Using Crude Corrective Movements to Learn Accurate Motor Programs for Reaching

Andrew H. Fagg
Department of Computer Science
University of Massachusetts
Amherst, MA 01002

Leo Zelevinsky
Department of Computer Science
University of Massachusetts
Amherst, MA 01002

Andrew G. Barto
Department of Computer Science
University of Massachusetts
Amherst, MA 01002

James C. Houk
Department of Physiology
Northwestern University
School of Medicine
Chicago, IL 60611

Abstract

A computational model that uses crude corrective movements to learn accurate reaching programs is presented. The model learns a feed-forward motor program in the form of pulse-step commands which drive a dynamic, 2 DOF, planar arm. The arm is actuated by two pairs of opposing muscles, which drive the shoulder and elbow joints. The model is able to learn motor programs that accurately bring the arm to the target, while producing in many cases bell-shaped tangential velocity profiles.

1 Introduction

Young infants reaching to touch or grasp an object make movements that differ greatly from adult grasping movements. Instead of tangential velocity profiles that are dominated by a single peak and are approximately bell-shaped, infant profiles consist of a sequence of peaks, which define segments of movement that are referred to as "movement units" (von Hofsten, 1979). As the infant develops, the number of movement units required to reach a target is reduced, until adult-like velocity profiles are ultimately achieved. Berthier (1994) suggested that some of the individual peaks result from the discrete triggering of a movement-generating mechanism, and that the individual movement units are attempts at correcting errors in the previous movement. Berthier, Singh, Barto and Houk (1993) hypothesized in a model of cerebellar learning that similar corrective movements, though crude, may serve as a source of training information for an adaptive process that tunes a motor program.

In this paper, we focus on the issue of using crude corrective movements to learn accurate motor programs for controlling a simulated dynamic arm. The arm is actuated by two pairs of opposing muscles, which drive the shoulder and elbow joints. For the purposes of the experiment described in this paper, the learning controller represents a simplistic motor program for moving the arm from a single initial position to a single target, and takes the form of pulse-step motor commands that are sent to the muscles. The pulse-step commands are tuned as a function of the corrective movements that follow the learned movement. Despite the crude nature of the corrective information, and the simple representation of the motor program, the system learns to generate accurate reaching movements that exhibit, in some cases, straight paths and bell-shaped velocity profiles.

2 A Simple Control System

The control system consists of two separate modules which generate motor signals in the form of activation levels for each of the four muscles (Figure 1). The *learning controller* stores a feed-forward motor program that attempts (with learning) to bring the arm to the target. The hard-wired *corrector module* is responsible for generating crude corrective movements when the learning controller does not produce accurate reaches. The output control signals of these two modules are summed, and then passed through a cascade of three first-order, low-pass filters before arriving at the muscles.

The muscles are modeled as visco-elastic (spring) elements. A single muscle control signal sets the threshold for the muscle's length-tension curve. We assume a linear dependence of tension on length (Houk and Rymer, 1981). The dependence of tension on velocity is described by the Hill equation for shortening (Winters and Stark, 1987), and is dominated by the stretch reflex during lengthening. The non-linear damping provided by the stretch reflex is approximated by $-\dot{l}^{1/5}$, where $\dot{l}$ is the velocity of muscle stretch (Gielen and Houk, 1984). The passive forces are modeled as exponentials, the magnitudes of which are only significant near the joint limit, as in Winters and Stark (1985).

For an agonist/antagonist pair of muscles, the control signals specify a unique equilibrium point for the joint. However, due to the high, non-linear damping forces generated by the stretch reflex, there exists a region around the equilibrium in which the joint velocity is rapidly forced towards zero, effectively causing the joint to "stick". A more detailed treatment of related muscle models may be found in (Wu et al., 1990; Fagg et al., 1997; Barto et al., 1998). For simplicity, the muscle moment arms are assumed to be constant over the range of movement.

