

P-121

**Acquisition and Improvement of  
Human Motor Skills: Learning  
Through Observation and Practice**

**WAYNE IBA**

**AI Research Branch, Mail Stop 269-2  
NASA Ames Research Center, Moffett Field, CA 94035**

(NASA-TM-107878) ACQUISITION AND  
IMPROVEMENT OF HUMAN MOTOR SKILLS: LEARNING  
THROUGH OBSERVATION AND PRACTICE (NASA)  
121 p

N92-29174

Unclas  
G3/53 0091509

**NASA Ames Research Center**  
**Artificial Intelligence Research Branch**

**Technical Report RIA-91-29**

**October, 1991**



# REPORT DOCUMENTATION PAGE

OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE Dates attached	3. REPORT TYPE AND DATES COVERED	
4. TITLE AND SUBTITLE  Titles/Authors - Attached		5. FUNDING NUMBERS	
6. AUTHOR(S)		8. PERFORMING ORGANIZATION REPORT NUMBER  Attached	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)  Code FIA - Artificial Intelligence Research Branch Information Sciences Division		10. SPONSORING / MONITORING AGENCY REPORT NUMBER	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)  Nasa/Ames Research Center Moffett Field, CA. 94035-1000		11. SUPPLEMENTARY NOTES	
12a. DISTRIBUTION / AVAILABILITY STATEMENT  Available for Public Distribution  <i>Peter Friedland</i> 5/14/92 BRANCH CHIEF		12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words)  Abstracts ATTACHED			
14. SUBJECT TERMS			15. NUMBER OF PAGES
17. SECURITY CLASSIFICATION OF REPORT			16. PRICE CODE
18. SECURITY CLASSIFICATION OF THIS PAGE			20. LIMITATION OF ABSTRACT
19. SECURITY CLASSIFICATION OF ABSTRACT			21. LIMITATION OF ABSTRACT



# Acquisition and Improvement of Human Motor Skills: Learning Through Observation and Practice

WAYNE IBA\*

AI Research Branch, Mail Stop 269-2  
NASA Ames Research Center, Moffett Field, CA 94035

## Abstract

Skilled movement is an integral part of the human existence. This is exemplified in a range of behaviors from concert violin performance, to picking up and drinking a glass of milk. A better understanding of motor skills and their development is a prerequisite to the construction of truly flexible intelligent agents. Existing computational models have mostly focused on low-level issues of controlling manipulators rather than on capturing skilled movements as conceptual units. The psychological literature provides very high-level abstract theories or low-level analysis of specific movement phenomena. Furthermore, the acquisition of skills is largely ignored in both bodies of work. In response to these issues, we present *MEANDER*, a computational model of human motor behavior, that uniformly addresses both the acquisition of skills through observation and the improvement of skills through practice.

*MEANDER* consists of a sensory-effector interface, a memory of movements, and a set of performance and learning mechanisms that let it recognize and generate motor skills. The system initially acquires such skills by observing movements performed by another agent and constructing a concept hierarchy. Observed movements are parsed and stored internally as motor schemas. Two subsystems of *MEANDER* interact to allow observed movements to be recognized and stored skills to be executed. The *OXBOW* module is responsible for constructing and modifying the skill hierarchy according to the observed experiences. Given a stored motor skill in memory, the *MAGGIE* component will take the motor schema and cause some effector to behavior appropriately. Errors in execution can be corrected through a closed-loop feedback control mechanism. All learning involves changing the hierarchical memory of skill concepts to more closely correspond to either observed experience or to desired behaviors.

One can evaluate the effectiveness of a model in a number of ways. We evaluate *MEANDER* empirically with respect to how well it acquires and improves both artificial movement types and handwritten script letters from the alphabet. We also evaluate *MEANDER* as a psychological model by comparing its behavior to robust phenomena in humans and by considering the richness of the predictions it makes.

\* Wayne Iba is affiliated with RECOM Technologies.



## Acknowledgements

Several people made significant intellectual contributions to this work. Pat Langley guided this research and helped me over numerous obstacles. He also taught me how to do good science and to write clearly. Jim Wogulis, John Gennari, and John Allen each contributed significantly through lengthy discussions of the work. Glenn Iba was responsible for getting me interested in machine learning.

I am also indebted to a number of institutions for their financial support. This work was initiated and largely developed while attending the University of California at Irvine, and was completed at the Artificial Intelligence Research Branch of the NASA Ames Research Center. I am grateful to both the Department of Information and Computer Science at UC Irvine and the AI branch at NASA Ames for the use of their extensive computational resources. I also thank the Joseph J. Fischer Foundation, Hughes Aircraft Company, and the Army Research Institute for grants that helped support my research. I refuse to mention the institutions to which I am indebted at a usurious level.

I also gratefully acknowledge the prayers and support of Frank Horne and Brian Ingrassia, two of my closest friends. Mostly, I am grateful to Fran, my very best friend and loving spouse, for putting up with me and this work. This wouldn't have been done without her. But lest I leave anyone out, I acknowledge every living soul on the planet; without your presence it just wouldn't have been the same.





# Contents

<b>Chapter 1</b>	<b>Context for the Dissertation</b>	<b>1</b>
1.1	Motivating a Study of Motor Learning . . . . .	1
1.2	Goals of the Research . . . . .	2
1.3	Evaluation of the Research . . . . .	3
1.4	Outline of the Dissertation . . . . .	3
<b>Chapter 2</b>	<b>A Review of Human Motor Behavior: Phenomena, Theories, and Models</b>	<b>5</b>
2.1	Introduction . . . . .	5
2.2	Phenomena of Human Motor Control . . . . .	6
2.3	Psychological Theories of Motor Control and Learning . . . . .	12
2.4	Computational Approaches to Motor Behavior . . . . .	21
2.5	Conclusions . . . . .	26
<b>Chapter 3</b>	<b>A Computational Theory of Motor Learning: An Overview of the Mæander System</b>	<b>27</b>
3.1	Introduction . . . . .	27
3.2	Refining the Task Specifications . . . . .	28
3.3	Mæander's World View . . . . .	29
3.4	The Structure of Mæander . . . . .	33
<b>Chapter 4</b>	<b>Learning to Recognize Observed Movements</b>	<b>35</b>
4.1	Introduction . . . . .	35
4.2	Representation and Data Structures in Oxbow . . . . .	36
4.3	Recognizing a Movement with Oxbow . . . . .	42
4.4	Learning from Unsupervised Experience . . . . .	45
4.5	Conclusion . . . . .	50
<b>Chapter 5</b>	<b>Skill Improvement Through Practice</b>	<b>51</b>
5.1	Introduction . . . . .	51
5.2	Representations for Generating Behavior . . . . .	52
5.3	Executing Motor Skills in MAGGIE . . . . .	54
5.4	Learning from Execution Errors . . . . .	58
5.5	Discussion . . . . .	62
<b>Chapter 6</b>	<b>Evaluating Movement Recognition in Mæander</b>	<b>65</b>
6.1	The Experimental Method . . . . .	65
6.2	Learning Single Movement Concepts . . . . .	67
6.3	Concept Formation for Multiple Movements . . . . .	73
6.4	Predicting Unseen Movement . . . . .	75
6.5	Recognizing Handwritten Letters . . . . .	79
6.6	Conclusions . . . . .	82



<b>Chapter 7</b>	<b>Evaluating Movement Generation in Mæander</b>	<b>85</b>
7.1	Introduction . . . . .	85
7.2	Behavior of the Performance System . . . . .	86
7.3	Behavior of the Learning System . . . . .	92
7.4	Generating Script Letters . . . . .	97
7.5	Conclusions . . . . .	99
<b>Chapter 8</b>	<b>Discussion</b>	<b>101</b>
8.1	Introduction . . . . .	101
8.2	Contributions of Mæander . . . . .	101
8.3	Limitations of the Model . . . . .	103
8.4	Future Work . . . . .	104
8.5	Closing . . . . .	107
<b>References</b>		<b>109</b>



## Context for the Dissertation

---

### 1.1 Motivating a Study of Motor Learning

The ability to manipulate objects in the environment is one of the intrinsic features that demonstrates intelligence, and human intelligence is distinguished from that of most other species by the sophisticated level of such manipulation (Rosenbaum, 1991). Learning is an especially important issue to any model of motor behavior, as evidenced by the difficulties encountered in constructing flexible and powerful robotic mechanisms. When considering human motor behavior, the significance of learning becomes even more apparent in the contrast between the breadth and proficiency of an adult's motor skills and that of a child.

Until recently, the topic of motor skills has been largely ignored within the machine learning community. We are encouraged by the recent interest demonstrated by efforts aimed at learning sequences of operators that can control effectors external to the learning agent (e.g. Laird, Hucka, Yager, & Tuck, 1990; Mason, Christiansen, & Mitchell, 1989; Moore, 1990). However, it is not clear that these methods can describe the kinds of complex movements involved in skills such as dance, Tai Chi Chaung, or violin playing. Furthermore, human learning involves both acquiring skills through observation and improving them through practice. A comprehensive model of motor behavior should address both of these issues.

There are two reasons to study *human* motor skills. A better understanding of the mechanisms involved in motor behavior may facilitate improved treatments for certain physical disorders. Also, a good model of skilled behavior in humans will help identify important issues and processes in the design of an artificial movement systems. Such a computational model will contribute greatly towards developing an intelligent agent that interacts with a complex environment.

Similarly, there are two reasons to study learning. As already mentioned, learning is an integral process in human behavior. But learning also addresses the knowledge acquisition "bottleneck". That is, appropriate domain knowledge is an integral part of intelligent behavior, and encoding that knowledge can be time consuming. Learning through observation is one way to simplify the knowledge encoding process.

## 1.2 Goals of the Research

Our purpose in pursuing the research described in this dissertation has been to develop a computational model of human motor behavior. That is, we want to construct and test a system that exhibits *skilled performance*, where this refers specifically to motions involving jointed manipulators. Secondly, and wherever possible, we want our model to be patterned after our knowledge of human constraints, performance, and learning. A complete model of human motor behavior is beyond our grasp and we must accept reasonable limitations on what we accomplish. Four characteristics identify the specific scope of our work.

The first characteristic our model should exhibit, mentioned briefly above, is the ability to both recognize and generate movements. We view much of intelligent behavior as a two-step process involving understanding and expression. For a given task, humans frequently acquire an initial level of skill through observation, and then refine their abilities through practice performing the task. Likewise, our model should acquire a knowledge base of movement skills by recognizing observed actions performed by some other agent. Given such a knowledge base, the model should be able to generate its own movements and improve these movements through practice.

We also intend our model to address movements that are concerned with the trajectories of limbs, as in dance or handwriting. This is in contrast to *aiming* tasks, which address moving an arm to a desired position (Fitts & Peterson, 1964). Likewise, this class of skills is distinct from *maintenance* tasks, such as driving a car or balancing a pole (Michie & Chambers, 1968; Selfridge, Sutton, & Barto, 1985; Sutton, 1984). We recognize the importance of these other tasks and do not suppose that the class we address subsumes them. Rather, we assume the presence of many low-level mechanisms that each contribute to a total understanding of motor skills, only one of which we consider here.

A third characteristic of our desired model is that its scope should include a wide range of movement complexities within the class of skills. That is, the representation, organization, and learning of movement skills should be flexible enough to handle both the simplest of movements and very complex ones. This is necessary to establish the flexibility and applicability of the model.

Finally, we desire that the model's behavior in recognition and execution correspond to that of humans for similar tasks. Computational models that address psychological phenomena have often proved insightful both to artificial intelligence and psychology. There are many well-documented phenomena in human motor behavior that have been identified and numerous models to explain them. We view these as constraints on the design and behavior of any psychologically plausible model. An ideal model should, within a single framework, account for a large portion of the phenomena that have been identified.

In summary, we want a computational model of skilled motor learning that addresses both the acquisition of skills through observation and the improvement through practice. The types and complexities of skills that the model handles should be as broad as possible, and its structure and behavior should be compatible with knowledge of human motor skills and learning. This particular conjunction of characteristics requires us to attend to and draw upon ideas from the fields of artificial intelligence, machine learning, and cognitive science. We want to pull together a number

of issues, problems, and techniques that have never been framed together before. We hope to connect high-level planning and low-level motor control by creating a model of skills that operates somewhere between the level of abstractions at which each work. That is, we want to provide a bridge between the “pick-up” and “move-to” operators common in planning and the very low-level control mechanisms necessary to move a real arm. We hope that both machine learning and psychologists can benefit from an intermediate model somewhere in between the two corresponding fields. We expect different aspects of the resulting model to make contributions to both fields.

### 1.3 Evaluation of the Research

Later in this dissertation we present a computational model that addresses the above characteristics. A natural question to consider for any such model is *how well* it satisfies the purposes for which it is intended. Langley (1987) outlines general types of evaluation – empirical, theoretical, and psychological – that would be applicable to any theory or computational model. In this work, we empirically evaluate our model as a machine learning system and compare its behavior, both quantitatively and qualitatively, to behavior observed in humans.

Empirical evaluation attempts to demonstrate the utility of the model’s representations, performance methods, and learning mechanisms. Kibler and Langley (1988) have outlined numerous approaches to empirically evaluating a machine learning systems. Although we utilize a number of their ideas, we emphasize the modest scope of our experiments. We argue that the conjunction of goals described above is unique and that, at this stage, it is sufficient to demonstrate the feasibility of our particular computational model.

Psychological evaluation involves comparing some aspect of an artificial model to what is known about humans. This can be done in a number of ways. One can compare the gross characteristics of the model’s design and assumptions to human physiology. Additionally, one can either qualitatively or quantitatively compare behavioral characteristics of the model and the human. In order to establish our model as psychologically plausible, we employ all of these approaches to evaluation.

### 1.4 Outline of the Dissertation

The characteristics presented as the goals of this research amount to a design specification for a computational model of human motor learning. In the remainder of this dissertation we proceed to develop and test such a model. We call this model *MEANDER*, and show that it satisfies, to varying degrees, the above characteristics.

In the next chapter we review a number of the psychological phenomena that our model should exhibit. We also look at several psychological theories of human motor behavior to determine if we could transform one of these into a computational model. Finally, we consider previous computational models to see if any could be extended or modified to satisfy our current goals. We conclude that none of the theories or existing computational models are satisfactory for our design specifications.

In light of these findings, in Chapters 3, 4, and 5 we present MÆANDER, together with its requirements, assumptions, and organization. Chapter 3 presents the contextual environment in which MÆANDER was developed and tested, as well as the assumptions of the model. Chapter 4 describes the details of OXBOW, our model of memory management. This chapter includes a description of the mechanisms that recognize observed movements and acquire movement concepts through observation. Chapter 5 presents the details of MAGGIE, a system that embodies our ideas on movement generation and modification mechanisms.

We empirically evaluate MÆANDER in the following two chapters. In Chapter 6 we consider OXBOW's ability to recognize movements as a function of observations. Then in Chapter 7 we evaluate MÆANDER's ability to generate movements and improve the quality of generated movements through practice. Here we also consider MÆANDER's behavior with respect to several aspects of human performance and learning.

We close the dissertation with Chapter 8, which reviews both the contributions embodied in MÆANDER and the areas in which the model was found wanting. In closing, we also discuss potential responses to these weaknesses, thus suggesting directions for continuing this line of work.



## A Review of Human Motor Behavior: Phenomena, Theories, and Models

---

### 2.1 Introduction

Motor skills play an essential role in human behavior. The modifications that humans make to their environment reflect high-level thought processes and planning, but the basic means available for such manipulations come through the use of our arms and hands. Note that many mammals are able to walk or run within minutes of birth, whereas humans generally require a year of development before taking their first tottering steps. Because learning plays such an important part in human motor behavior, we are interested not only in how humans control their limbs in interesting and skillful ways, but also in how such abilities are acquired through observation and practice.

Researchers must address both planning and control issues in order to gain a greater understanding of how humans interact and manipulate their world and how they acquire this ability. This involves understanding a variety of issues, including high-level thought processes, cognitive development, and muscular control. We would like to find a computational theory that cuts across all of these areas.

The study of limbed movement is called *kinesiology* or, more simply, *human motor behavior*. This field is largely a synthesis of muscular physiology and experimental psychology. Historically, the earliest notions on the subject were proposed by the fathers of modern psychology (e.g., James). When behaviorism became popular, interest in motor behavior died, as all actions were thought to be explained by stimulus-response theory. During World War II, interest in motor control was renewed in an attempt to understand the performance requirements for tasks of interest to the military. This stage was largely influenced by cybernetics and control theory due to the feedback-driven nature of radar tracking and gunnery tasks. More recently, researchers have focused on developing process-oriented theories that account for a range of phenomena pertaining to the control of limbs. Since then, more experimental work attempts to validate and falsify the predictions and explanations made by the various theories that have been proposed.

In this chapter, we identify connections between theories of human motor behavior, and the design and control of artificial manipulator systems. Furthermore, we want a computational model that incorporates both motor issues and cognitive issues. However before beginning on this goal, we must decide how to recognize a good theory when we have found one. We start by considering a number of the phenomena that have been identified from research on human motor control. In the next section, we describe the nature of these phenomena, the empirical evidence upon which they are based, and their respective implications for theories of human motor control. In Section 2.3 we focus on psychological theories of motor control, presenting three theories of human motor skills. We rate each based upon their ability to explain and account for the phenomena and according to their suitability for computational implementation. Of course, complete coverage of the phenomena is not imperative, and we are looking for a semi-formal means of comparison. In Section 2.4, we consider systems for controlling artificial limbs. We consider these systems with respect to their adequacy as models of human motor learning. In the closing section, we evaluate the psychological theories and computational models with respect to our original goal – a computational theory of human motor learning dealing with complex behaviors. We conclude that the theories surveyed in this chapter provide insights along various dimensions, but that none are satisfactory for our stated goals in Chapter 1. In the following chapters we proceed to present our computational model designed with these specifications in mind.

## 2.2 Phenomena of Human Motor Control

Science attempts to explain and predict phenomena. These phenomena are regularities in events that, given similar situations, can be repeatedly observed. For the purposes of this chapter, we will focus on phenomena that have already been identified rather than on predictions made by theories of motor control.

Learning always occurs in the context of some performance task, so we will also examine performance aspects of human motor control. We will consider these issues separately, first reviewing the performance phenomena and then the learning phenomena. We will concentrate on robust regularities that have been repeatedly observed. We are concerned mostly with *whether* a given theory or model accounts for a particular phenomenon, and not as much with *how* such an explanation is made. In each subsection, we will focus on describing the phenomena and the experiments associated with them, delaying discussion of explanations until the next section.

