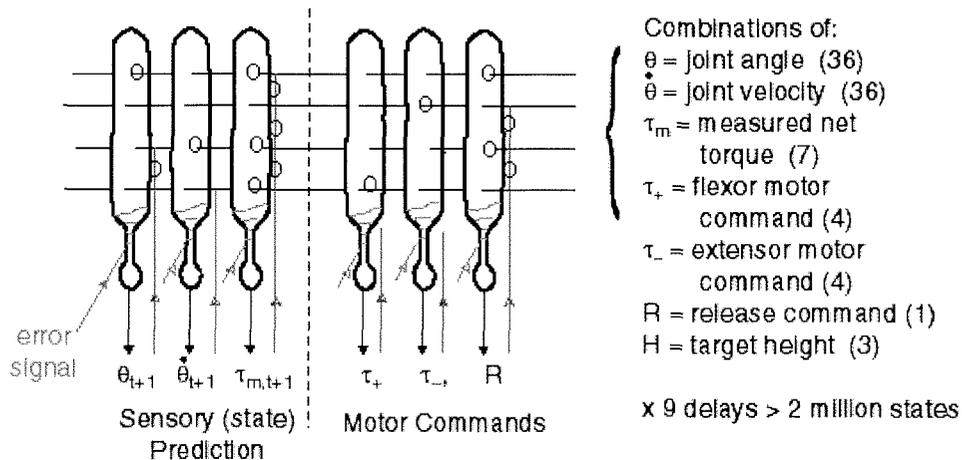


Figure 2. Schematic of the model network.