The learning controller represents a motor program in the form of a muscle activation pattern (the *pulse*), which is followed at a specified time by a second pattern of activation (the *step*). [1] The details of these pulse-step programs are collapsed in the learning controller into a set of two parameters for each joint: the heights of the pulse and step (the activation pattern for each flexor/extensor muscle pair is specified by a single parameter). For simplicity, we focus here on the case of learning to reach from a single initial point to a single target. Therefore, only four parameters are represented by the learning controller. The time of transition from

[1] Pulse-step control of rapid eye movements has been well established (Robinson, 1975); this form of control appears to be a reasonable abstraction in the context of arm movements (Ghez and Martin, 1982).

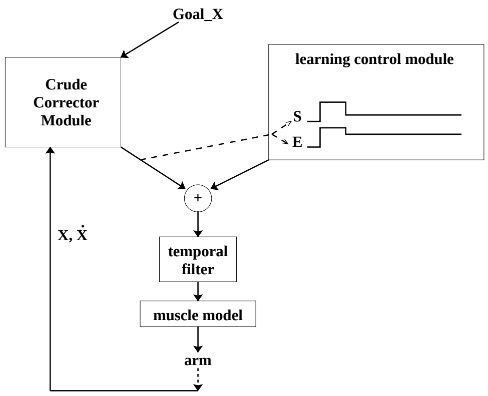

Figure 1: The Control Architecture. Both the *crude corrector* and *learning* modules produce muscle activation signals. These two sets of signals are summed, the result of which is passed through a temporal filter, before affecting the state of the muscles. The learning module produces pulse-step activation patterns that are intended to bring the arm from a given state to a target. The corrector module generates pulses of activation when the arm comes to rest at a point far from the target.

the pulse to step is linearly scaled with the maximum of the shoulder and elbow pulse heights. When appropriately tuned, the pulse phase serves to initiate the movement of the arm towards the target. The step phase sets the threshold of the stretch reflex that is responsible for slowing the arm, and thereafter maintaining the arm at the target (Ghez and Martin, 1982).

In addition to the learning controller, our model assumes the existence of a pre-wired corrective control module that is capable of making crude movements in the approximate direction of the target. This module produces a corrective movement only in cases in which the arm stops moving before the target is reached. The movement is generated using fixed-duration, constant-magnitude bursts of muscle activity. For a given correction, the burst magnitudes for the shoulder and elbow agonists are chosen according to three heuristics: 1) Inspired by the observations made by Karst and Hasan (1991), the relative activation level of the shoulder and elbow muscles is derived from the target's position in a visual coordinate system that is rooted at the wrist, and whose orientation is that of the forearm (Figure 2). Target locations along the axis of the forearm are translated into pure shoulder movements, whereas those located along the perpendicular axis result in pure elbow movements. Off-axis targets recruit a mixture of shoulder and elbow muscles; 2) The magnitude of the muscle bursting pattern is scaled with distance to the target; and 3) If the arm has not moved significantly since the last correction, then the magnitude of the muscle bursting pattern is increased by a constant factor.

In many (but not all) instances, the movements generated by this set of heuristics bring the arm closer to the target. However, the target is rarely reached on the first correction. Instead, a sequence of corrections is typically required to complete a given reach.

3 The Learning Algorithm

The protocol for training is as follows: At the beginning of a trial, the arm is placed in a starting configuration, and a target location is specified. The reaching movement is then triggered – the learning controller executes its pulse-step motor program. If the arm stops moving at a position not near the target, then the corrector module produces a brief pulse of activation. If the arm again stops far from the target, then the correction process is repeated until the wrist is within 1 cm of the target.