### 2.2.1 Performance Phenomena

The first two phenomena that we will consider reflect performance issues in the execution of motor skills. These are exhibited during the course of movements and do not depend upon any improvement in performance quality over time. That is, these phenomena are observable at any stage of learning to varying degrees of influence.

## THE SPEED-ACCURACY TRADEOFF

Perhaps the most well documented phenomenon in the study of human motor behavior is the *speed-accuracy tradeoff*. This is the seemingly obvious regularity that, the faster a particular skill is attempted, the more difficult it is to perform the skill accurately. Although others discussed this phenomenon even earlier, Fitts (1954, 1964) was possibly the first to rigorously examine, study, and report the phenomenon. His careful studies led to the formulation of a relation, known as *Fitts' law*, that captures the maxim "haste makes waste" with quantitative values. This law relates the movement time ( $MT$ ) to the index of difficulty ( $ID$ ),

$$MT = a + bID \quad . \quad (1)$$

That is, if the constants  $a$  and  $b$  are known (for a particular set of time and distance units), then the  $MT$  of the arm for a task with a particular  $ID$  can be predicted.

Fitts (1964) motivated the index of difficulty using information theory, defining it with the equation

$$ID = \log_2 \frac{2A}{W} \quad . \quad (2)$$

This amounts to the ratio of the movement amplitude ( $A$ ) to the target width ( $W$ ). Now let us examine how this is demonstrated and observed in movements in the laboratory.

Fitts and Peterson (1964) manipulated two independent variables in a discrete motor task: the distance or amplitude to be moved and the width of the target to be touched. Subjects were required to make rapid aimed movements to one of a pair of targets; the appropriate target was indicated by a stimulus light. The targets were replaceable with variable widths and at different distances from the starting button. The subjects would hold a stylus on the starting button and move the stylus to the appropriate target as rapidly as possible. Fitts and Peterson reported several slight variations on this procedure, but the results were essentially identical and the results conformed to the predictions made by Fitts' law.

In an alternative methodology, Schmidt, Zelaznik, Hawkins, Frank, and Quinn (1979) used a *time-matching* task to test this tradeoff. In this case, the subject is required to enact a movement to a target at a fixed distance  $D$ , but must match the duration of the movement to a target time  $T$ . This temporally constrained task (Meyer, Abrams, Kornblum, Wright, & Smith, 1988) yields a quite different tradeoff relation. Schmidt et al.'s (1979) results conform to the equation

$$S = a + b \frac{D}{T} \quad ,$$

where  $S$  is the standard deviation of the movement endpoints in space (variable error),  $D$  is the mean movement distance, and  $T$  is the mean movement duration. If we think of the variable error as an effective target width, then this relation describes movement time as a linear tradeoff in distance and target width. That is, we can rewrite this relation as:

$$T = b \frac{D}{S - a} \quad ,$$

where  $S$  corresponds to the target width  $W$  in equation 2.

Apart from the quantitative differences, these two relations qualitatively capture the complementary nature of distance and precision. Each applies in particular tasks but all tasks exhibit the general qualitative effect of decreased accuracy with increased speed. Of the phenomena discussed in this chapter, the speed-accuracy tradeoff is especially well documented. Many other studies have shown that Fitts' law generalizes to other types of movements, including ones using joints other than the shoulder and elbow. Langolf, Chaffin, and Foulke (1976) have demonstrated that movements of the finger, wrist, and arm all conform to Fitts' law, but that the constants differ from one set of joints to another. That is, the wrist is more accurate than the arm and the fingers are more accurate than the wrist. These results are for finger movements of around  $\frac{1}{10}$  inch in length and wrist movements of  $\frac{1}{2}$  inch in length performed under the magnification of a microscope. Thus, no matter what the task, a model of human motor behavior should reflect this robust tradeoff.

#### INTER-LIMB SIMILARITIES FOR SKILLS

The other performance phenomenon that we will consider involves the similarities observed when a skill is performed on different limbs. This can be thought of as transfer of skill between limbs.<sup>1</sup> More specifically, characteristics of skills learned with one limb are evident when the same skill is performed by another limb. This result suggests a single underlying representation for a given movement skill.

For example, consider a comparison of samples from someone's handwriting or signature with various limbs, such as the dominant hand, opposite hand, foot, and mouth. This is a well-known demonstration, and the comparison is usually done qualitatively by simply looking at the handwriting samples and noting common characteristics (Raibert, 1976). Figure 2.1 shows several samples of handwriting generated by a single subject using different limbs.

There is additional evidence for corresponding characteristics for movements executed on different limbs in Rosenbaum's (1977) study of fatigue in the *rotor task*. His experiment examined two basic conditions. Rosenbaum had subjects either crank a handle in a circular motion as rapidly as possible for 30 seconds, or twisted a handle back and forth for 30 seconds. With minimal interruption, the subjects were then required to crank or twist (a  $2 \times 2$  factorial design) with the *other* hand as rapidly as possible. The dependent measure of interest was the speed of cranking or twisting with the second hand. The results indicated that fatigue from one task transferred to the same task but not to the other task.

Both the qualitative results in the handwriting comparison and the quantitative results from the fatigue study support the notion of a uniform underlying representation for motor skills. Although the transfer of skills between limbs is not as well documented as the speed-accuracy tradeoff, these two phenomena provide a starting place from which to compare models of motor control along performance dimensions. Next we consider several learning phenomena in turn.

---

1. One should not confuse this phenomenon with the more widely studied issue of transfer of learning between *tasks* (see Schmidt, 1975a).

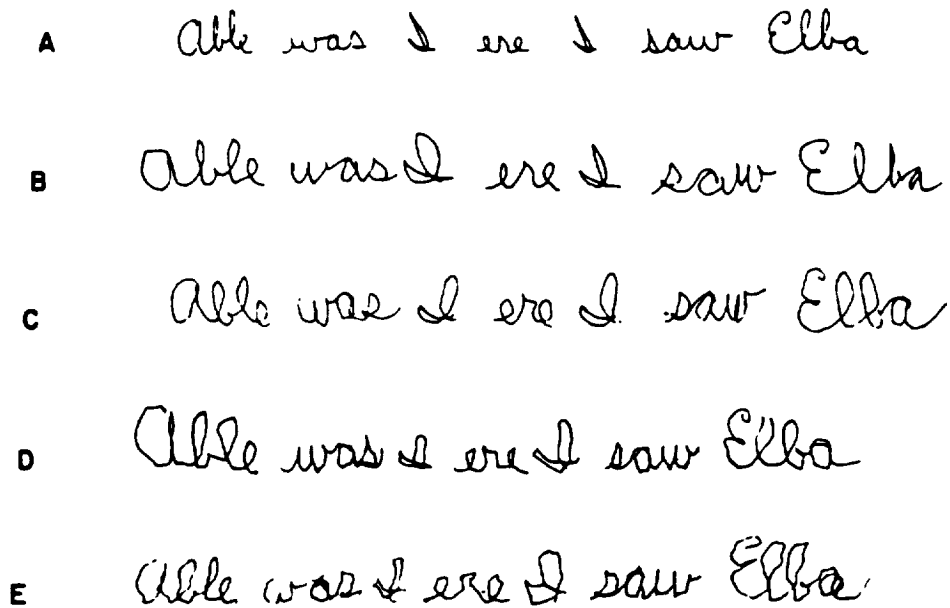


Figure 2.1. Five samples of handwriting from the same person using the right hand (A), right arm (B), left hand (C), mouth (D), and right foot (E), taken from Raibert (1976).

### 2.2.2 Learning Phenomena

Learning is demonstrated through the improvement in performance of a particular task. Often, improvement comes as a result of experience or practice. The phenomena we consider here relate to factors that influence the rate of such gains in performance, or describe the conditions that facilitate improvements. Also, we consider how the attentional overhead associated with performance can change as a result of learning.

#### THE POWER LAW OF PRACTICE

In general, performance appears to improve with practice, but this is not the full story. The type, quality, quantity, and scheduling of practice are all significant factors that influence the degree to which improvements (if any) are gained. In this section we consider a quantitative result that relates the improvement in performance speed to the amount of practice.

This relationship has been known as the *log-log linear learning law* (Snoddy, 1926), as *DeJong's law* (Crossman, 1959), and simply as the *power law of practice* (Newell & Rosenbloom, 1981). All versions of this law make the same claim – that a logarithmic improvement in performance speed requires a logarithmic amount of additional practice. Performance speed is simply the time required to complete a given task. The phenomenon has yet again been referred to as the *law of diminishing returns*, referring to the fact that the practice necessary to improve performance by a given amount increases over time.

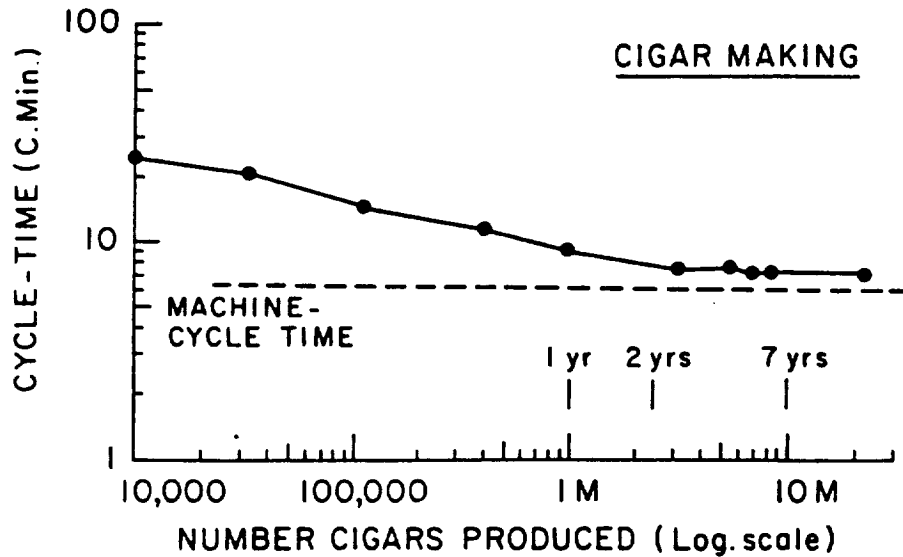


Figure 2.2. Cigar manufacture time as a function of the number of previous cigars manufactured on logarithmic scales (from Crossman, 1959).

This regularity was well documented by Crossman (1959), who studied a number of workers making cigars. The cigars were made on a machine that was operated by the workers in the study. Over a period of seven years, data were collected for the same workers on how fast they were able to make a cigar.

Figure 2.2 shows a graph of the time to make a single cigar as a function of the number of cigars previously made. The results indicate that decreases in the time to make a cigar were achieved only after increasingly greater amounts of practice. That is, the *rate* of improvement declines with increasing practice. When plotted using log scales for the horizontal and vertical axis, the data points describe a straight line up to two years. At two years the operators appear to have stopped improving. Crossman attributed this to the minimum cycle time of the cigar making machines; that is, after two years the operators were producing cigars in the minimum time allowed by the machinery.

Newell and Rosenbloom (1981) present a comprehensive discussion of power laws and how the experimental data fit these theoretical curves. As they point out, it is not clear if the data are better fit by a power law or an exponential curve. They suggest that there may be other learning processes involved that mask the power-law curves. Whether it is a power law or exponential, this quantitative relation has only been demonstrated to hold for speed of performance. We might also expect it to apply to other aspects of performance, such as the amount of error and the need for attention. Although speed and error are related by the speed-accuracy tradeoff discussed above, in these types of learning studies, error is kept constant at a minimum level. Whether this relation also holds for skills such as free-throw accuracy remains to be demonstrated. Next we turn to the need for attention during the performance of a task and how that need changes as a result of practice.

### TRANSFER FROM CLOSED-LOOP TO OPEN-LOOP BEHAVIOR

Considerable attention has been paid to the automation of skills. However, much of the discussion generated around this issue has focused on defining and identifying automation. That is, what does it mean for a skill to become "automatic" and when does such a transition occur? We will consider a trend toward automation to be a reduction in the attentional resources necessary to perform a particular task. Unfortunately, this only pushes the problem back one level. What do we mean by attention and how do we measure it? For our purposes, the amount of attention necessary for a given task is directly related to the amount of interference (in performance) caused by a coincident distraction task.

A common method of exploring this interference has been the use of a secondary reaction time task. That is, during the performance of a main motor task, the subject is required to respond to a probe as quickly as possible. The degree to which the tasks interfere should be reflected in an increased reaction time to the probe. Ells (1969) used just such a design with a main task of moving a pointer to a target as quickly as possible and varying the temporal presentation of the probe. The results indicated that, with practice, subjects reduced their reaction times on the secondary probe task.

Unfortunately, the results from this and other experiments do not tell us clearly what is actually happening with respect to automation and attention. Currently there is considerable debate about the nature of attention and about skills that are said to be "automatic". Other studies have shown that combining two tasks or skills can result in interference, whereas one of the two paired with yet another task will yield no interference. For now, however, our main concern is satisfied by these results. They indicate that when two tasks *do* interfere, practice tends to reduce such interference.

This aspect of the phenomena is also closely associated with what can be called the shift from *closed-loop* to *open-loop* control (Pew, 1966). Closed-loop control implies feedback, error detection, and error correction; a movement performed in open-loop control receives no feedback and is run to completion without opportunity for adjustments. Here, the issue is the presence and use of feedback instead of the availability of attentional resources. But clearly these are closely related in so far as it requires attention to evaluate feedback information and determine what to do to improve the movement. A restatement of our phenomenon then would be that through learning a subject is able to shift motor control from a jerky, feedback-dependent performance to a smooth execution of feedback-free movement.

### PRACTICE VARIABILITY EFFECTS

Most of the phenomena in our list have historically been explored in their own right and then later included and explained in a particular theory of motor learning or control. The practice variability effect is unusual in this respect in that it was predicted by Schmidt's schema theory (1975b).

The prediction can be stated as follows: the more varied the practice, the more accurately a novel but related task will be performed. McCracken and Stelmach (1977) tested this prediction in an experiment requiring subjects to make timed movements of 200 msec. The goal was to reach a barrier marking the end of the movement distance as close to 200 msec. as possible. The length of

the movement was manipulated according to the experimental conditions. There were two training conditions – high variability and low variability. In the high-variability condition, subjects were trained on four different length movements. In the low-variability condition, subjects were trained only on a single length movement. After training, both groups were required to perform a novel movement, where the length had not been previously performed, again in a 200 msec. time period.

The results demonstrated a weak support for the initial prediction – that the high-variability practice group would perform better on the transfer task. Although the low-variability group appeared to have lower errors than the high-variability group on the initial task, the high-variability group had significantly lower errors on the transfer task. Other researchers have demonstrated similar results, and Frohlich and Elliott (1984) have extended these results beyond motor control. They have obtained variable practice effects in operating dynamic systems that are external to the human motor system. Unfortunately, there are also studies that fail to support this phenomenon (Melville, 1976) or that even present contradictory evidence (Zelaznick, 1977). Although some controversy exists around this phenomenon, it is clearly in operation in some circumstances and the question becomes one of qualifying those contexts. Therefore, a good model of human motor control should be able explain the phenomenon in some situations but not others. Now let us turn to some of the psychological motor theories that have been proposed and see whether they account for the phenomena discussed above.

### 2.3 Psychological Theories of Motor Control and Learning

As we have stated, early research on motor behavior was characterized by the identification of phenomena. Of course, this is an important stage of any developing discipline. Ultimately, however, such phenomena must be collected into a coherent theory that explains as many of the known phenomena as possible and makes predictions about new phenomena. As predictions made by one theory are falsified, new theories arise that make the “correct” prediction and additionally make new predictions. Such is the progression of science.

This is precisely what has happened in the field of human motor behavior. Adams (1971) proposed one of the first comprehensive theory of human motor behavior. Concurrently, Pew (1974, 1970) suggested an alternative theory that emphasized different aspects of the complete story. In response to these (and other accounts), Schmidt (1975b) proposed his own theory, which has gained acceptance and has stood the test of time quite well up to the present.

Certainly there were other theoretical results before, during, and after this period, and we are not intending to exclude this work. However, we are considering a theory to be *comprehensive* if it includes at least the following: a reasonably detailed description of the memory structures required, a detailed outline of the modules responsible for the production of motor behavior, and a careful description of the processes involved in acquiring the representations in memory used to generate movement. As an example, in this light Saltzman (1979) would not be considered as comprehensive as those mentioned above. Although he provides an extremely detailed analysis of representation structures, he only alludes to the production and acquisition components. Thus, we will consider



only the theories we have mentioned above and focus on their memory structures, performance mechanisms, and learning processes.

### 2.3.1 Adams' Closed-loop Theory of Motor Learning

The scope of Adams' (1971) theory is intended to include "the instrumental learning of simple, self-paced, graded movements, like drawing a line, even though the implications extend further. And the bounds include only learning by humans old enough to have a verbal capability" (p. 122). As the title of the theory implies, it is a closed-loop, feedback-centered approach. Drawing upon early servo-mechanism ideas, Adams' model resembles the classic closed-loop control mechanism found in control theory.

#### MEMORY STRUCTURES

There are two basic memory structures in Adams' theory – the *perceptual trace* and the *memory trace*. The perceptual trace is memory of previous experience in movements, and the memory trace is the pattern used for generating movements.

The perceptual trace is based upon multiple sources of sensory feedback. Proprioception is a predominant source, but visual and tactual information are also very important. Even auditory feedback can be useful in many situations. For example, the sound of the ball on a bat resulting from a "good" hit is distinctive and will provide cues for predicting the result. Although the perceptual trace is thought of as a single memory structure, Adams (1971, p. 125) states that "in actuality it is a complex distribution of traces." The movement on any given trial creates a trace that contributes to the total distribution of traces. Each individual trace will tend to fade and ultimately be forgotten, but the distribution somehow manages to get stronger, although this process is not explained. The strength of the perceptual trace, thought of as a unit, is an increasing function of the number of trials on which feedback was given. As similar traces are repeated over and over, the mode of the distribution becomes strong and allows a distinctive trace to arise as the means of comparison. The perceptual trace comes to correspond to the sensations associated with the correct end point of a particular movement.

In the context of simple, self-paced movements and feedback control, the extent of a movement is the predominant controlling property. In such movements, feedback plays an integral role, but the feedback must be compared to some standard of reference to determine the correct extent of the movement. The perceptual trace performs this role in Adams' theory.