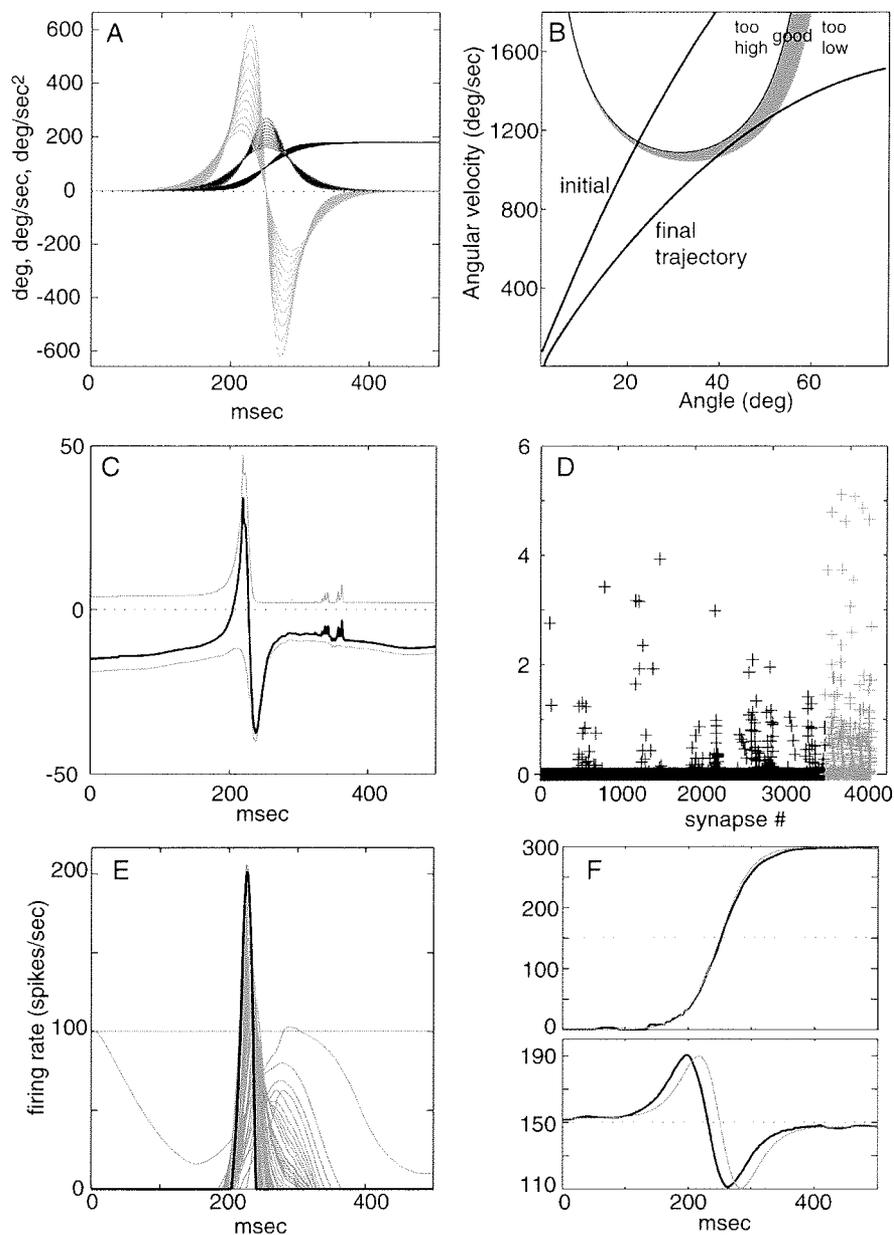


Figure 3. **A.** Arm trajectory variables: angle (black), angular velocity/10 (dark gray) and acceleration/100 (light gray). The torque gain was lowered whenever the throw result was higher than the target (10 times over 1000 trials). **B.** State space representation of throwing task for the single link arm shown in Fig. 4A and B. The gray region indicates where a release would result in a throw hitting the target. The lines indicate initial and final arm trajectories from before and after training. **C.** Net synaptic currents into the release PC after training. **D.** Relative distribution of synaptic weights after training, for excitatory (black) and inhibitory (gray) inputs. **E.** Output firing rate of the release PC over a throwing trial, during training (gray) and after training (black). **F.** Other PCs learned to estimate state variables and predict the torque function necessary to generate the trajectory.

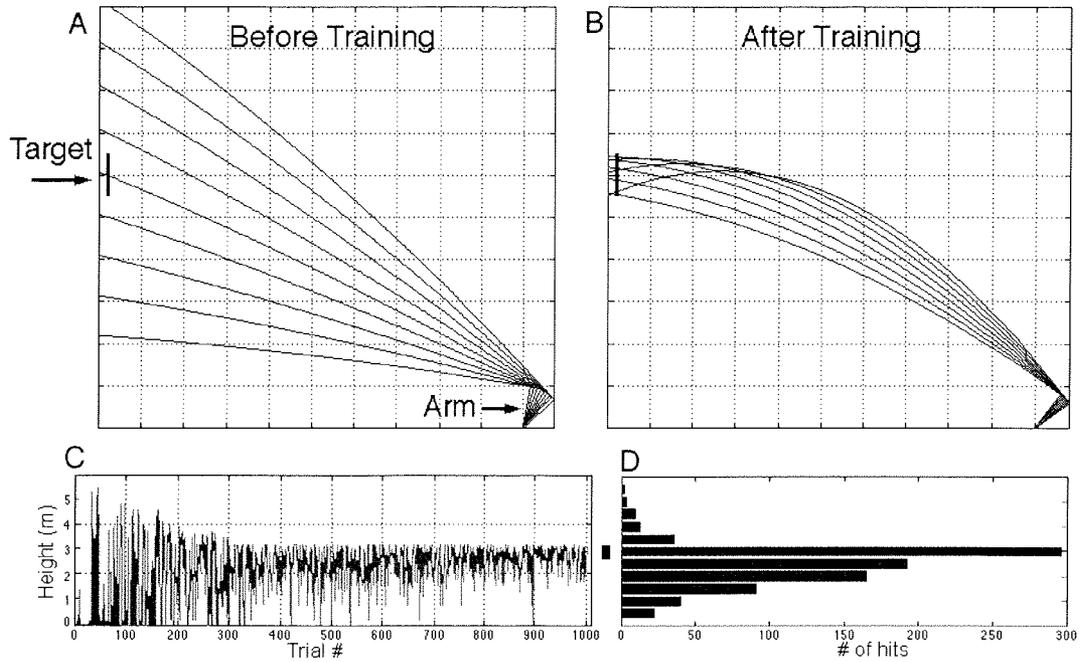


Figure 4. **A.** Throws at 2 msec intervals before training. The random release time results in throws uniformly distributed (only those near target plotted here). **B.** Throws at 2 msec intervals after training. The throws are concentrated nearer to the target and there is a wider window of good release times. **C.** Height of the object when it hits the wall versus trial number. **D.** Histogram of same data in C.