Not only do the pulses generated by the corrector module take the arm closer to the target, but they also provide information about how the learning module's pulse-

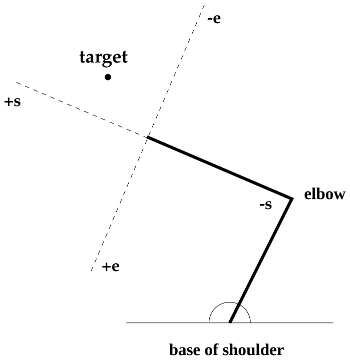

Figure 2: Heuristic method for choosing the relative magnitudes of the elbow and shoulder flexor/extensor muscle bursts. Targets located in the half-plane denoted by $+e$ recruit elbow flexors; the $-e$ half-plane recruits elbow extensors; $+s$: shoulder flexors, and $-s$: shoulder extensors.

step motor program should be updated. If the step levels are miscalibrated, then even if the pulse is able to bring the arm near the target, the arm will subsequently drift away. When the arm stops moving, this will result in the production of a corrective movement, whose direction will indicate how the learning controller's *step command* should be updated.

In contrast, if the pulse command is drastically miscalibrated (but the step is correct), the position of the arm shortly after the execution of the learning controller's pulse-step program will be far from the target, resulting in a sequence of corrections. In this case, the training information derived from the corrections is used to update the *pulse command* of the learning controller.

The full learning procedure, in which the pulse and step are learned in parallel, is summarized below. L_p and L_s are vectors representing the pulse and step commands of the learning controller, respectively; $C(X, T)$ represents the corrector motor signal, which is a function of the arm (X) and target (T) positions.

FOR each trial
    Set the target and initial arm positions: T and $X(0)$, respectively
    Execute the pulse command: L_p; Wait for pulse duration
    Execute the step command: L_s; Wait for arm to stop
    UNTIL arm is at the target
        Execute the corrective pulse: $L_s + C(X, T)$
        IF the arm has been near the target during this trial
            $L_s \leftarrow L_s + \alpha C(X, T)$
        ELSE
            $L_s \leftarrow L_s + \hat{\alpha} C(X, T)$
            $L_p \leftarrow L_p + \beta(i) C(X, T)$
        Wait for arm to stop

α, $\hat{\alpha}$, and $\beta(i)$ are step-size parameters in which $\hat{\alpha} \ll \alpha$, and $\beta(i)$ decays exponentially with the number of corrections, i, made since the beginning of the trial.

In the above algorithm, the corrective pulse is taken to be the sum of the learning controller's step (L_s), and the pulse generated by the corrector module ($C(X, T)$). Without the learning controller's step, it would be necessary for the corrector module to also take into account the absolute position of the target in order to produce movements in the correct direction, thus adding to the "intelligence" that would have to be hard-wired into this module.

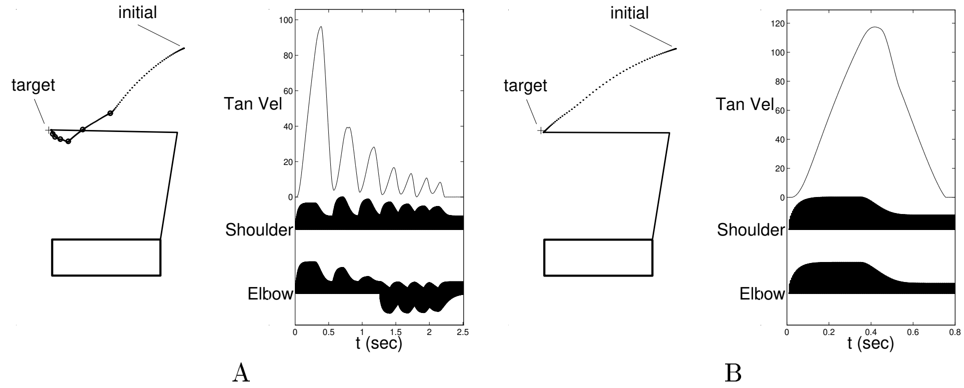

Figure 3: System behavior at trial 11 (A) and trial 35 (B) during learning for a single initial position/target pair. The small dots indicate the path taken by the wrist; larger dots show points at which the corrective movements were generated. The temporal plot shows the tangential velocity of the wrist (cm/sec), and the filtered motor commands for the shoulder flexor (extending upward), extensor (downward), and the elbow flexor (up) and extensor (down).