It might seem that the perceptual trace alone is sufficient for the generation and control of movement; however, there are several problems associated with this position. First, every movement will appear to be correct if it is initiated by the same structure as is used for the reference in a typical closed-loop system. Also, using only the perceptual trace as the reference of correctness requires feedback, which is not available until approximately 200 msec. into the movement. Finally, results from verbal behavior indicate that recall and recognition, or the production and recognition of responses, respectively, are based on two different memory states (Adams & Bray, 1970; Kintsch,

1970). To account for these, Adams includes in his theory another structure called the *memory trace*.

The memory trace is introduced to "select and initiate the response, preceding the use of the perceptual trace" (p. 125). This structure is responsible for controlling a movement once initiated, until sensory feedback can be compared with the perceptual trace. The remainder of the movement is governed by feedback and the perceptual trace. Adams admits that he is uncomfortable with this form of two-state memory, but sees it as the most reasonable choice given the closed-loop assumptions and the nature of the proposed perceptual trace. He contrasts the perceptual trace, which controls the extent of a movement, with the memory trace, which controls the selection of a movement. Here the limiting context of self-paced straight line movements mentioned above is particularly evident, as more complex movements cannot be described by duration or length.

#### PRODUCING AND IMPROVING MOVEMENTS

In Adams' theory, the performance component is quite simplistic, so we will consider both performance and learning issues together. Consider how the memory structures described above are utilized to produce voluntary movements. The production of movements in Adams' theory involves using the perceptual and memory traces in a typical closed-loop feedback control system. The memory trace is the (initial) generator and selects the path to be followed. After the initial delay, feedback becomes available and the perceptual trace comes into action, controlling the remainder of the movement. The perceptual trace is compared with the sensory feedback, and adjustments are made in an effort to reach a zero error end state.

In order to improve performance, one or both of the memory structures used to control movement must somehow be modified. The memory trace is strengthened as a function of knowledge of results and practice. However, Adams claims that this is not the source of significant improvement. Instead, the building and strengthening of the perceptual trace is credited with improvements.

As stated above, the strength of the perceptual trace is a function of the sensory feedback experienced on each trial. Improvements could be gained simply from the drift in the mode of the distribution of sensory traces as a result of more correct sensory experience, but this implies a conscious change in the tendency of the movements. Learning actually occurs when the subject uses the knowledge of results to make the next response be different than the previous one. That is, the perceptual trace is modified and applied with respect to the previous knowledge of results.

Since movement in Adams' theory is explicitly controlled by the perceptual trace, an "average" over many similar experiences, it cannot explain the generation of different movements, except with different traces. This requires a separate trace for every movement ever produced, even when two movements are relatively similar, thereby introducing a massive memory load. Below, we see that Pew (1974) presents a theory that addresses this issue by including a more general memory structure.

### 2.3.2 Pew's Closed-loop Theory

Pew (1974) presents a closed-loop theory of human motor performance that is very similar to Adams' but with a somewhat different flavor. Although the theory is oriented towards performance issues, Pew does outline what would be involved in the acquisition of motor skills within his framework. Most of the attention is focused on performance, leaving representational issues more sketchy than in Adams' theory.

#### MEMORY STRUCTURES

The basic motor memory structure in Pew's theory is the *movement pattern*. This is similar to the concept of a *motor program*, in so far as it is a string of motor commands that can accept parameters to slightly alter the resulting movement along certain dimensions. The movement pattern "may be thought of as a stored representation of a path in space through which the members of the body will move" (Pew, 1974, p. 31). These patterns are stored or collected under the second memory structure – the *schema*. The idea for schema learning is credited to Bartlett (1958) and Posner and Keele (1968), but probably goes much further back than that. However, in Pew's theory, the exact nature of the schema is even more unclear than the movement patterns. "What properties of a movement pattern are encoded? What properties are intrinsic to a particular schema and what properties are only dimensional parameters that are free to vary from one execution to another?" (p. 28) These are all questions that Pew asks but leaves unanswered.

The schema and the *schema instance* (which is nothing more than the movement pattern generated or selected from a given schema) are the necessary memory structures for the generation of movements. But as we saw in Adams' theory, this is not sufficient for the closed-loop control of voluntary movements. Pew posits that the result of selecting a particular movement pattern, the schema instance, is the generation of an image of the sensory consequences experienced when actually executing the movement pattern. The sensory consequences are analogous and perform the same role as the perceptual trace in Adams' theory. It is the image of the sensory consequences that allows the detection and correction of errors in movements while they are in progress.

#### PRODUCING MOVEMENTS

Since both Pew and Adams' present closed-loop theories, the means of movement generation will be very similar, though the memory structures used are different. In Pew's theory, a particular movement pattern is selected from the schema (the generalized source of movement information) according to the stimulating conditions existing in the environment. Of course, the selection process depends upon both the dynamic state of the subject and the environment at the current time. Once the schema instance has been selected, it must be translated into a temporal string of motor commands recognizable by the limb effectors. Pew suggests that at this stage the timing (or speed) information is added to the string of muscle commands. This allows the movement to be speeded up or slowed down as a whole. Schmidt et al. (1985), Schmidt (1982b), and Armstrong (1970) present evidence that practiced movements maintain their temporal relationships independent of

performance speed. This suggests a speed parameter applied to a string of motor commands that stretches and shrinks the entire movement uniformly.

Once the temporal sequence of muscle commands is formulated, all that remains is to execute this program. The muscles are then activated according to this sequence, producing a movement in space and time. However, for various reasons movements do not always proceed exactly as intended. In these cases, one needs some correction mechanism.

One interesting point about Pew's theory is that he stresses multiple levels of feedback and expected consequences. For example, he describes knowledge of results as a high-level feedback and details about the goal to be achieved as high-level expected consequences. At a lower level, the actual sensory consequences received from executing the movement pattern can be compared with the perceptual trace of expected sensory consequences. He lists these two levels as examples of a possible larger set of levels that interact during the performance of movements. Therefore, it is difficult for Pew to explicate the comparison process that results in alterations to the ongoing movement.

However, a unique point in this matter is that, in Pew's opinion, "corrections are executed . . . not on the basis of deviations from a predetermined path but rather on the basis of revised estimates of where the target is with respect to where the subject's hand now is" (p. 25). This implies not only a significantly different comparison and correction mechanism from Adams', but also a more complex one involving the integration of multiple sources of information. Information from the high-level goals, the sensory consequences, and the limbs must all be integrated to allow modifications to either the schema instance selector or the actual generalized schema. Given sufficient execution time, Pew allows modifications to ongoing movements either by low-level corrective mechanisms to the movement pattern or the initiation of a modified schema instance. But we want to know how the schema structure is updated according to corrections made during a movement so as to improve the same movement in the future.

Pew hedges at this point and claims that, at the time of his theory, it was too early to determine the nature of the changes resulting from experience. He hazards the guess that learning involves modifications to the generalized schema structure, to the process of choosing a schema instance based upon environmental conditions, and to the nature of the implementation of the motor command sequence as generated by the movement pattern. These latter two imply that learning involves changes in the processes that control the generation of movement. In general, this is an undesirable position unless satisfactory constraints are imposed on the allowable changes. However, remember that Pew was mainly focusing on performance. He does make an important point about learning, once again relating to the multiple levels of feedback. He claims that the knowledge of results for a given movement is not sufficient to allow the subject to improve performance. According to Pew's model, "information about the expected sensory consequences, and about the actual sensory consequences together with the success or failure of the movement pattern, all converge in the Comparator Mechanism to produce the basis for modifications to the generalized schema, the instance selection rules, and the temporal implementation of the command sequence" (p. 32).

This broader view of feedback and comparisons, which incorporates multiple levels of information, gives Pew's theory more explanatory power than Adams' account. But before comparing these two theories, we turn to Schmidt's schema theory, which synthesizes those of Adams and Pew.

### 2.3.3 Schmidt's Schema Theory

Adams' and Pew's theories, proposed in 1971 and 1974, spurred a flurry of experimental studies testing the predictions and claims contained therein. Schmidt proposed his schema theory (1975b) largely in response to explanatory weaknesses that were revealed as a result of these studies. However, Schmidt credits both Adams and Pew for his conceptual foundations, and the similarities to both are striking.

#### MEMORY STRUCTURES

Schmidt takes the ideas of the motor program (movement pattern) and the schema from Pew and develops them more fully. The latter avoided the term *motor program*, although he did think of his schema instance as "a computer program waiting to be read" (p. 31). The motor program here is analogous to Pew's schema instance, but perhaps a bit more generalized. It is presented as requiring multiple parameters for full instantiation. Parameters include speed, as with Pew's schema instance, but also force, distance, and the possibility of others that are unmentioned. The motor program is intended to provide the means of producing a whole class of similar movements from a single memory structure. This occurs in the same way that a program designed to calculate the average of a set of numbers is usually not limited to the calculation of a single average for a fixed set of numbers. Instead, it can calculate virtually any average given the input data. In this way, Schmidt's motor program is actually a means of producing a sequence of muscle commands based upon parameters and is not the actual sequence of commands itself. The motor programs are stored collectively under, or at least are indexed through, the motor schemas.

As mentioned above, the idea of the motor schema is not new. In Schmidt's theory, it is viewed as a general rule that can be used for generating, or selecting, a motor program. In this respect it is like Pew's schema, which bundled the movement patterns. However, Schmidt proposes three different types of motor schemas – the *recall schema*, the *recognition schema*, and the *error-labeling schema* – and goes into greater detail of description than Pew. Like the work on verbal behavior and memory, the recall schema is responsible for producing movements, whereas the recognition schema is responsible for recognizing particular movements.

The recall schema is an abstraction of previous attempts at a particular class of movements. Specifically, the abstracted information includes the initial conditions at the beginning of the movement, the response specifications, and the response outcome from each movement. The initial conditions are simply a representation of the beginning state of the subject and the environment. The response specifications correspond to the parameter values used in the motor program that generated a particular movement instance. Finally, the response outcome is a qualitative assessment of whether or not the original higher level goal was satisfied. This is commonly referred to as *knowledge of results*, since there is an implied ability to make a judgement about the success of

the movement. These three pieces of information are collected and stored, as in a vector, and it is the relationship among all of them that is captured as a recall schema.

The recognition schema is similar to the recall schema, but instead of storing the response specifications, it stores the actual sensory consequences. As before, the sensory consequences are the trace of feedback (not limited to proprioceptive) resulting from a particular movement. Thus, the initial conditions and the response outcome are again stored, along with the sensory consequences, and the relationship among these three is abstracted to form a schema.

Finally, the error-labeling schema takes the raw sensory signals coming from the limbs and the environment, and converts this input into a qualitative evaluation of the completed or ongoing movement. This labeled error signal is known as *subjective reinforcement* and can be substituted for true knowledge of results, although it will be less accurate. The error schema stores the past sensory signals along with the actual knowledge of results and builds a relation between knowledge of results and the sensory signals received. Once this relation is well developed from previous experience, it can be used to predict the movement outcome just from the sensory consequences.

In summary, Schmidt proposes three types of schemas – the recall, recognition, and error-labeling schemas – in addition to the motor program. Next we look at how these structures are used together to produce skilled, controlled movements.

#### PRODUCING MOVEMENTS

The performance component of Schmidt's theory can be split into two parts or phases – the movement preparation stage and the actual movement generation. These happen in sequence, but they can loop as well. His theory assumes that a motor response schema (combined recall and recognition schemas) already exists.

The movement preparation stage involves taking the specified desired outcome and determining the initial conditions. Based upon the relationship developed over previous movement experience between these two variables and response specifications, the motor program is supplied with a new set of response specifications (hopefully appropriate to the situation and desired outcome). The initial conditions and desired outcome may never have been encountered before, and the resulting response specifications will be determined by "interpolating among past specifications" (p. 236). This may result in novel behaviors that have never been performed before. Simultaneously, the response schema selects the expected proprioceptive and exteroceptive feedback based upon the relationship between previous outcomes, initial conditions, and sensory consequences. Once the motor program and expected sensory consequences have been prepared, the actual movement can be initiated by running the motor program on the limb effectors.

As the muscles are activated by the motor program, the movement proceeds uninterrupted for the first 200 msec. That is, the motor program completely specifies the movement for at least this initial period. When sensory feedback becomes available, it is compared against the expected sensory consequences as given in the recognition schema. Note that the actual sensory information is coming both from the limbs and the environment, and that the expected sensory consequences likewise include multiple modalities. This comparison leads to a raw error signal which is fed back

to the schemas so that adjustments may be made if necessary. The error signal is also input to the error-labeling schema for a qualitative evaluation that results in subjective reinforcement.

Once the raw error signals and subjective reinforcement are available, the entire process begins again. The desired outcome will be the same, but there will be new initial conditions and a potentially different motor response schema based upon the immediately prior movement. Each segment is performed in open-loop mode. This cycle repeats, effectively yielding closed-loop control, until the resulting error signals indicate no further movement is necessary, or until the subjective reinforcement predicts the accomplishment of the desired outcome.

#### MODIFYING THE RESPONSE SCHEMAS

Schmidt proposes that the schema structures are modified by the trace from each movement. A trace starts with the initial conditions and response specifications, with the sensory consequences being added when they become available. Finally, at the end of the movement, the outcome of the movement is added to the trace, either in the form of knowledge of results or as subjective reinforcement. These four items are used to revise the means of predicting sensory consequences and response specifications on future trials. A trace is hypothesized to be rather short-lived in duration. Although this trace is unstable as a memory structure, it persists long enough to modify the recall and recognition schemas in memory.

The schemas are much more permanent memory structures that are generally resistant to forgetting. The strength of the schema increases in proportion to the number of trials of a particular class that are "sufficiently similar" to be grouped together. Also, the reliability of the relationship given in the schema increases with better quality feedback from the response outcomes.

However, the nature of the modification to the schemas is difficult to assess. Schmidt uses the term "abstraction" to describe the process of bundling up the four pieces of information described above. He states that "it is the relationship among the arrays of information that is abstracted rather than the commonalities among the elements of a single array" (p. 235). By this he seems to mean that the multi-way relationships between the four items is more important than the relationship between any particular set of initial and final conditions, response specifications, and sensory consequences. This is important because the methods for choosing the response specifications (and sensory consequences) rely on interpolating between previous experiences or using a function that is based on an interpolation of previous experiences. Recall and recognition schemas are both treated similarly with respect to learning.

The formation and modification of the error-labeling schema is even less well formulated than with the recall and recognition schemas. The strength of this schema again depends on the amount and the quality of prior experience. Previous raw error signals (the discrepancies between the expected and actual sensory states) have been stored in association with the resulting qualitative feedback (knowledge of results). Of course, the schema as a whole would have to be associated with the recall and recognition schemas to allow retrieval, since the initial and final conditions are *not* part of this memory structure. Again, as in Adams' and Pew's theories, we see that Schmidt's

framework leaves much of the learning processes to the readers' imagination. However, we can still compare these theories' learning components, their explanatory powers, and their complexities.

#### 2.3.4 Analysis of the Three Theories

Although there are many similarities among the theories we have discussed, each has strengths in different aspects. All three theories contain feedback components, but only the first two, Adams' and Pew's, should be considered as closed-loop theories of motor control. In these models, once the movement is going, the control is based on feedback compared with the standard of correct movement. On the other hand, Schmidt's theory uses feedback to revise the selection of open-loop movements in the course of trying to satisfy the desired behavior designated to the motor system. In Schmidt's theory, each individual segment is considered to be under open-loop control. This actually blurs the distinction between closed-loop and open-loop processing.

Furthermore, Adams' and Pew's theories are very much alike in form and process (with the exception of Pew's omission of learning), but mainly different in representation. Adams recognizes the need for two memory structures, whereas Pew avoids this point by introducing a second structure, the expected sensory consequences, from the movement pattern used to generate the movement. On the other hand, Pew's inclusion of a schema memory structure allows greater flexibility in movement generation. Schmidt's overall framework bears many similarities to Pew's in representational structure, but borrows from Adams' in processes for learning and the basis for the recognition schema. From a purely theoretical and structural view, Schmidt borrows heavily from previous work, but his synthesis stands as a significant improvement.

As we stated at the beginning of the paper, the purpose of considering the human phenomena was to evaluate and constrain theories of human motor learning. All of these theories can account for the speed-accuracy tradeoff by the greater number of chances to correct errors during slower movements. However, whether the quantitative results from these theories would correspond to those predicted by Fitts' law is an open question. Such verification would require instantiating these theories as computational models – which has not yet been done. Similarly, the transfer of skills between limbs could probably be handled by appropriately transforming the memory representation for a given skill to be executed on another limb.

Since Pew's theory does not explicitly address learning issues, we cannot say much about his theory with respect to the learning phenomena. Certainly, all three theories predict improvement based upon experience, but whether any of them would yield power-law learning curves is difficult to answer. Even if the theories were stated in computational terms and allowed the collection of numerical results, there would still be the problems associated with discriminating power-law curves from exponential ones (Newell & Rosenbloom, 1981; Rosenbloom, 1986).

The closed-loop and open-loop distinction provides a better contrast between the theories. Adams' and Pew's models cannot easily account for any open-loop behavior. The former's memory trace could conceivably become sufficiently strong that simple movements could be performed in open-loop mode. Pew's schema instance can be forced into open-loop mode, since it is converted to a temporal sequence of muscle commands that theoretically could be executed entirely without



feedback. Schmidt's theory is almost entirely open loop, although it can give the appearance of closed-loop behavior. However, none of the theories give good explanations of how behavior could progress from closed loop to open loop as a result of practice.

Finally, only Schmidt's schema theory is able to explain the practice variability effect. Of course, this phenomenon was predicted by (and observed after) the introduction of his schema theory. As discussed by Schmidt (1975b), Adams' theory has no way to account for such a phenomenon. However, Frohlich and Elliott (1984) claim that even Schmidt's explanation is too weak and they present an alternative view on this subject. Although the empirical results are still inconclusive, it seems clear that, at least in some cases, the effect holds consistently. A full theory of human motor learning should be able to account for at least some of these effects.

All of the theories (including Pew's with a hypothetical learning component) explain the psychological phenomena rather well (not surprisingly). However, they are all limited to simple, ballistic movements. Most work has been done on single-joint tasks in one dimension. Consequently, the existing psychological theories have little to say about more complex tasks involving the interaction of multiple joints in non-trivial manners. As mentioned above, a computational model of these theories would facilitate a more thorough evaluation and, in general, could provide much needed insight to the nature of such theories.

## 2.4 Computational Approaches to Motor Behavior

Now let us consider models of jointed motor control that specify the representation, performance, and learning processes as computational mechanisms. Again, we must choose some method or dimension to limit the systems we consider in this chapter. In this case, we will focus on heuristic methods that employ learning techniques to sidestep weaknesses in computational power, along with systems that are heavily geared toward modeling some aspect of human motor control. This means excluding much of the robotics literature in so far as the methods commonly used in that area are intended to find exact or optimal trajectories for mechanical manipulators. Also, such methods tend to focus on low-level motor control, involving torques and voltages, which we intend to ignore.

We will also exclude the literature on robot planning (e.g., Segre, 1987; Andrae, 1985), which is mainly concerned with problems of planning and operator sequencing, as opposed to the execution of varied limb movements. Of course, both this type of work and the low-level robotics work are important in their own right, but they are not directly related to the concerns of this chapter. As we stated before, we are interested in theories or systems that address both the cognitive and physiological aspects of motor behavior.

We start by considering several systems that have been designed as models of the human motor system or that have paid close attention to constraints imposed by this system. Then we turn to several other implementations that deal with the control of dynamic systems and that could conceivably be applied to jointed limbs, but which are not explicitly presented as models of human motor control. We close by examining the plausibility of both types of systems with respect to the constraints and phenomena we introduced earlier.

### 2.4.1 Chunking Goal Hierarchies as a Model of Motor Learning

Rosenbloom (1986) presents a model that accounts for both the power law of practice and the reaction time data on stimulus compatibility. The latter phenomenon concerns the effect on the reaction time to a given stimulus, according to the compatibility between that stimulus and the required response. For example, if a tone in the left ear requires a button press with the right hand, the reaction time will be longer than if a button press with the left hand were required.

Rosenbloom's XAPS architecture accounts for both of these phenomena. The representation consists of goal hierarchies that determine the solutions to particular tasks. These are mostly simple choice reaction-time tasks in which an appropriate response must be selected to a given stimulus. The nature of the goal hierarchies used to solve these tasks gives rise to the compatibility effect. Learning consists of creating chunks from sequences of subgoals that have been solved in a given situation, and the coinciding decrease of necessary processing explains the power law of practice.

This model can be viewed as an explanation of task-independent practice effects; however, we are specifically taking a motor learning perspective. It accounts for the two phenomena mentioned above, as well as a number of others, but it does not explain such phenomena as the speed-accuracy tradeoff, sequential dependencies, interference, discrimination, and reaction time distributions. The model has been applied only to tasks that involve minimal motor control – the execution of a selected response – and these responses have been modeled as primitive operators. However, one can imagine adapting the architecture to include lower-level motor primitives, allowing the creation of goal hierarchies of motor movements and subsequent chunking of portions of such hierarchies. A further limitation is the absence of a mechanism that can acquire the necessary goal hierarchies. Several extensions are described that could conceivably alleviate this limitation. Although Rosenbloom's theory is rather weak on issues of motor control, it is the only model we will consider that significantly address cognitive aspects. As such, it perhaps holds the greatest promise for addressing both high-level planning issues and low-level control issues, but the details have not been specified, and so we turn to a model that focuses on low-level control issues.

### 2.4.2 A State-space Model of Motor Learning

Raibert's (1976) model of motor control and learning is one of the most serious attempts at carefully dealing with issues in the human motor system. He presents four properties of this system that he attempts to model: the ability to gain control of the limbs through experience, the ability to maintain control in the context of changes to the limbs, the ability to compensate for mechanical interactions between serial joints, and the ability to convert a desired movement from one representation to another. He qualifies this model as only a sub-system of a more complete model of motor control and learning. In particular, this sub-system is responsible for acquiring appropriate feed-forward commands. This constraint allows the model to ignore interactions with the environment (which would require a feedback mechanism) and the issue of motor programs (although their existence is not questioned). The model is intended to process the class of ballistic movements, such as swatting a fly or swinging a bat.

Raibert's work focuses on the construction of a translator that takes descriptions of desired movements and converts them to commands directly interpretable by muscles or motors. The main difficulty of such a task is encoding or solving the mechanics of the particular limb. In Raibert's model, this information is extracted from the relationship between the limbs' inputs and outputs that result from previous attempts to move or position the limb. This extraction is made feasible by discretizing time and space. Time is sliced up into sufficiently small pieces to allow the simplification of the equations describing the motion of the jointed limb to a set of constants. These constants cannot be stored for the infinite number of possible states of the arm, so the state space of the arm must be divided into regions or hyper-cubes. This memory associates one set of constants with each hyper-cube in the state space. These constants are assumed to be satisfactory for "near" states, or ones within the same hyper-cube (given sufficiently small hyper-cubes). This process is referred to as a *piece-wise linearization* of the mechanical system representing the limb.

Learning in this model involves the storage of the parameters for individual states of the state-space memory. The constants stored are based on averages of previously calculated values for given situations. The calculation is based on the commands issued to the limb and the resulting accelerations (see Raibert, 1976, for details). As experience occurs, more parts of the state-space memory are visited and filled. On average, behavior will improve as a greater percentage of this memory is filled in. Noise in measuring the accelerations of the joints is dampened by averaging the calculated constants with existing values in a particular hyper-cube of the state-space memory. One might obtain practice variability effects from this model, since the novel task will be "closer" in the hyper-space to previous experience in the variable practice condition than in the constant practice condition.

### 2.4.3 Generalizing Motor Control Using Knowledge

One of the limitations of Raibert's (1976) tabular approach is that transfer between dissimilar movements is difficult or impossible. Atkeson (1987) presents an adaptive feed-forward method that overcomes this limitation. His system acquires a global model of the arm dynamics that requires one to learn only one set of parameters for the equations. This contrasts with the many sets of parameters necessary in tabular approaches, where each set of parameters applies only to the small, corresponding region of the state space. Not only does Atkeson's approach reduce the number of necessary parameters, it also reduces the learning necessary to achieve a comparable level of performance. As stated above, the state-space method must "explore" the space of possible arm states and store parameters for each, whereas the global model can be learned in just a few "test movements". The system requires torque/force sensors at the wrist and arm joints in order to measure the torques resulting from the test movements. Given the relationships between the measured values and the commands, the system can infer a model of the rigid body dynamics for the arm. Note that the table lookup methods did not require torque sensing devices on the arm but only the ability to sense where the arm was currently positioned in joint coordinates.

The global model lets the parameters be used for controlling a variety of movements within the given arm's state space. Unfortunately, using the global model to assign the parameters introduces small errors, which arise because the arm is not entirely rigid, as the global model inference mecha-

nism assumes. If the global model were modified to correct for these small errors in one particular trajectory, the performance on other movements would in turn deteriorate. Instead, Atkeson includes a mechanism for learning single trajectories that takes advantage of both the global model and the feedback information from a particular attempt at executing the trajectory. Given several practice attempts, the commands for the trajectory can be improved to a level arbitrarily close to the sensitivity of the manipulator hardware. The introduction of a single-trajectory learning mechanism involves altering the control system memory to allow the storage of commands for particular trajectories. The details of this memory are not discussed, and it appears to be an unwieldy addition to the system.

For future research, Atkeson proposes the use of local models that would store the more correct dynamic model for local portions of the space. This proposal involves either learning the dynamics of a "central" movement for a set of similar movements or a tabular approach giving the dynamics for a local portion of the space. Either way, the local model would serve as a correction factor to the global model when generating the feed-forward commands of a movement related to the local model. A unique feature of this proposal is that it effectively suggests a hierarchy of models. This allows a tradeoff between the generality of the global models and the accuracy of the local models that would "gain the benefits of each and the drawbacks of none" (p. 30).

#### 2.4.4 A Connectionist Approach to Hand-eye Coordination

Recently, connectionist and neural network architectures have received considerable attention as models of human cognitive processes, and Mel (1988) presents a robot arm controller called MURPHY that utilizes such an architectural framework. Although he did not specifically intend this system as a psychological model, the design process was constrained by knowledge of nervous system structures and their operation.

The architecture is based on two interconnected sets of neuron-like units. A visual array represents the field of view and a kinematic population represents the angles of the three joints that are controlled by MURPHY. These units are overlapping, so that a single image or joint angle will activate a small population of units; this distinguishes the approach from state-space schemes. Learning involves the creation of weighted associations between these two populations of units. The visual units that are activated by the joints are associated with the joint angle units that describe the position of the arm. Because of the overlapping structure of these populations, the level of activation for a given set of units decays gradually as the arm moves away. Training consists of stepping through a representative portion of the possible joint configurations and creating the weighted associations.

After training, MURPHY can "grab" a visually presented object. The distance from the tip of the arm to the goal is evaluated and a move is selected that will reduce the distance by the greatest amount. This is described as an internal search, after which the arm is moved to the target destination in a single execution. Mel presents no results on learning, but it seems plausible that the number of search steps should decrease with the extent of training. Alternatively, the search trajectory should approach the straight line between the initial and target configurations

as training is increased. The approach is an interesting one, although the current system is very limited in that it has no facility for the representation, execution, or acquisition of arbitrary arm trajectories. Still, it bears further attention as MURPHY continues to be developed.

#### 2.4.5 Adaptive Feedback Control

All of the systems we have considered in this section have either used a constant feedback controller or ignored feedback entirely. Improvements in performance were gained by modifying the commands responsible for generating the original movement. There has also been considerable research in the area of adaptive mechanisms for feedback control; that is, feedback controllers that learn from errors in previous experience. Several of these studies have focused on the "pole-balancing" task (Michie & Chambers, 1968), which consists of a cart on a one-dimensional track with a pole attached via a hinge. The cart can be moved left or right with a constant force. The goal is to keep the pole in a near vertical position by selecting appropriate sequences of left and right forces on the cart. Although these systems have not been proposed as models of human motor control, in some cases they have been associated with claims as to the viability of the approach for robotics in general (Sutton, 1984; Selfridge, Sutton, & Barto, 1985).

Michie and Chambers (1968) implemented an early program, BOXES, utilizing a reinforcement learning mechanism in the pole-balancing domain. They used an independent-association approach that involved discretizing the environment into a state space using pre-defined ranges. The average time to failure (falling of the pole) was updated from experience and the action with the longest average was selected for a given state. This should not be confused with Raibert's state-space memory, which discretized only memory and not experience. That is, Raibert distinguished between arm configurations down to the resolution of the sensing equipment, but used the same set of constants in the dynamics equations for both configurations if they fell within the same hypercube. In BOXES, two cart-pole configurations are considered identical if they fall within the same region of the discretized space. That is, as the system learns the appropriate action to make in given states, the only generalization would be to other configurations considered as the same state. Sutton (1984) and Selfridge et al. (1985) present another reinforcement learning method using a linear-mapping approach. This also required the discretizing of the space into regions, but the choices made in a region are based on the probability of maintaining balance. The number of trials required to learn to balance the pole for some criterion number of time steps was significantly less than BOXES. Connell and Utgoff (1987) present another program, CART, that does not discretize the space and further reduces the required learning time. Their system employs a Shepard function to determine the degree of desirability of a particular state (cart-pole configuration), and learning involves adding a point from the cart-pole space with an evaluation of its desirability (provided by a critic) to the instance memory. CART learned to balance the pole in less than 16 trials, as opposed to an average of 75 for Selfridge et al. and 600 for BOXES.

Although these systems have no provision for motor programs or feed-forward control of any sort, they represent important progress in adaptive feedback control. A mechanism that can improve its responses to errors is an important part of a complete model of human motor behavior. However, the amount of increased understanding from these systems is limited. The approaches are made

manageable by the simplicity of the pole-balancing domain, in which there are only two operators. Also, when applied to the control of robotic arms, the complexity of the state space will increase dramatically. This does not mean that these problems cannot be overcome, but it does mean there remains a need for continued work in all areas of motor control.

## 2.5 Conclusions

In this chapter, we have attempted to cover multiple facets of the literature on motor behavior and learning. There exists an enormous amount of previous work and some means of constraining the coverage must be employed. We have focused this survey around our goal of developing a computational theory of human motor behavior that can learn to perform complex tasks such as swinging a golf club, shooting a basketball, or juggling pins. We selected some of the more significant phenomena as a basis for constraining the type of motor model we would examine. The leading psychological theories were considered in this context, followed by a number of implemented computer models and systems.

Our real interest lies in a computational model of human motor learning on reasonably complex tasks. That is, we want to move beyond ballistic movements to skills with complex trajectories. In such movements, the path of the arm is of primary importance rather than the ending position. The survey of phenomena was intended to constrain and help evaluate psychological models, but we considered existing theories in the hopes of building on previous work.

Although the psychological theories accounted for the phenomena rather well, we were unsatisfied with their level of operationality. Considerable amounts of detail were left to the reader's imagination, and it is relatively easy to account for phenomena if the level is abstract enough. Even if the effort were made to implement these theories, they would still be limited in scope to simple, ballistic movements. In contrast, our model of motor behavior, described in the next few chapters, borrows many ideas from the psychological theories reviewed here, but is not a direct implementation of any of them.

For the most part, the computational work on motor control has focused on low-level issues of controlling the hardware. These contributions tell us little about how humans direct their limbs or the types of behaviors one can expect from humans in particular situations. Furthermore, the computational work has ignored the task of recognizing motor skills when performed by another agent. Finally, these models typically address only one movement task at a time. They do not present accounts for how different skills can be stored and organized as concepts in long-term memory.

In summary, there remains a need for a computational model of human motor behavior. The phenomena identified in the literature provide a set of constraints for such a model and a framework for evaluating it. The psychological theories provide many ideas for organizing the processes that control the recognition and generation of motor skills. The computational approaches provide little theoretical influence for the kind of model we want, but they do provide low-level mechanisms that our model may rely upon for manipulating jointed limbs. We now turn our attention to the design and implementation of *MEANDER*, our approach to the goals set out in Chapter 1.

## A Computational Theory of Motor Learning: An Overview of the Mæander System

---

### 3.1 Introduction

In the previous chapter we examined a number of phenomena that have been consistently observed in humans. These provide a number of possible constraints for a computational model of human learning behavior. Additionally, in Chapter 1 we specified a set of characteristics, one of which was that our desired model address complex movements. The psychological work discussed in Chapter 2 has not addressed the range of movements we are targeting. We want our model to go beyond this set of carefully studied phenomena, yet still be consistent with them. We want to begin answering more general questions such as “How is a tennis serve initially learned?”, “How do children learn to write and draw shapes?”, and “How do adults master extremely complex or difficult motor tasks like playing a violin or throwing a knuckle ball pitch?” The range of tasks represented in this set of questions involves at least two well-defined stages or types of learning. First, people learn from observing others performing particular skills and, second, people learn through practicing those skills.

These two types of learning imply an acquisition mechanism and an improvement mechanism. We posit that any *comprehensive* theory of motor learning must address both of these stages. In order to acquire a skill, either it must be created from nothing (e.g., through exploratory practice), or it must be communicated by another agent (e.g., through demonstration or advice). In a rich environment, such as the one in which humans live, both sources are constantly providing information from which to learn. To make sense out of the host of observations available, a given movement must be *classified* or recognized. When attempting to improve a skill through practice, the agent must assign blame to the current form of the skill. This can occur either through a teacher who observes the practice and informs the learner of mistakes, or by comparing feedback to a “mental” image of the desired movement that was previously acquired through observation.

Unfortunately, the questions posed above are too broad to be dealt with effectively by the current state of the art. In order to progress toward such a complete theory, we need to constrain the search in two general ways: we must limit the tasks that are addressed by the theory and we must simplify the world in which these tasks are performed.

### 3.2 Refining the Task Specifications

The term *skill* has been used in a wide variety of contexts not limited to motor behavior. In this work we will narrow its use to refer to the specific task of representing and following *trajectories* of the parts of a limb in two dimensions. That is, we are interested in models that let a trajectory be represented, stored in memory, and replicated with a given manipulator or set of effectors. This contrasts with the more commonly studied task of reaching for an object at a specified position, where the desired or final state of a movement drives performance. In our task, the movement itself determines the resulting endpoint, which is secondary to the behavior generated.

In light of the preceding discussion, we define two performance tasks and two learning tasks. The performance tasks correspond to two (of the potentially many) competencies that must be addressed by a general theory of skilled movement. The first of these two tasks is:

- *given*: an observed movement in the environment;
- *classify*: the movement according to previously stored experiences.

That is, a new observed movement is considered in the context of the agent's current level of knowledge about movements in general. This amounts to recognizing the new movement as either similar to some type of movement previously observed or as something completely new. We measure success on this task by determining how well the learner classifies or recognizes an observed movement. Recognition of an observed movement implies the ability to predict some missing information about the movement. For example, in American Sign Language, many words or concepts are denoted by motions – not just configurations of the hands and arms. Recognition in this case means retrieving the appropriate concept from long-term memory given the observed movement. Furthermore, if an agent observes the first portions of a “throw” concept, the agent might recall that such movements precede moving projectiles, recall that such a projectile would intersect the agent's position with high probability, and decide to get out of the way.

The associated learning task is to improve the ability to recognize or classify movements as a result of experience. This statement of learning as “improvement in recognition” will be viewed in the context of *unsupervised* learning, where the observations are not labeled by a teacher and must be organized and labeled by the learner. Improving the ability to recognize could imply a more rapid or efficient classification process; however, here we will focus on increasing the accuracy with which the trajectories of the various movement types are recognized. We will measure the similarity between the observed movement and the average trajectory associated with the selected concept that classifies the new movement. But these are evaluation issues, to which we will return in Chapters 6 and 7.



The second task that is necessary to our comprehensive theory of movement skills is the *generation* of movements. Again, because we are limiting our discussion to following trajectories, we will not address the full range of generative behaviors. Our task can be stated as:

- *given*: a desired trajectory for a jointed limb
- *move*: the limb through positions over time that correspond to the desired trajectory.

This assumes that the agent can control its manipulator within the environment. We will turn to these issues of control shortly. We will measure the performance of an agent on this task by comparing the similarity of the generated movement to the desired trajectory. Given the performance task outlined above, the obvious learning task is to improve the movement of the limb as a result of movement experience or practice. Naturally, improving a movement skill means modifying it in such a way that it corresponds more closely to the desired movement.

The first learning task above can be thought of as unsupervised trajectory learning, whereas the second can be thought of as supervised trajectory learning. In the next several chapters we present MÆANDER, a computational model of skilled movement acquisition and improvement, as a response to both of these tasks. In accord with the previous chapter, this model has been designed to account for a number of the constraints and phenomena that have been identified in the psychological literature.

In the remainder of this chapter we outline the envisioned contexts in which MÆANDER will function, describe the simplifying assumptions that we have made, and provide an overview of the system architecture. In the following two chapters we consider the two major components of MÆANDER – OXBOW and MAGGIE. These chapters provide detailed descriptions of the tasks introduced above and the mechanisms that achieve these tasks.