4 A Learning Experiment

The behavior of the system while learning to move from a single initial position to a single target is illustrated in Figure 3. After 10 learning trials (Figure 3A), the pulse signals for the shoulder and elbow flexors that are generated by the learning controller are not large enough to bring the arm to the target. This results in a sequence of six corrective movements that make smaller and smaller "hops" towards the target. The corrector module recruits the shoulder flexor for all six corrections (indicated by six additional peaks in the flexor motor command). For the elbow, however, the first two corrections recruit the flexor, but the remaining corrections involve the extensor. We also see flexor activity during these latter corrections; this is due to the fact that the learning controller's step phase recruits the elbow flexor (recall that the learning controller's step is combined with the corrector's pulse in order to generate a corrective movement).

After 34 trials of learning (Figure 3B), the wrist makes a relatively straight-line movement to the target, and the velocity profile is approximately bell-shaped. Furthermore, the learned movement does not require the invocation of the corrector module.

Figure 4 demonstrates the paths learned by the system for 3 initial positions and 47 target positions. [2] For each initial position/target pair, the pulse-step commands of the learning controller were first cleared; learning trials were then presented to the system until the movement was completed with no corrections. For these movements, an average of 33 trials were required for each training pair (the range was 16-98 trials). The hooks that occurred at the end of many of the movements resulted from one joint completing its motion before the other. This happened typically when the magnitudes of movement in joint space differed significantly between the two joints. Also, in these cases, the tangential velocity profile showed a high velocity peak, followed by either a second (lower) velocity peak or a flat region.

[2] The one path not shown (center starting position of Figure 4B, movement to the right and down) was not learnable by the system due to limitations in the algorithm in dealing with the high passive forces near the joint limit.

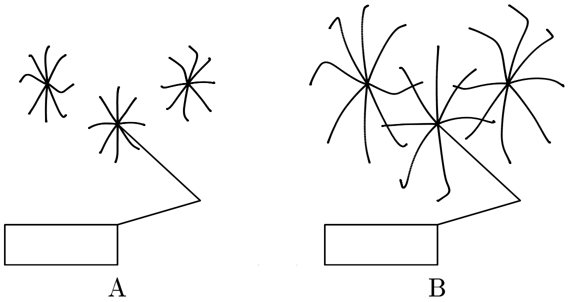

Figure 4: Learning movements for a set of center-out movement tasks of length 10 (A) and 20 (B) cm. Each path is the result of separate learning runs, each consisting of an average of 33 learning trials.

5 Discussion

This paper has attempted to demonstrate that a crude error correction device may be used to drive the adaptation of a high-quality motor program. Our model is able to learn feed-forward motor programs for reaching that accurately bring the arm to the target, while producing in some cases tangential velocity profiles that are single-peaked and bell-shaped. The trajectories generated by our model that do not reflect these adult-like properties end with hooked paths, which are the result of one joint completing the movement before the other.

Hollerbach and Atkeson (1987) have suggested that by staggering the onset of movement of one joint relative to the other, curved movements may be made to appear straighter. Several authors have reported experiments which suggest that such a strategy may be utilized by humans (Kaminski and Gentile, 1986; Karst and Hasan, 1991b). Karniel and Inbar (1997) have proposed a model similar to ours that makes use of pulse-step control to drive a 2 degree-of-freedom arm. In their model, the motor programs are described not only by the height of the pulse and step for each joint, but also by the relative timing of the joint pulses. By using these additional degrees of freedom, they are able to learn movements that do not reflect the hooks that are seen in our model. Their learning procedure, however, relies on a systematic variation of the control parameters to estimate an error gradient, which is then used to further reduce movement errors.

Our approach is also related to Kawato's feedback error learning, in that training is driven by movement produced by a controller external to the learning system (Kawato and Gomi, 1992). However, our model differs from this work in two key ways. First, feedback in our model is not provided continuously, rather it is available sporadically. Second, our model does not require the use of a high-quality reference trajectory from which the corrective movements are computed.