### 3.3 Mæander's World View

Skill learning cannot occur in the absence of some performance task, and any performance task requires some environment in which to perform. In this section we describe the associated features and requirements that make up the environment within which MÆANDER operates. As implemented, our model interacts with a simulated environment that contains a jointed limb in two dimensions. The features and requirements for this simulated environment can be thought of as a set of inputs to the model.

#### 3.3.1 Inputs to the Model

MÆANDER's performance component incorporates only very general assumptions about the nature of the agent and its environment. Additional inputs required for its operation include:

- a *simulated environment* in which to operate, along with a set of objects existing in this environment;<sup>2</sup>

---

2. Some of these objects will correspond to the agent's *effectors*, which it can use to manipulate the environment.

- an *effector* such as an *arm*, which can be manipulated by the agent and which has well-specified relations with other objects in the environment;
- a *sensorimotor interface*, which handles communication between the agent and the environment.

We will consider each of these inputs in turn.

#### THE SIMULATED ENVIRONMENT

Rosenbaum (1985) has argued that motor behavior implies purposes and that purposes necessitate an agent. However, it makes no sense to refer to an agent in the absence of the environment in which it operates. One can conceive of alternative environments that obey physical laws different from those in the real world, but since we are interested in human motor behavior, we will consider a “standard” environment. However, this flexibility indicates one of the advantages of using simulated environments.

A complete specification of an environment entails listing all the objects and their associated attributes. Interactions between objects must be defined, such as the nature of connections and collisions. For the purposes of developing and testing our model, we have implemented a simple environment that contains objects with position, length, and velocity, but that ignores mass, friction, and force. In the experiments reported in this dissertation, the only objects in the world are the components of the agent’s arm. We could directly apply *MEANDER* to a more complex environment that includes free objects and interactions between them. However, given the current set of simplifying assumptions described below, this would not add richness to the work.

#### THE ARM

We think of an *arm* as a collection of objects in the environment that an agent can manipulate in certain predefined ways. Although the components or links of the arm are specified as ordinary objects in the environment, the arm merits special treatment here because of additional attributes that are inherently necessary for jointed movement.

We can think of the links of the arm as regular objects that are connected by joints. A joint, rather than being an object in the world, is a relation that exists between two objects that are attached to each other. This relation determines the relative positions and orientations between two kinematic links. Such a relation has certain properties that influence or determine the behavior of the two links that are connected.

In general, a joint’s attributes would include the type of joint, its friction coefficient, its maximum force and velocity, and its range of movement. However, for the purposes of our implemented system *MEANDER*, we have made a number of simplifying assumptions. First, the joints we consider are restricted to hinge joints – those having a single degree of freedom. These would be analogous to the human elbow joint. Multiple hinge joints can connect arbitrary links, but the axis of rotation must be perpendicular to a common plane. That is, we limit all movement to be in the plane.

Finally, we have ignored effects of mass, friction, force, and inertia. This reduces the meaningful attributes to the limits on allowable rotations and on rotational velocity. Currently, we restrict each joint's motion to the range  $(-\pi, \pi)$  with respect to the zero or resting position. This allows a complete circular movement, since the range lets the arm rotate halfway around a circle in each direction, but it prevents any continuous circular movements where the arm is repeatedly swinging in circles.

#### THE SENSORIMOTOR INTERFACE

An agent cannot interact with its environment unless it can perceive that environment and control its effectors. In our simulation, both of these are accomplished through a sensorimotor interface. The 'motor' component of the interface lets the agent control the motion of its arm. The 'sensory' component relays sensory information to the agent about the location of objects – in this case, just the arm.

The transfer of sensory information can be viewed as a filtering operation. Essentially, the sensory filter takes a complete description of the world and passes a subset of this information to the agent. MÆANDER accepts two forms of sensory input: visual information giving the absolute positions and velocities of objects, and proprioceptive information giving the relative positions and velocities of the arm's joints with respect to the previous joint.<sup>3</sup> Visual information is given in a *viewer-centered* representation, whereas proprioceptive information is provided in a *joint-centered* representation. We give detailed descriptions for both of these coordinate systems in the next two chapters.

The motor interface can also be viewed as a filter, since not all possible motor commands are legal in the simulated world. For instance, if the agent specifies an arm movement that would exceed the allowed ranges, the interface filters or "clips" the command so that the resulting movement is within the allowed limits. Likewise, if a sequence of commands would cause a joint to exceed the rate at which it is allowed to move (rotational velocity), then the resulting movement would reflect the maximum allowable velocity during those periods in which the limit was exceeded and would therefore not end up where the sequence of commands specified. Except for such cases, controlling the arm in MÆANDER amounts to simply setting the relative positions of arm components to the values specified by the agent's movement commands. Of course, these commands must be given in a representation that corresponds to the local rotations of each joint. This joint-centered representation will be discussed in full detail in Chapter 5.

#### 3.3.2 Assumptions of the model

At the most abstract level, the items discussed above can be thought of as inputs to our theory. That is, the model's operation is partly dependent on the instantiation of the above inputs. In the discussion of these inputs we introduced several simplifying assumptions. To review, we ignore friction, mass, force, and inertia, we restrict each joint to a single degree of freedom, and we allow joints to move in two dimensions only. It is important to note that these assumptions relate to

---

3. We define the *previous joint* as the joint that is adjacent and closer to the base of the arm in the kinematic chain.

our current implementation of MÆANDER and not to the model itself. However, our theory does include several assumptions that are more fundamental, but that are based upon what is known about human movement. These assumptions can be considered as constraints imposed by the real world.

First, we assume that the motor interface receives commands specifying the rotational increment for each of the joints and causes the arm to move accordingly. This implies a complete set of mechanisms whose responsibility it is to calculate and apply the appropriate torques at each of the respective joints given the current state of the arm. On the computational side, this is the domain of traditional robotics applications, and we are happy to assume that such lower-level mechanisms are available in pre-packaged form. With respect to human motor behavior, evidence indicates that humans can "set" the positions of limbs without feedback (Kelso, 1982). Therefore, we will continue with our high-level approach and not concern ourselves further with low-level neuro-motor issues.

We also assume that movement representations are invariant with respect to time. In our model, movements are carried out by a sequence of commands specifying the rotational velocities of each joint (in a local polar-coordinate system) for each time-slice over the course of the movement. Internally, MÆANDER represents these movements as a few control points. Therefore, a single representation can be used to execute a movement at different speeds. The speed would be declared at run time instead of being stored in memory. Again, this assumption is consistent with existing knowledge about the observation and generation of human movements (Rubin, 1985; Schmidt, 1982b; Kelso, 1982).

We want our model to explain and relate to a wide range of movements and experience; however, we have restricted ourselves to the class of movements that are generated by linear accelerations at each of the joints. By this we mean that the rotational acceleration at each joint changes linearly over time.<sup>4</sup> The motivation for this position is our use of a cubic parametric equation to describe movements; we discuss these details when we introduce our representation of motions in the following chapter. The implicit assumption in this design decision is that the space of movements generated with linear accelerations is a rich and varied space of movements.

We propose that skill improvement occurs after either observing or executing a movement. This implies the existence of a memory that can store information about the arm positions during a recent movement. We will call this structure the *motor buffer*. This is analogous to a pre-perceptual store that has rapid decay, thereby allowing only limited access (Sperling, 1960). Schmidt (1975b) assumes an analogous structure in the context of his recognition schema discussed in Section 2.3.3.

Finally, we know from experiments on human subjects that there is a minimum time that is required before alterations to an ongoing movement can be initiated. This cycle time has been found to be 200 msec. in humans (Schmidt, 1982a; Pew, 1974). This means that if an error in a movement is detected, at least 200 msec. must pass before the subject can initiate any corrective measures. We refer to the minimum cycle time as the *feedback delay*.

---

4. Naturally, we allow the slope of the line to change at specified transition points. With a sufficient number of transitions, arbitrary acceleration patterns can be simulated. However, relatively few transitions are necessary within our scheme to generate surprisingly complex behaviors.

We will address each of these assumptions in later parts of this dissertation, but explicitly stating each of them here will facilitate explanation. Again, these assumptions reflect constraints on psychologically plausible models, which are imposed by our understanding of the human motor system. This contrasts with the simplifying assumptions discussed earlier, which we introduced to limit problems to a manageable size and number.

### 3.4 The Structure of Mæander

In Section 3.2, we identified two different tasks our theory will address – movement recognition and movement generation. MÆANDER's architecture predominately consists of two subsystems. OXBOW is largely responsible for recognizing movements and acquiring movement concepts, whereas MAGGIE is mostly responsible for generating and improving behavior using the movement concepts stored in memory. However, the subsystems do not divide cleanly along the task boundary of movement recognition and movement generation. Although OXBOW has the dominant role in movement recognition and MAGGIE has the dominant role in movement generation, each overlaps into the other. That is, portions of MAGGIE are necessary to the working of OXBOW, whereas MAGGIE must use OXBOW as an entire sub-routine.

Another way of looking at this distinction is to consider the functionality of each sub-system. OXBOW can be viewed as the memory management and indexing system, which handles all modifications to memory and any recalls from memory. Because learning to recognize movements is undirected and mainly involves cataloging observed experiences, OXBOW dominates the movement recognition process. On the other hand, MAGGIE can be thought of as an execution system that takes abstract movement representations as they are stored in memory and transforms them into movements. This involves a closed-loop feedback control mechanism and a learning mechanism to improve movement representations.

However, for changes to be remembered, they must be stored in MÆANDER's memory. OXBOW handles this storage process, but MAGGIE is largely responsible for movement generation as specified above and for suggesting the changes that could lead to improved performance on future movements. The rest of MÆANDER deals with communications between the two modules and between the agent's sensors and effectors.



## CHAPTER 4

# Learning to Recognize Observed Movements

---

### 4.1 Introduction

Human motor behavior covers a remarkable range of abilities – from simple tasks such as an infant’s learning to reach for and grasp toys, to complex tasks such as learning to play a violin or to throw a knuckle ball. Although motor learning is usually thought of as improvements in performance as a result of repetitive practice, an agent must first acquire an initial movement in order to improve it. A learner acquires initial movement representations when it is generating movements by chance, observing another agent (such as a teacher) perform a particular skill, or problem solving to achieve a particular goal. In this chapter, we consider the case in which the learner observes movements as they are performed by another agent. As a function of multiple observations, a person acquires the ability to “understand” or recognize a new movement as being similar to a set of previously observed movements. This understanding consists of two steps: breaking a stream of sensory information into a sequence of states (parsing the movement), and finding the most appropriate match of the parsed movement with movements that have been previously experienced and stored in memory (classifying the movement).

Acquiring the ability to understand motion involves clustering sets of similar movements that, taken together, correspond to “concepts”. For example, we would think of “throws” as a class of movements involving an arm and an object (say a ball) that are similar in many ways. Furthermore, we could distinguish among types of throws; for pitching a baseball we might have classes for fast balls, curve balls, and sinkers. As the system learns from observing throw movements, its set of classes should adjust to accurately reflect the domain. Over time, this set of concepts should let the learning agent better recognize and classify movements it observes in the environment.

In this chapter we will focus on movement recognition and show its relationship to the task of concept formation. In the next section, we give a statement of the problem addressed here. Next, we introduce OXBOW, a computer system that embodies some of our ideas about motor learning. To do this, we discuss the system’s representation for movements and concepts, its approach to parsing and classifying movements, and the learning that occurs during movement recognition. We close with a summary of the recognition task as it fits in the context of MEANDER, our overall model of motor behavior.

#### 4.1.1 Statement of the Problem

Movement recognition is the process that occurs when an agent observes others performing particular skills. To attach meaning to an observation, it must be *classified* and related to previously stored knowledge or experiences. We refer to this performance task as *movement recognition*, and define it as:

- *Given*: an observed movement in the environment;
- *Classify*: the movement according to stored knowledge.<sup>5</sup>

Classifying a movement means that the system chooses some “movement concept” (a stored description for a class of movements) as most appropriate for the new movement.

Movement recognition requires that each observed movement be compared to previously stored knowledge. One way to access and update a set of experiences is to cluster them into concepts and arrange these hierarchically. This is one version of the unsupervised concept formation task:

- *Given*: a sequential presentation of instances and their associated descriptions;
- *Find*: clusterings that group those instances in categories;
- *Find*: characterizations or abstractions of these clusters;
- *Find*: a hierarchical organization for these abstractions.

These two task descriptions define both learning and performance for concept formation in general; in this dissertation, we are concerned with the formation of movement concepts.

One important aspect of concept formation is that it is an *incremental* process. This means that learning occurs with each instance, and that the system does not need to reprocess all previously seen examples in order to learn. This is a fundamental constraint imposed by psychological results: humans observe a never-ending sequence of instances, and they can use their learned knowledge at any point in time.

Given the specification of our performance and learning tasks, we now present OXBOW, a system designed to form concepts for use in movement recognition. The methods implemented in this module incorporate many ideas from two earlier concept formation systems – CLASSIT (Gennari, Langley, & Fisher, 1989) and COBWEB (Fisher, 1987).

## 4.2 Representation and Data Structures in Oxbow

Any computational model of motor skills requires some representation to operate upon. Likewise, if such a model is to store and retrieve skills, then it must also have a means of organizing their representations in a flexible manner. In this section we introduce OXBOW’s format for representing observed movements and its method for organizing these representations.

---

5. In order to classify the movement, it must first be parsed into a sequence of states. We will describe this process in more detail in Section 4.3.



### 4.2.1 Representation of Movements

We assume that movements given to OXBOW are generated by a jointed limb and that information about each of the joints is available to the system. This generation may either be observed or performed by the learning agent. In this chapter we focus on observed movements, but our representation is similar for generated movements. Furthermore, although the representation described in this section describes movements of only a single limb, the extension to multi-limb movement is straightforward. A *movement* is presented to the system as a sequence of *state descriptions* that characterize the arm at uniform intervals of time. These intervals reflect the granularity of the system's perception of continuous time. The state of an arm is described by listing the positions, rotations, and velocities for each of the joints at a given time.

Although we present movements to the system as a complete sequence of state descriptions, OXBOW does not store these representations in a long-term memory. Instead, the information necessary to recall and carry out a movement is stored as a *motor schema*. This is similar in intent to Schmidt's (1982b) use of the term. As with our definition of an observed movement, we represent a motor schema as a sequence of state descriptions. However, instead of storing state descriptions for every time slice, a schema specifies the state of the arm at only a few times during a movement. That is, we claim that smooth continuous movements are often adequately described by just a few state descriptions. The intermediate positions of the arm (between state descriptions) are implicitly specified by an interpolation mechanism. We use the Hermite form of a parametric cubic function, which produces a smooth transition between two points based on their positions and the velocities at both endpoints (Foley & van Dam, 1982). Because a motor schema explicitly represents arm positions at only a few selected points over the course of a movement, we refer to a schema as *sparse* with respect to time.

More formally, we define a motor schema as a sequence of states,  $(S_1, S_2, \dots, S_n)$ , where each state  $S_i = (t_i, \{\{J_k, p, \dot{p}\}, \dots\})$  contains a time value  $t_i$ , and a set of 3-tuples. The states,  $S_i$ , are ordered such that the time values,  $t_i$ , are in an increasing sequence:  $t_i < t_j$  for  $i < j$ . Each 3-tuple contains: a joint name  $J_k$ , which identifies the joint described by the 3-tuple; a position  $p$ , which is the intended position of the specified joint at time  $t_i$ ; and a velocity vector  $\dot{p}$ , which describes the desired velocity of the joint upon reaching the position  $p$ . Each state contains a *set* of such 3-tuples, each of which describes one of the effector's joints, although not all joints need be specified.<sup>6</sup> The exceptions are the first and last state descriptions in the schema, which must specify a 3-tuple for every joint.

Figure 4.1 gives a pictorial example of a movement and a schema. The movement in Figure 4.1(a) shows the position of the arm at equal time slices or snapshots during the course of the movement. Tightly packed arm positions correspond to slow velocities, whereas more loosely spaced positions indicate higher speeds. Note that the movement shows the position of the arm at every time during the movement (with respect to the granularity of the simulation). In contrast, motor schemas specify arm positions only at a few times during the course of a movement. This can be seen in the schema shown in Figure 4.1(b), which *represents* the movement shown in Figure 4.1(a) but

6. In this chapter we do not utilize this capability. In general, the information for each of the joints may not be available initially and so we have designed our representation to handle such situations.

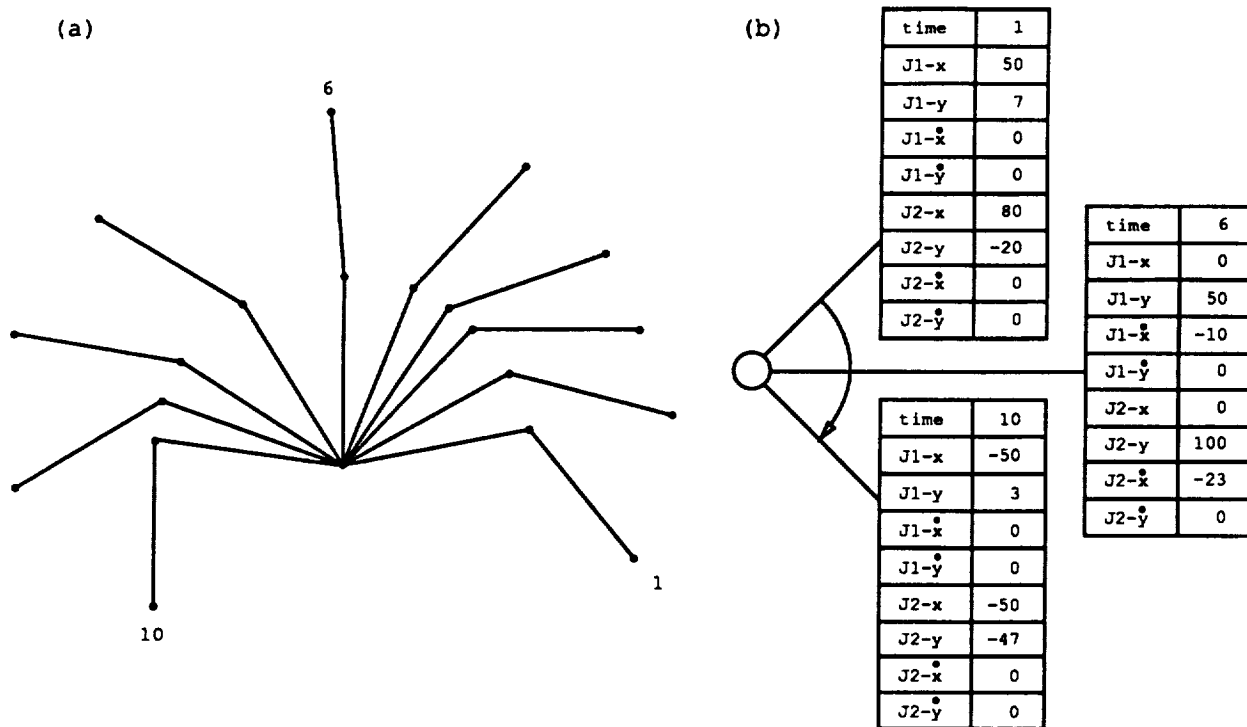


Figure 4.1. Pictorial representation of (a) a movement and (b) a motor schema.

only *specifies* information for the arm three times. In our framework, movements and schemas are closely related. In Section 4.3.1 we will discuss the parsing mechanism that takes a movement and returns a schema based upon that movement.

The representation used here derives its flexibility for both recognition and generation of movements by way of alternate formats used to specify joint information. The positions and velocities of the joints as given in the 3-tuples can be represented in either viewer-centered or joint-centered coordinates. Because these two formats are based upon differing coordinate systems, they give rise to two types of schemas that lend themselves to different performance tasks. In this chapter we are mainly concerned with *viewer-centered* schemas, but in the next chapter we focus on *joint-centered* schemas, which are used by MÆANDER to generate movements.

A viewer-centered schema represents the position and velocity vectors using Cartesian two-space coordinates with the origin centered at the agent. For the purposes of this chapter, the center of an agent will always be located at the base of its arm. Therefore, in a viewer-centered schema, the first 3-tuple (describing joint  $J_0$ ) would specify the  $x$  and  $y$  coordinates at the end of the first arm segment (actually the location of joint  $J_1$ ) relative to the origin located at the base (or joint  $J_0$ ). Similarly, the information stored at each joint  $J_i$  reflects the position and velocity of joint  $J_{i+1}$  relative to the base at joint  $J_0$ .

The viewer-centered representation gets its name from the source of this information — the agent's visual sensors. These can be thought of as generalized world sensors: anything that lets the agent observe objects and their positions relative to the agent's current location. In the case

of a more complete agent, one can imagine other origins for a viewer-centered schema, such as the agent's eyes. The choice of origin and axes should not affect the behavior if we assume a linear translation from the chosen origin to the base of the effector. This translates any given viewer-centered representation into our canonical viewer-centered representation.

#### 4.2.2 Probabilistic State Descriptions

When motor schemas are combined to form abstractions or generalizations, one can think of the resulting structures as *concepts*. In order to represent multiple instances with a single item, the values representing a movement must somehow be relaxed. One way to represent concepts in this type of model is to use probabilities (Smith & Medin, 1981). In the previous section we described a motor schema as a sequence of states in which each of the states contained specific values describing the set of joints. Here we introduce the *skill concept*, which represents both specific and abstract schemas in memory. Each skill concept consists of two components, a viewer-centered schema and a joint-centered schema. These two components have their own internal structure and have an associated conditional probability of occurring given an instance of the skill concept. The schema components are structured as described above, but each specific value has been replaced by a normal probability distribution defined by a mean and a variance. Additionally, each state description in the schema has a conditional probability of occurring given an instance of the specific schema type within the skill concept. That is, a state has a certain probability of appearing in a given schema and, if it does, then the values for its time and joint positions each have associated probability distributions. Likewise, the given schema has a certain probability of appearing for a given concept. Our notion of skill concepts is quite similar to our original description of motor schemas, except that there are two schema types for a single concept and each value in a state description is replaced by a mean and a variance. Note that nothing prevents one of the schema types in a skill concept to be unused or empty. Therefore, a skill concept can be either a very specific motor schema with minimal variance (a schema representing a single movement), or a more abstract entity with both viewer-centered and joint-centered schemas, each having values with high variance (a schema representing many movements). In further discussion, we will simply use the term viewer-centered and joint-centered schemas to refer to the appropriate component of a skill concept.

In general, concept formation systems may use discrete (nominal or ordinal) attributes or continuous (real-valued) attributes. In this dissertation we will only consider continuous, real-valued attributes, since we describe the positions and rotations of joints numerically. However, OXBOW has been implemented to allow either nominal or continuous attributes. Whether discrete or continuous attribute values are used to describe the joints, the information can be represented with a probability distribution. The only difference between the two cases is that in the nominal case the probabilities are stored explicitly for each possible value of a given attribute, whereas in the continuous case, the observed data are summarized as a normal distribution (using the mean and standard deviation of that distribution). This is a common assumption in work on concept formation (Fried & Holyoak, 1984; Cheeseman et al., 1988; Gennari et al., 1989; Anderson & Matessa, 1991).

In the following sections we discuss in detail how OXBOW acquires and uses skill concepts, focusing on using viewer-centered schemas to observe and recognize another agent's movement. In the next chapter we briefly describe how skill concepts containing both viewer-centered and joint-centered schemas are used to *generate* movement. Thus, our representation can be used for both recognizing a movement and monitoring the progress of a self-initiated movement.

### 4.2.3 Memory Organization

We have introduced a representation for movements that we refer to as the *motor schema*. However, in order to access or retrieve stored schemas, they must be organized in some consistent manner that facilitates efficient access according to some retrieval mechanism and that fares well with respect to representational economy. Here we describe the organization used to store these schemas in long-term memory.

In OXBOW, knowledge about movements is organized into a *hierarchy* of skill concepts. Nodes in this hierarchy are partially ordered according to generality, with concepts lower in the hierarchy being more specific than their ancestors above them. Thus, the root node summarizes all instances that have been observed, terminal nodes correspond to single instances, and intermediate nodes summarize clusters of observations. Fisher and Langley (1990) review arguments for organizing probabilistic concepts in a hierarchy.

Figure 4.2 shows a possible hierarchy for observed baseball pitching schemas.<sup>7</sup> This represents the memory of an agent that has experienced a *sidearm* pitch and three overhand throws — a *fast-ball*, a *curve-ball*, and a *fork-ball*. The leaf nodes of the tree in the figure represent the motor schemas from specific observed pitches. However, instead of simply storing the observed values, these values become the means (with a very small standard deviation) for the most “specific” concepts in the hierarchy. The node labeled *overhand* represents a generalization of the three specific throws stored below it in the hierarchy. This generalization is also a motor schema, but instead of specific values, the generalization contains means and variances for each attribute in its state descriptions. The higher variance makes the representation more abstract than a motor schema resulting from a single observed movement, in that more instances will readily match an abstract concept than a specific one.

Recall that our skill concepts consist of two components — a joint-centered schema and a viewer-centered schema. These schemas can be thought of as components of the entire skill. Furthermore, recall that a motor schema consists of a sequence of state descriptions. These states can, in turn, be thought of as components of the motor schema. It is important to note that this representation of skills is structural in nature. In particular, the sequential representation of state descriptions imposes a structure based upon *temporal* relations, as opposed to the more traditional *spatial* relations in the context of PART-OF hierarchies. This structural nature of skills and schemas significantly complicates the concept formation task as it is commonly conceived.<sup>8</sup> As a further complication,

7. Figure 4.2 only shows a conceptualization of the skill concepts without any joint-centered schemas present. Keep in mind that there would at least be place holders if no joint-centered information was available

8. For one conceptualization of the concept formation problem in the context of structured representations, see Gennari et al. (1989). Thompson and Langley (1991) present another approach to solving this extended problem.

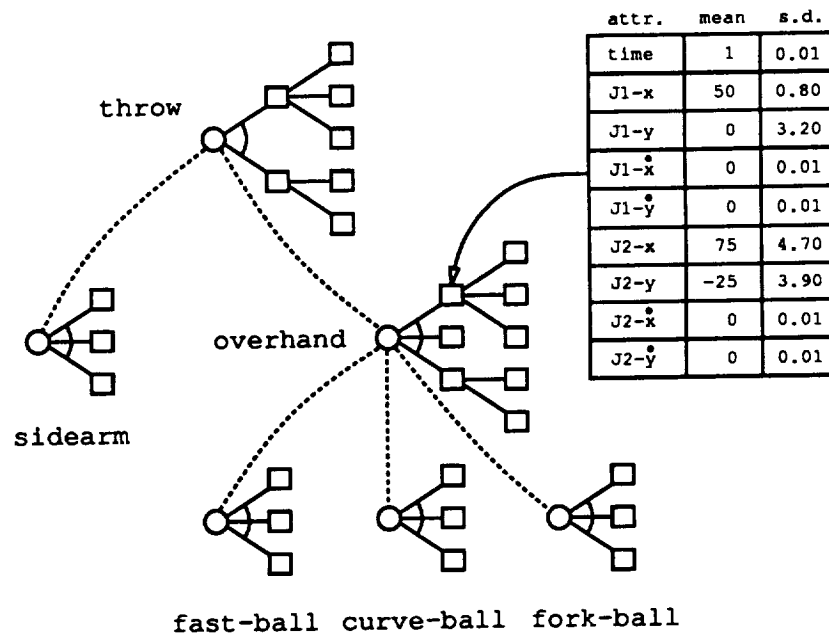


Figure 4.2. An IS-A hierarchy of schemas for "throw" movements, along with a portion of their internal structure.

there may be a variable number of states in a given motor schema. In Section 4.4.2 we discuss our response to these issues, but for now one needs only to understand the structural nature of our representation for skills and motor schemas.

The way OXBOW stores and organizes state descriptions introduces an additional hierarchy of *state* descriptions. Earlier we said a node in the skill hierarchy represented a movement concept that generalized some set of motor schemas. Now let us add that within the node, the state descriptions comprising the motor schemas are organized into their own IS-A hierarchy of state descriptions. Thus, each schema in a skill concept of the main hierarchy has its own private *state* hierarchy. The top level of this hierarchy represents the PART-OF relations between each state and the schema as a whole. That is, the set of classes at the top level of the state hierarchy will be the state descriptions comprising the motor schema and will be ordered according to the values for the time attribute in the respective nodes.

Figure 4.2 shows the node in the skill hierarchy corresponding to overhand throws in slightly more detail (again, only for the viewer-centered information); the other nodes in the hierarchy are similarly represented but we have not attempted a complete presentation of the memory structures for the purpose of clarity. The root of the internal hierarchy of state descriptions is stored at (but is distinct from) the skill concept that the state descriptions represent. This tree of state descriptions captures the structure of the abstract schema, and the time values stored in the state descriptions determine the temporal ordering. The figure shows the internal nature of one node in the hierarchy of state descriptions within the viewer-centered schema of the overhand node. The mean and standard deviation for each of the attributes correspond to the first node in the three "overhand" schemas. Remember that each node in the hierarchy of motor skills consists of two components,

both of which have their own internal hierarchies of state descriptions analogous to the one shown for the viewer-centered schema of the overhand skill concept.

### 4.3 Recognizing a Movement with Oxbow

As previously described, OXBOW's performance task is to recognize an experienced movement in the environment according to the current knowledge base of movements. The recognition process can be broken into two sub-processes — parsing the motion and classifying the resulting parsed structure. This section describes these performance aspects of OXBOW, while the next section will focus on the learning methods used to modify and update the knowledge base in response to experience. As we mentioned in the introduction to this chapter, learning and performance are closely tied in our view of concept formation. We separate them here only for the sake of presentation.

#### 4.3.1 Parsing a Movement

A movement is a continuous experience over some period of time. In order to understand a movement, OXBOW breaks the continuous experience into a set of discrete representations. This results in a sequence of snapshots of the environment (specifically the arm) over the course of a particular movement. Recall that our motor schema representation for movements stores only a few points in time for a given movement. The movement *parser* is responsible for selecting the points that are to be used for recognition and remembering.

We base our parser on Rubin and Richards' (1985) theory of elementary motion boundaries. They propose four primitive motion boundaries: starts, stops, steps, and impulses. The first two, starts and stops, represent zero crossings in velocity and are obvious choices for boundary points, since without them it would be impossible to distinguish a period of movement from a period of rest. The second two, steps and impulses, refer to discontinuities of force. However, as Rubin and Richards state, this set of motion boundaries is insufficient to represent many of the movements that we are interested in for this work. We have augmented these elementary boundaries with an additional boundary that represents zero crossings in acceleration. This gives us the desired representational power at the expense of additional motion boundaries or states in a schema.

Given the boundaries defined above, we must still define how these are used to parse a given movement. The agent observes a movement (in discrete time slices as described in Section 2.1) and maintains current values for position, velocity, and acceleration for each of the joints in the arm. Whenever the value for either velocity or acceleration at any one of the joints in the arm has a change in sign, the position and velocity information for all the joints<sup>9</sup> is collected and formed into a state description as specified in Section 3. Over the course of a movement, these boundary states

---

9. This represents a simplification on our part. Alternatively, we could store only the information for the joint that triggered a break point. Although our representation handles this, our implemented mechanisms would become rather more complicated.

Table 4.1. OXBOW's algorithm for movement classification.

---

```

Classify(movement, skill-node)
If leaf(skill-node) or recognized(movement, skill-node),
  Then return skill-node;
Else for each child of skill-node,
  Compute a score for Incorporating(child, movement).
  Let best be the child with the largest of these scores.
  If the score for putting movement by itself
    is greater than the score for best,
    Then return the skill-node;
  Else Classify(movement, best).

```

---

are identified, generated, and collected. At the completion of the movement, the resulting sequence of states is returned as a single motor schema.

Note that the entire movement is parsed and that it is the resulting schema that is given to the classification mechanism for recognition. Theoretically, it would be possible (and perhaps desirable) to have the parser and classification mechanisms working more hand in hand. That is, as each boundary is observed and the associated state is generated and appended to the end of the partial schema, this partial schema could be classified. This could conceivably lead to advantages in constraining the work necessary for later classifications of the more complete schema. We leave this as an idea to pursue in future work.

#### 4.3.2 The Classification Mechanism

Table 4.1 presents the basic OXBOW classification algorithm. At this level of abstraction, the classification process is no different from that used in Fisher's (1987) COBWEB and Gennari et al.'s (1989) CLASSIT. In these concept formation systems, the processes of classification and hierarchy formation are tightly coupled. We have separated these two components to provide a different perspective on this algorithm.

Upon encountering a new instance  $I$ , the system starts at the root and sorts the instance down the hierarchy, using an evaluation function (described below) to decide which action to take at each level. The termination condition of this recursive algorithm corresponds to the instance already having been recognized. This can occur in two cases: the current node may be a leaf in the concept hierarchy, or the evaluation function may consider the current node to be close enough to the instance that no further descent is necessary. The latter case requires the use of a *recognition criterion*; as described in Gennari (1990), this parameter determines when the system "recognizes" an instance and is especially useful in noisy domains.

At a given node  $N$  where the instance  $I$  is still unrecognized, OXBOW retrieves all children and considers placing the instance in each child  $C_k$  in turn; it also considers the case where the instance would be treated as a separate child. The algorithm uses its evaluation function to determine which

of the resulting partitions is “best”,<sup>10</sup> and then continues either by recursively classifying with the chosen best or stopping and returning the current node as the classification of the new instance.

More specifically, if the instance  $I$  is sufficiently different from all the concepts in a given partition according to the evaluation function,  $I$  is considered to be a member of a new category and no further classification is necessary (or useful). The current parent class is returned as the label of the new instance. The classification process halts at this point, since the new node has no children.

### 4.3.3 OXBOW's Evaluation Function

We have mentioned that OXBOW uses an evaluation function to determine the appropriate branch to sort new instances down during classification. Since a major goal of concept formation is to let the agent categorize new experience and make predictions, the system employs *category utility* — an evaluation function that attempts to maximize predictive ability. Gluck and Corter (1985) originally derived this measure from both game theory and information theory in order to predict basic-level effects in psychological experiments, and Fisher (1987) adapted it for use in his COBWEB model of concept formation. The measure assumes that concept descriptions are probabilistic in nature, and it favors clusterings that maximize a tradeoff between intra-class similarity and inter-class differences.

One can define category utility as the *increase* in the expected number of attribute values that can be correctly predicted, given a set of  $K$  categories, over the expected number of correct predictions without such knowledge, normalized by the size of the partition. This expression was originally designed for nominally valued attributes and summations of probabilities of attribute values. As used by COBWEB, these probabilities were computed from stored counts of attribute values.<sup>11</sup>

OXBOW works with continuous attributes, and the original expression for category utility had to be modified for such domains (Gennari et al., 1989). For such attributes, probabilities are computed by assuming a normal distribution of values and finding the standard deviation over observed instances. More precisely, category utility for continuous attributes is

$$\frac{\sum_k P(C_k) \sum_i \frac{1}{\sigma_{ik}}}{K} - \sum_i \frac{1}{\sigma_{ip}}, \quad (1)$$

where  $P(C_k)$  is the probability of class  $C_k$ ,  $K$  is the number of classes at the current level of the hierarchy,  $\sigma_{ik}$  is the standard deviation for an attribute  $i$  in class  $C_k$ , and  $\sigma_{ip}$  is the standard deviation for attribute  $i$  in the parent node.<sup>12</sup>

However, this expression assumes that every class consists of a simple list of attributes. For OXBOW, we must extend this to consider classes made up of two components, a joint-centered and

10. This lets the system avoid the need for an all-or-none match between the nodes in a given partition and a new instance being classified.

11. See Fisher and Pazzani (1991) or Thompson and Langley (1991) for more details and discussion of COBWEB's category utility equation.