Although what is learned in the present model is a pure feedforward motor program for one initial position/target pair, the learning approach is more generally applicable at two levels. First, it would be possible to learn to compute the pulse-step parameters as a function of the current state of the arm and target position, allowing the model to learn reaching movements over the entire workspace. Second, in the context of a more realistic cerebellar learning model, it is possible to learn to combine contextual information, motor efference copy, and delayed sensory inputs in order to compute, in an on-line fashion, the outgoing motor commands (Fagg et al., 1997; Barto et al., 1998).

Acknowledgments

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

References

Barto, A. G., Fagg, A. H., Houk, J. C., and Sitkoff, N. (1998). A cerebellar model of timing and prediction in the control of reaching. *Submitted.*

Berthier, N. E. (1994). Infant reaching strategies: theoretical considerations. *Infant Behavior & Development*, 17:521.

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

Fagg, A. H., Sitkoff, N., Barto, A. G., and Houk, J. C. (1997). A model of cerebellar learning for control of arm movements using muscle synergies. In *Proceedings of the IEEE Symposium on Computational Intelligence in Robotics and Automation*, pages 6–12.

Ghez, C. and Martin, J. M. (1982). The control of rapid limb movment in the cat. III. Agonist-antagonist coupling. *Experimental Brain Research*, 45:115–125.

Gielen, C. C. A. M. and Houk, J. C. (1984). Nonlinear viscosity of human wrist. *Jornal of Neurophysiology*, 52:553–569.

Hollerbach, J. M. and Atkeson, C. G. (1987). Deducing planning variables from experimental arm trajectories: Ptifalls and possibilities. *Biological Cybernetics*, 56:67–77.

Houk, J. C. and Rymer, W. Z. (1981). Neural control of muscle length and tension. In Brooks, V. B., editor, *Handbook of Physiology: Sec. 1: Vol. 2. Motor Control*, pages 247–323. American Physiological Society, Bethesda, MD.

Kaminski, T. and Gentile, A. M. (1986). Joint control strategies and hand trajectories in multijoint pointing movements. *Journal of Motor Behavior*, 18(3):261–278.

Karniel, A. and Inbar, G. F. (1997). A model of learning human reaching-movements. *Biological Cybernetics*, to appear.

Karst, G. M. and Hasan, Z. (1991a). Initiation rules for planar, two-joint arm movements: Agonist selection for movements throughout the work space. *Journal of Neurophysiology*, 66(5):1579–1593.

Karst, G. M. and Hasan, Z. (1991b). Timing and magnitude of electormyographic activity for two-joint arm movements in different directions. *Journal of Neurophysiology*, 66(5):1594–1604.

Kawato, M. and Gomi, H. (1992). A computational model of four regions of the cerebellum based on feedback–error learning. *Biological Cybernetics*, 68:95–103.

Robinson, D. A. (1975). Oculomotor control signals. In Lennerstrand, G. and Bach-y-rita, P., editors, *Basic Mechanisms of Ocular Motility and their Clinical Implications*, pages 337–374. Pergamon Press, Oxford.

von Hofsten, C. (1979). Development of visually directed reaching: The approach phase. *Journal of Human Movement Studies*, 5:160–168.

Winters, J. and Stark, L. (1985). Analysis of fundamental human movement patterns through the use of in-depth antagonistic muscle models. *IEEE Transactions on Biomedical Engineering*, BME-32(10):826–839.

Winters, J. and Stark, L. (1987). Muscle models: What is gained and what is lost by varying model complexity. *Biological Cybernetics*, 55:403–420.

Wu, C. H., Houk, J. C., Young, K. Y., and Miller, L. E. (1990). Nonlinear damping of limb motion. In Winters, J. and Woo, S., editors, *Multiple Muscle Systems: Biomechanics and Movement Organization*. Springer–Verlag, New York.