12. As discussed in Gennari et al. (1989), the value of  $1/\sigma$  is undefined for any concept based on a single instance. We adopt their solution of using an *acuity* parameter, but we are not greatly concerned with its value. See Gennari (1990) for empirical analysis of the impact of this parameter on performance.



a viewer-centered schema. Each schema, in turn, consists of a set of components or, in this case, state descriptions. We break this into two parts; the first equation describes the score attributable to a particular schema, and the second calculates the total score over both schemas for all the skill concepts in a given partition of the main hierarchy. For the first part, the information in each component is weighted by the probability of that component, because the number of states is not the same for all schema instances. The partial category utility score of a viewer-centered schema  $m$  stored as part of a skill in the hierarchy is given as

$$\Psi_{VC}(m) = P(m) \sum_j^J P(S_{mj}) \sum_i^I P(A_{mji}) \frac{1}{\sigma_{mji}} \quad , \quad (2)$$

where  $P(S_{mj})$  is the probability of the  $j$ th state description of the viewer-centered schema  $m$ . This is the proportion of all state descriptions from schema instances stored in  $m$  that are locally stored under the state description  $S_{mj}$ . The term  $P(A_{mji})$  is the conditional probability of seeing the  $i$ th attribute given a state description in state  $S_{mj}$ . The leading term  $P(m)$  is simply the probability of the schema itself occurring given the skill concept to which it belongs. The score  $CU_{JC}$  for the corresponding joint-centered schema is similar and is not shown here. Given this expression, we may compute the overall category utility for a partition of the skill hierarchy as

$$\frac{\sum_k^K P(C_k)(\Psi_{VC}(C_{kvc}) + \Psi_{JC}(C_{kjc}))}{K} - (\Psi_{VC}(R_{vc}) + \Psi_{JC}(R_{jc})) \quad , \quad (3)$$

where  $P(S_{kj})$  is the probability of the  $j$ th state description in class  $C_k$ , or the proportion of all the state descriptions from schema instances of node  $C_k$  that are classified at state description  $S_{kj}$ . The probability  $P(S_{pm})$  is similarly defined for the  $m$ th state description in the parent of the current partition.

## 4.4 Learning from Unsupervised Experience

In our introduction to this chapter, we introduced a learning task associated with the recognition of motor schemas. To review, the task involves incorporating a newly experienced movement and parsed motor schema into long-term memory in such a way that one can more accurately recognize similar movements when they are presented in the future. In Chapter 6, we define exactly what we mean by “more accurately” and present some experimental results that support our claim that OXBOW accomplishes this learning task. We begin this section by describing the learning algorithm at a high level, at which the system borrows many ideas from Gennari’s CLASSIT and Fisher’s COBWEB. Then we proceed to the details of incorporating new movements into an existing schema class; this is where OXBOW makes some important extensions to previous work.

### 4.4.1 The OXBOW Learning Algorithm

Table 4.2 provides a brief description of OXBOW’s learning algorithm. Again, because learning and performance are integrated, the learning algorithm looks similar to the classification algorithm

Table 4.2. OXBOW's learning algorithm.

---

```

Build-Tree(movement, skill-node)
If leaf(skill-node) or recognize(skill-node, movement),
  Then halt;
Else for each child of skill-node,
  Let best be the result of
    Incorporate(child, movement) with the best score.
  Let second be the result of
    Incorporate(child, movement) with second best score.
Compare four cases, letting selected-child be the best of:
  best;
  by-itself;
  merge(best, second);
  split(best).
If selected-child is by-itself,
  Then halt;
Else Build-Tree(movement, selected-child).

```

---

introduced earlier. The primary difference is that the system makes permanent changes to memory structures when learning, whereas the original memory structure is retained for future use when classifying. There are some subtle differences as well. As with classification, the system considers incorporating the new instance in each of the existing children of the current node, as well as creating a new child with the single instance. If the instance *I* is sufficiently different from all the concepts in the current partition, a new singleton class is created containing *I*. In this case, the learning procedure halts since the new node has no children. However, when learning, the instance must be permanently incorporated into the nodes of the schema hierarchy along the path from the root to the leaf where the instance is finally placed.

Sometimes peculiarities in the order in which movements are observed can lead to an "incorrect" hierarchy structure. For example, this can occur when, after forming two classes of movements based on experience, the system observes several new instances that at first appear to be minor variants of one of the two existing classes. However, as OXBOW gains additional experience, it becomes clear that this "variant" is actually a distinct class representing a separate movement concept. In such cases, the concept formation system should be able to gracefully recover from previous errors. Therefore, in addition to comparing the result from incorporating instance *I* into the best of the current children and creating a new singleton class containing *I*, OXBOW considers two alternative actions. One involves combining the two best existing children into a single node. In this case the subtrees are spliced together such that the new node's children are the union of the children of the best and second best nodes. This new combined node is evaluated within the partition with the remaining nodes. The other alternative replaces the best child with its constituent subtree branches. That is, all the children of the best node are promoted and become direct children of the current node, and the best node disappears.

These final two actions are referred to as *merge* and *split* operations. They are intended to aid the system in recovering from poor choices earlier in training, perhaps due to order effects. Fisher and

Pazzani (1991) argue that some form of "backtracking" operators are necessary for any incremental learning system, particularly in pure hill-climbing systems such as OXBOW, which can get stuck at local optima. One can imagine additional operators that would take larger steps through the space of possible partitionings. However, these amount to some sequence of applications of the simple merge and split operators. This does not mean that such compound operators may not be necessary in order to find an ideal concept hierarchy (a learning evaluation issue), but they are not necessary in theory.

We now turn to the largest difference between OXBOW and concept formation systems such as CLASSIT – the instance incorporation process. This difference arises due to the structural nature of our motor schema representation.

#### 4.4.2 Incorporation of Motor Schemas

Every concept formation system must address the question of how to incorporate a new instance into an existing class. This is the essence of learning in these systems. An evaluation function can be used to determine which node, out of a set of candidates, should have the instance incorporated. But the incorporation process actually changes the memory structures and lets one make predictions from the stored information.

In Fisher's COBWEB, incorporating a new instance was a simple matter of updating the counts associated with a class node based on the attribute values occurring in the instance. The system assumed that each instance had a fixed set of uniquely labeled attributes, although an instance could omit a value for a given attribute. Gennari et al.'s CLASSIT (1989) extended this approach to include a notion of structured objects made up of multiple components. These objects were restricted to a single level of components, where each component was a primitive object analogous to the objects given to COBWEB. Also, CLASSIT assumed that each structured object had the same number of components and that each component occurred in each instance. That is, the structure of these objects was uniform across all classes and did not vary.

Here we are interested in forming concepts of movement representations (as defined earlier), and neither COBWEB nor CLASSIT has satisfactory mechanisms for handling this task. Recall that a skill concept consists of two components – a viewer-centered schema and a joint-centered schema. This much structure could be handled by CLASSIT as described in Gennari et al. (1989); one simply provides the correct mapping, since there are always exactly two. However, recall that a motor schema is also a compound object made up of components that represent the states of the arm at specified time values. Each state satisfies the notion of a primitive object since we are mainly dealing with the parts of the arm; each joint contributes its own unique attributes to the total description of a state.<sup>13</sup> However, because a motor schema may have any number of state descriptions, there may not be a one-to-one correspondence between two schemas' states. Therefore, one cannot uniquely associate the attributes (at the state description level) from one schema to another.

---

13. Of course, one could think of each state again as a compound object made up of components corresponding to the parts of the arm. This goes beyond the scope of the current research and we leave it to future work.

OXBOW includes a solution to this state correspondence problem that is specific to temporally structured domains, but that allows more flexibility than the one implemented in CLASSIT. Both systems must determine mappings between components in an instance and components in a stored concept. However, OXBOW can combine an instance and a concept with differing numbers of states by allowing multiple states in the instance to map onto a single state in the concept, and by allowing individual states in the instance to become new and separate states in the concept. The category utility scores for incorporating single states from the instance into the hierarchy of state descriptions determines the mapping between the instance and the concept. This method is clearly more flexible and (we believe) more elegant than CLASSIT's, although both methods have the same  $O(n^2)$  computational complexity, where  $n$  is the number of components in the concept.

For example, suppose OXBOW observes a movement that is parsed into a schema having three states. In the process of incorporating this schema into the memory presented in Figure 4.2, the system must consider including it as an overhand schema. This involves establishing the mapping between the states in the observed movement and those in the schema node. The general solution applied here is to use category utility as an evaluation function for determining how to match states from respective schemas with each other and for deciding when to leave states unmatched (in the case where category utility prefers creating a new disjunct). This application of category utility is based upon treating each state of a new schema to be incorporated as a separate instance in and of itself. However, instead of passing each state down through the hierarchy of motor schemas, they are passed through the separate PART-OF hierarchy within the schema node under consideration.

More specifically then, for a given state,  $S$ , and a hierarchy of states associated with a node in the hierarchy of schemas, we execute the same learning algorithm as described in Table 4.2 with the following differences. First, at this level "incorporate" simply involves updating all the attribute counts, means, and variances for the given state. State descriptions can be thought of as primitive objects with a fixed set of attributes that can be compared between states. Second, the evaluation function used is a simplified version of category utility. Because its goal is to capture the temporal structure present in the data, OXBOW only considers the time attribute in determining the score, instead of summing over all the attributes.<sup>14</sup> The resulting form of the equation is

$$\frac{\sum_k P(C_k) \sum_j P(S_{kj}) \frac{1}{\sigma_{kjtime}}}{K} - \frac{\sum_m P(S_{pm}) \frac{1}{\sigma_{mptime}}}{M}, \quad (3)$$

where  $\sigma_{kjtime}$  and  $\sigma_{mptime}$  are the standard deviations for the time attribute in the  $j$ th state of class  $C_k$  and the  $m$ th state of the parent, respectively. All of the attributes that describe a state are updated when a new state is incorporated, but only the time attribute is considered when evaluating the score for a node and its children. Also, notice that this form of the category utility equation applies to both of the internal hierarchies, one for viewer-centered states and the other for joint-centered ones.

The incorporation of a new schema instance effectively establishes a mapping among states. As each state in the new instance is considered individually, it is either "mapped" onto one of the existing states and is incorporated, or onto nothing and becomes a separate state by itself.

14. We have implemented the latter alternative as well and pilot studies show little difference in overall behavior.

Given this implicit mapping, we can ignore the structure of the schema and compute the score of the partition at the motor schema level. We remove the structural information by treating the attributes of each state's description as unique at the motor schema level. That is, a schema consisting of three state descriptions, each with 13 attributes, would have three times that many ( $3 \times 13 = 39$ ) unique attributes used in the calculation of category utility. This process is reflected in the additional nested summation in equation (2): sum over states, and for each state, sum over the state's attributes. In other words, states are classified only with respect to time, whereas schemas are classified with respect to all of the attributes.

Since schemas are composed of the first-level nodes beneath the root of the state hierarchy,<sup>15</sup> we may think of this hierarchy as representing the PART-OF structure for the schema. We believe this way of viewing concept hierarchies is one of the contributions of our work, and it is based on the insight that the first level of the tree reflects a *partition* of some outer environmental context. The COBWEB and CLASSIT systems use category utility to determine IS-A relationships between instances (and classes) and more general classes. OXBOW uses the same function to determine appropriate matches between parts of complex objects. Every instance processed by a concept formation system can be thought of as PART-OF the environment being addressed. That is, some agent or mechanism parses the world and hands "instances" to the learning agent one at a time to be incorporated. These instances are used to construct a concept hierarchy in which the children of every node share IS-A relations with the abstraction stored at their parent.

We are not claiming that the top-level nodes are parts of the generalization stored at the root of the hierarchy; likewise, the top-level concepts are not instances (specializations of an abstract description) of the outer context or environment. Instead, we claim that the top-level nodes are items that make up, or are parts of, the environment. Therefore, we have a PART-OF structure at the top-most level of the hierarchy *with respect to the environment from which the instances are observed*. In application to OXBOW, our claim is that the top-level nodes of a state hierarchy share PART-OF relations with the associated schema concept in which they are stored. For example, the first state in the internal hierarchy of the *overhand* node from Figure 4.2 is not PART-OF its parent in the state tree (the root node is not shown), which summarizes all the state descriptions of the *overhand* schemas. Rather, this state is PART-OF the *overhand concept*, which summarizes the skill concepts below it in the hierarchy of motor schemas (rather than state descriptions).

OXBOW takes advantage of this characteristic by creating hierarchies of states in which the top level provides the states to be used in the motor schema. This works out conveniently because motor schemas are presented as parsed structures consisting of a sequence of states. Although we do not propose our system as the final solution to learning structured concepts, we consider it satisfactory for our present purposes and the intended context of our work.

---

15. Lower levels of the state hierarchy are retained in case subsequent splits are necessary. They do not enter into the current argument.

## 4.5 Conclusion

In this chapter, we have presented a computational model of movement recognition. As we have stated in Chapter 1, a comprehensive model of motor learning should address both this task and that of movement generation. In the next chapter we present MAGGIE, the subsystem of MEANDER responsible for executing motor skills. MAGGIE generates movements using the joint-centered schemas alluded to earlier. The joint-centered schemas specify the desired behavior of the joints in terms of local rotations that, when executed, should correspond to the generalization of the movements observed and stored by OXBOW.

At one point in this chapter we alluded to an integration of parsing and recognition. Before moving on to a discussion of MAGGIE and movement generation, we want to summarize these thoughts. Interleaving the parsing and classification mechanisms would entail trying to recognize partial schemas before they were completely finished. Ideally, as the movement proceeds and more information is available, the classification process should gracefully adjust and make better recognitions. In our evaluation of OXBOW in Chapter 6, we take the first step toward this by testing the system's ability to classify partial schemas. Additionally, this would reduce the necessity of the motor buffer introduced in Chapter 3, as significant events or zero crossings could immediately be appended to the structure in short-term memory that is currently being classified.

We feel that OXBOW makes a number of important contributions. First, we have built a flexible representation for modeling movements. This representation allows the flexible recognition of newly observed movements, as well as the generation of movement behavior, as we show in the next chapter. Furthermore, the representation should be applicable to a wide range of motions. As we said earlier, this representation fills a gap between robotics, which generates movements with low-level models, and psychology, which employs high-level models but without complete computational mechanisms.

Second, we have uncovered an exciting duality between IS-A and PART-OF relations. The duality depends upon the context of the instances that are being observed by the concept formation system and the interpretation of the root node. An instance stored in the hierarchy IS-A member of the set of all experiences, but it is also a PART-OF the learning agent's environment, at least at some point in time. We are currently exploring the implications of this duality and believe that a more complete understanding of concept formation will result from this insight.

Finally, by exploiting this duality, we have been able to extend concept formation methods to a new class of domains. Although there has been some research in concept formation with structured data (Segen, 1990; Thompson & Langley, 1991; Stepp & Michalski, 1986; Levinson, 1985), most work has been restricted to instances described by simple attribute-value vectors. By using category utility on the nodes in the part-of tree, and therefore by establishing a labeling between states in a new instance and states in a stored motor schema, we have applied the concept formation ideas of COBWEB and CLASSIT to structured objects.

## Skill Improvement Through Practice

---

### 5.1 Introduction

By its very nature, skill is exhibited only through active performance. In the previous chapter, we focused on OXBOW, the component of MEANDER that builds the memory structures that represent observed movements. This is only the first part of developing a skill; the next part is performing the movements that correspond to the acquired skill. The memory structures acquired through observation let an agent *recognize* a particular movement as being similar to movements observed in the past. Additionally, they allow a quantitative evaluation of the accuracy of self-initiated movements. However, they do not provide the means for an agent to enact a particular movement.

In this chapter, we present MAGGIE, the second significant subsystem of MEANDER. We address the problem faced by an agent that has acquired a concept of a particular skill (as evidenced by recognition) but wishes to perform the skill. As we noted in Chapter 4, viewer-centered schemas are not executable structures. They are appropriate for recognizing visually observed movements, but they are not useful for manipulating the arm. MAGGIE controls the joints of the arm by specifying rotations in each joint's local (polar) coordinate system. Below we describe the *joint-centered* schema that represents such values. We also describe how a joint-centered schema is initially generated and how movements described by a joint-centered schema are actually executed. Recall that the schema only specifies the positions (joint angles) and velocities at a few time points during the course of a movement. We introduce the *motor program* as the executable structure that describes all the intervening positions of the movement.

When an agent manipulates an arm, the resulting movement may not turn out as intended. In MAGGIE, errors can result from starting with a poor initial joint-centered schema, from inherent variance in the mechanical system, or from external interference. In order to overcome any of these problems, MAGGIE has a simple mechanism for error correction. This mechanism uses simple closed-loop feedback control with the viewer-centered schema serving as the standard of reference. Thus, MAGGIE's performance task is to move the limb through a movement trajectory specified in a joint-centered schema; this involves obtaining a joint-centered schema, generating a motor program, running the program on an arm, and possibly checking for errors and correcting them.

As before, we want our model to exhibit improvements in performance over time. We are not just concerned with a performance task, in this case movement generation; we also want the agent's skill level to increase through practice. The learning task for MAGGIE, in the context of the performance component outlined above, is to improve the quality of generated movements as a result of experience or practice. In the next section we review the schema representation from the previous chapter and describe MAGGIE's joint-centered schemas. Then we describe MAGGIE's performance component, which operates upon these representations to achieve movements with the arm. In Section 5.4 we present the learning component that produces modified joint-centered representations and how it incorporates these changes into long-term memory.

## 5.2 Representations for Generating Behavior

In Chapter 4, we showed how motions were parsed into motor schemas and stored in memory. Before MEANDER can perform actions with its limbs, it must convert the stored schemas into a form that is compatible with the effector interface. Recall from Chapter 3 that this requires the specification of the arm's behavior at each simulated time slice. In this section we review the joint-centered schema and MAGGIE converts it into an executable form. We also review how viewer-centered and joint-centered schemas are associated and organized in long-term memory.

### 5.2.1 Joint-centered Schemas

Recall from the previous chapter that a motor schema consists of a sequence of states, in which each state describes the status of each of the joints in the arm at a specified time. Also, remember that the states were sparsely distributed (in time) across the duration of a movement. That is, a few points were satisfactory to describe a complete action. In particular, we introduced the notion of a viewer-centered schema, in which the positions and velocities at each joint are represented in a Cartesian coordinate system with the origin at the base of the arm. These viewer-centered schemas represent motions that were observed, and they allow recognition of similar movements.

In this chapter we describe the counterpart to the viewer-centered schema – the *joint-centered* schema – which is used for generating or executing movements rather than recognizing them. The structure is essentially identical to the viewer-centered schema, but the information stored within the schema is quite different. As before, each state in the sequence specifies the state of an arm (positions and velocities for each joint) at a particular time during the movement. In the viewer-centered representation presented earlier, the positions and velocities associated with given joint describe the movement of the end of the link that is attached to the joint. In a joint-centered schema, the positions and velocities of each joint refer to the state of rotation for the joint itself. More specifically, the position and velocity for a given joint in a joint-centered schema refer to the *rotation* and *rotational velocity* of the joint. These rotational values are given in local polar coordinates, where rotations are defined with respect to the y axis. This reference for each local coordinate system is a linear extension of the previous joints' link member as described in Chapter 3.



The joint-centered and viewer-centered schemas may be thought of as dual representations. That is, there exist well-defined<sup>16</sup> functions that convert one representation into the other in either direction. However, we will only be interested in converting from viewer-centered to joint-centered representations. Any realizable arm position can be described or represented in either of the formats. Although the viewer-centered information refers to the link end position (and velocity) and the joint-centered information describes each joint's specific rotation (and rotational velocity), the values are constrained by the lengths of the links in the arm. Because these are fixed in length and because the local coordinate system for each joint is based upon the previous joint's link, a straightforward transformation can convert one format into the other. Although both frameworks are representationally equivalent, each is better suited for some types of movements than for others. The different nature of compatible movements arises from the way *MÆANDER* treats motor schemas when extracting the movement from the schema. We discuss this treatment in more detail below.

Just as the viewer-centered schemas were motivated by the visual sensory system of the agent, the joint-centered schemas are motivated by the control mechanisms of joints. From psychological studies, we know that humans can move limbs to a specified location without any feedback, either visual or proprioceptive (Kelso, 1982). In robotics, artificial jointed limbs are controlled by specifying torques or voltages at each individual joint (Hardy, 1984). Joint-centered schemas specify the local rotations of each joint and are therefore appropriate when generating behavior. When dealing with artificial limbs (robot arms), it is regularly assumed that local joint control commands are given to the hardware level. These are typically voltage or torque values, but an analogy holds for velocities and positions. It is common for robotics problems to involve both a work space (our viewer-centered representation) and a joint space (our joint-centered representation). These factors motivate our dual representations of viewer-centered and joint-centered schemas.

The sparse representation of a motor schema seems plausible for storing motor skills in long-term memory, but to actually generate motor behavior, one must specify the missing points. We use the term *motor program* to refer to such a dense representation for a skill. A motor program can be viewed as the corresponding structure to an observed movement prior to parsing, as described in Chapter 4. It is important to distinguish motor programs from joint-centered schemas. The latter specify the rotations and velocities of joints only at selected times; in contrast, motor programs specify joint rotations at *every* point in time (with respect to the granularity of the temporal simulation). Such information can be generated dynamically from a joint-centered schema, as we discuss in Section 5.3.

### 5.2.2 Memory Organization in Review

As we discussed in the previous chapter, a skill concept is represented in memory as a pair of viewer-centered and joint-centered schemas. Each of these schemas, in turn, is represented as a hierarchy of probabilistic state descriptions (the internal state hierarchies within a skill concept). In Chapter 4, we focused on the hierarchy of viewer-centered state descriptions, but joint-centered schemas are stored in an identical hierarchy as part of a given skill concept. The joint-centered

---

16. As described earlier, we restrict rotations to be in the interval  $(-\pi, \pi)$ , thereby keeping a one-to-one mapping.

data are stored in state descriptions that are analogous to those containing the viewer-centered data. In this case, each state description has attributes for each of the joints representing the local rotations and rotational velocities, instead of the  $x, y$  position in Cartesian coordinates as used for viewer-centered state descriptions. When learning, both types of motor schemas in the skill concept can be accessed and manipulated independently of each other. Whenever a schema is to be executed, both the viewer-centered and joint-centered schemas are extracted from the skill concept.

This organization of the skill concept resolves the issue of establishing correspondences between a joint-centered schema and a viewer-centered schema. If the two schemas for a given skill were stored separately in memory, then we would have to propose a mechanism for linking them. Such a mechanism would create links between a joint-centered representation and viewer-centered schema that describes the desired movement for the joint-centered schema in question. Instead, we suggest that the information is stored together in a single node of the skill hierarchy. The representation we have chosen reflects the way we think of a skill as a single concept that contains (at least) two sets of data with two representations: one for recognition and feedback control and the other for execution. This organization bears obvious similarities to some psychological theories of motor control discussed in Chapter 2 (e.g., Schmidt, 1975b; Pew, 1974).

### 5.3 Executing Motor Skills in MAGGIE

We have stated our concern with generating accurate movements. In order to do this, MAGGIE must be able to use the representations constructed by OXBOW and those introduced in the previous section. A formal statement of the performance task is:

- *Given*: a viewer-centered schema describing a desired movement;
- *Move*: the limb through the trajectory specified in the viewer-centered schema.

This implicitly assumes that the intended limb is known (if there are more than one) and that the desired speed of execution is given. Again, the desired trajectory is specified by the viewer-centered schema which, along with the joint-centered schema, is extracted from the skill node that is selected for execution. MEANDER's performance system attempts to carry out this behavior using the specified limb. This involves a number of processes. First, the joint-centered schema must be 'run' by generating an executable motor program and carrying out the specified actions. Simultaneously, the agent may monitor the resulting states, comparing actual positions with the intended ones as given in the viewer-centered schema. In this case, execution and monitoring proceed in parallel until an error is detected. In the event of a detected error, the system invokes an error correction mechanism to return the limb to the desired path. Below we consider each of these steps in more detail.

#### 5.3.1 Retrieving the Joint-centered Schema

We assume that the viewer-centered schemas that MEANDER wants to execute have been acquired by observing another agent's actions, as described in Chapter 4. Naturally, if there is a joint-centered schema associated with the given viewer-centered schema, then it is used for generating

the movement. However, the first time a particular motor skill is performed there can be no joint-centered information present. One approach to obtaining this initial joint-centered schema is to apply an *inverse kinematic transform*<sup>17</sup> to the given viewer-centered schema. That is, the position of each joint  $J_i$  in Cartesian coordinates (with origin at the base of the arm) is converted to a rotation of the previous joint  $J_{i-1}$ . This reflects an offset correspondence between joints at the ends of links in the viewer-centered format and joints that have attached links in the case of joint-centered descriptions. The resulting rotation is based upon the position of this joint and all the previous joints back to the base of the arm.

Applying this transform to every state description in the viewer-centered schema would result in a complete joint-centered representation that can be directly executed. Unfortunately, this transformation must be done serially across the joints of a limb, making this a time-consuming computation. First the base joint must be evaluated and then each successive joint must be processed in turn. We choose to minimize our use of this transform by only applying it to the first state description of the given schema and only when there is no joint-centered information available at all. The result of transforming just the first state in the viewer-centered schema is a joint-centered schema that, when executed, will hold the arm motionless. That is, we assume that an arm is in place and ready to go (similar to meeting the preconditions of an operator) when a skill concept is retrieved for execution, but that the arm will stay still (except for error corrections described below) if no previous experience has informed otherwise.

### 5.3.2 Executing the Joint-centered Schema

Joint-centered schemas only specify the positions and velocities of the joints at selected points in time. Within our framework, the control of actual motor effectors requires the specification of the relative rotational velocities for every joint at every simulated time step. As described above, a *motor program* satisfies this requirement, since it specifies the respective joint positions for every time value. MÆANDER does not store motor programs in memory; the system creates them in real time as it executes the skill. This is accomplished by generating a spline for each joint between successive pairs of the states specified in the joint-centered schema.<sup>18</sup> During a movement, when the limb reaches the end of the spline segment between two state descriptions,  $S_{i-1}$  and  $S_i$ , the latter becomes the source and the next state in the sequence,  $S_{i+1}$ , becomes the target for the next spline. This method yields a smooth, continuous curve throughout the execution of the schema.

This process is the logical inverse of the parsing mechanism described in Chapter 4. Instead of taking a raw movement representation specifying arm states at every time step and producing a motor schema, the interpolation process takes a joint-centered schema and produces a motor program that specifies (joint-centered) arm states at every time. This process is also used to

17. This transform re-represents a state of the arm given in global Cartesian coordinates as a state described by local joint rotations for each respective joint. The details of this transformation are not important to this discussion, but they can be found in Wylie (1975).

18. We assume that low-level neural circuitry can take relatively sparse inputs from a schema and generate such a motor program in real time. Specifically, in MÆANDER we use a Hermite parametric spline that interpolates between two state descriptions with given velocities. This splining technique maintains smoothness in both position and velocity.

determine the trajectories of the desired movement specified by a viewer-centered schema. Instead of interpolating between joint angles, MAGGIE interpolates between Cartesian coordinates describing the positions of the joints in viewer space. However, interpolations in Cartesian space may result in joint positions that are physically impossible because the links of the arm are of fixed length. We define the arm state specified by the interpolation of a viewer-centered schema to be the positions of the arm if each of the links were "pointing" through the interpolated point. Mathematically, this amounts to the expression

$$(x, y) = (l \cos(\arctan(y'/x')), l \sin(\arctan(y'/x'))) ,$$

where  $l$  is the length of the link that is attached to the joint in question and  $(x', y')$  are the coordinates given by the spline function. For each subsequent joint, the resulting  $(x, y)$  position is used to adjust for the actual position of the previous joint.

Like the inverse kinematic transform, the process of generating the motor program is assumed to take some time. However, it is not necessarily a serial process and we do not consider it a bottleneck. In experiments with humans, a preparation period is observed prior to the actual movement of joints (Fischman, 1984). In MEANDER, we interpret this to correspond to the "set-up" time necessary to retrieve the schemas from the movement concept and to generate the motor program itself.

### 5.3.3 Monitoring the Progress of a Movement

At any stage of learning, there will typically be some discrepancy between the movements described by the viewer-centered and joint-centered schemas of a given skill concept. This is most pronounced before MEANDER has had an opportunity to practice movement (i.e., when there is no joint-centered schema). Thus, MAGGIE must have some means of detecting errors, and this is the role of the monitoring process. If we consider the viewer-centered information to represent MEANDER's notion of a desired movement, one can detect errors whenever the state of the arm (as controlled by the motor program during execution) diverges from the desired state given by the associated viewer-centered schema.

In order to detect deviations, MAGGIE compares the state of the arm during a movement execution to the description of the desired trajectory itself. In the present implementation, we only consider visual sensory feedback on the state of the arm.<sup>19</sup> This information is represented in viewer-centered coordinates. The desired trajectory is obtained by interpolating between the points given in the viewer-centered schema, as described above. This interpolated information is analogous to the motor program, but it is useless for actually controlling the joints of the limb. MAGGIE compares the information from these two sources when monitoring the execution and determines the difference, or error, between them. When the difference obtained from this comparison becomes noticeable (i.e., exceeds a parameterized threshold), the system does two things. First, the *failure point*, which describes the errors for each joint at the current time of comparison, is stored in a motor buffer for later processing. Second, MAGGIE invokes the error correction process with respect

19. Proprioceptive feedback is an additional source of information that would naturally benefit performance and that seems to be used in humans. We do not explicitly limit the feedback sources to visual senses.

to this failure point. This process (described in the next section) does not interrupt the ongoing execution but rather augments the movement already determined by the motor program.

A *monitoring frequency* parameter determines how often MAGGIE examines the ongoing movement. We have implemented MAGGIE to monitor on a regular cycle based on the setting of this parameter with a random offset from the start of the movement. However, nothing precludes the model from sometimes monitoring frequently (perhaps with novel skills) or not monitoring at all (in the case of highly automated skills). We envision a higher-level control module (outside the scope of this work) that would determine when to *attend* to sensory feedback.

#### 5.3.4 Correcting Detected Errors

Once MAGGIE detects a significant divergence from the desired trajectory, it must still recover from that error. When invoked by the monitoring process, the error recovery mechanism applies a "burst of force", or correction, in a direction that will reduce the size of the error. This process models the type of corrections that result from error detection at the brain level of the nervous system, and not corrections resulting from servomechanisms at the spinal level. That is, we think of these corrections as purposeful responses to recognized errors during the course of a movement.

The nature of the correction is determined by the observed error and two system parameters. We use an inverted U-type correction based on the absolute value function, which causes a gradual change in the limb's actual movement over the lifetime of the correction process. The *correction magnification* parameter controls the size of the generated correction (relative to the size of the error) and the *correction duration* parameter controls the length of the correction in simulated time. In the default condition, the magnification factor is set at one; in this case the area under the curve is the same as the amount of error detected and the duration parameter is set so that corrections are completed before another monitoring cycle begins. This means that if the trajectory specified by the motor program does not diverge further from (or get closer to) the desired trajectory, then the limb would be back at the desired position at the end of the correction. However, if the arm behavior was converging with the desired trajectory, then the correction adjustment will cause the arm to overshoot. Likewise, if the error is getting worse, then the correction will be insufficient to bring the arm back to the desired path. Such cases require multiple calls to the error recovery process.

Accessing the visual sensory buffers, performing the comparison with the desired trajectory, and determining the type of response (if any) all take some amount of time. In humans, the minimum cycle time from error in the environment to initiation of corrective measures is approximately 200 msec. Although implemented as a parameter, the *error-correction delay* determines the *granularity* of our simulation. That is, the length of a simulated time step is determined by dividing 200 msec. by the error-correction delay. It is important to understand the distinction between this delay and the monitoring frequency introduced above. The latter controls how often MAGGIE checks for errors, whereas the former determines the time from an error's detection to the beginning of its correction.

Taken together, monitoring and error correction make up a relatively basic and straightforward closed-loop feedback mechanism. We have mentioned some of the parameters that impact this

mechanism's behavior: the sensitivity of the system to divergences, the frequency of checking for errors, the duration of error corrections, and the magnification of corrections for a given sized error. The particular settings of these parameters are not important to the theory implemented as MEANDER, and we will show in Chapter 7 that the behavior of the system is relatively robust with respect to a range of settings for these parameters.

When movements, even novel ones, are performed slowly enough, monitoring and error correction allow a near perfect reproduction of the desired movement. However, not only do agents sometimes need to perform movements quickly, conscious monitoring and error correction consumes cognitive resources that might better be spent on other processes. Therefore, there is great incentive to improve the representation of the joint-centered schema so that the path will more closely approximate the desired trajectory even *without* monitoring and error correction. This is the job of MAGGIE's learning component.

#### 5.4 Learning from Execution Errors

Let us reiterate the learning task we are addressing. For a given motor skill present in memory, MEANDER should improve its ability to perform the movement through practice. Any improvement should be independent of monitoring and error correction. That is, an improved representation must yield superior performance whether or not the system monitors for errors and corrects them. In MAGGIE this is accomplished by modifying the joint-centered schema according to information from a recent execution so that its behavior diverges less from the associated viewer-centered schema the next time it is executed. As a whole, MEANDER employs two interacting learning mechanisms to improve its joint-centered schemas. In this section we describe these mechanisms.

Improvement through practice in MAGGIE is more active than simply incorporating movement after movement into long-term-memory. In the previous chapter, we considered the OXBOW subsystem, which carries out pure unsupervised learning; its task is to construct summary descriptions of movements that it has observed. In MAGGIE, learning occurs in a *self-supervised* manner (Sammut & Banerji, 1986; Langley, 1985; Mitchell, Utgoff, & Banerji, 1983). There are two parts to directed experience: the first involves determining when to learn and the second concerns determining what to learn. These issues, addressed by all supervised learning systems, are discussed in the remainder of this section.

We have seen that error detection invokes the error recovery process, but it also triggers learning. Whenever the path of a joint diverges noticeably from the desired path, the failure point is temporarily stored in the motor buffer. This lets MAGGIE delay learning until after the execution has been completed. Table 2 presents the model's basic learning algorithm. Since a number of errors may occur in a given trial, the first step involves selecting a failure point from which to learn. MAGGIE selects that failure point in the motor buffer with the largest error. Thus, larger errors are reduced before smaller ones. Once MAGGIE has selected a failure point, it applies a set of critics that generate candidate replacement motor schemas. The system evaluates these candidates and selects one as the best revision. This far, MAGGIE has improved the joint-centered schema in question, but it has no memory to remember this improvement. Therefore, OXBOW is used to

Table 5.1. MAGGIE's schema revision and learning algorithm.

---

```

Modify-Schema(joint-schema, viewer-schema)
Let fail-point be the largest error from the motor buffer.
Let new = applying(velocity-critic, fail-point, joint-schema).
Find the percentage improvement over the current form of the schema.
If improvement(new, jc-schema) < bias,
    Then let new = applying(add-point-critic, fail-point, joint-schema,
Call OXBOW with new and viewer-schema.

```

---

incorporate the new schema structure in MÆANDER's long term memory of movement concepts. We now consider each of these steps in more detail.

#### 5.4.1 Monitoring and Opportunities to Learn

Every learning system must address the issue of determining when to learn. OXBOW and many related unsupervised learning systems (Fisher, 1987; Gennari, 1990) learn from every instance that is presented, unless it is specifically presented as a test instance. In MAGGIE, as in a number of supervised learning systems (Iba, Wogulis, & Langley, 1988; Aha, 1990), this is not the case; learning occurs as the result of detected errors during the execution of a skill. That is, the monitoring process provides the opportunities for MAGGIE to improve its representation of a movement skill.

As already mentioned, the failure point from the memory of corrections is selected for further processing. This seems plausible in so far as the largest errors receive the most processing and therefore should decay the least rapidly (Massaro, 1975). That is, limitations on memory access constrain the types of learning that take place in humans (and therefore in MAGGIE). Although MAGGIE retrieves the largest error, we do not require this as part of the model. Alternative schemes could be based on recency or primacy, as long as only a single event is recalled and processed further by the learning component.

Thus, MAGGIE focuses on the largest error detected for a given movement skill. Note that this implies that the current level of quality for the given joint-centered schema determines the error information that will be available to the learning process. In this way, the opportunities for learning within a single movement concept are constantly changing as MAGGIE's skill at the concept improves. This approach to determining when to learn implicitly selects the information that determines what to learn.

#### 5.4.2 Critics and Modified Motor Schemas

Determining what to learn essentially involves deciding how to modify a particular representation such that future performance will be improved. To accomplish this, MAGGIE employs a set of *critics* similar in principle to those used in HACKER (Sussman, 1975). The critics are responsible for constructing candidate joint-centered motor schemas based upon the motor schema that was originally executed and the largest error detected during execution. Strictly speaking, the critics

do not really affect the long-term memory structures.<sup>20</sup> Instead, as we will see shortly, one of the candidate schemas is selected and given to OXBOW to be incorporated into the hierarchy of movement skills, thereby modifying memory. The learning operators (the critics) are responsible for constructing effective revised motor schemas, and it is OXBOW's responsibility to see that they are stored appropriately and can be remembered in the future.

Theoretically, there is no limit to the number of critics that could function simultaneously, each producing its own candidate. However, recall that MAGGIE specifies a motor schema as a sequence of states, each describing the locations and velocities of a set of joints. This suggests two natural approaches to modifying joint-centered schemas:

- modifying one of the fields in an existing state for a particular joint; or
- modifying the structure of a schema by removing or adding a state.

The first of these seems the less drastic action, since it leaves the basic structure of the schema unaltered. However, there may be limits to what can be accomplished by modifying numeric values; in such cases, one may need to revise the schema structure by adding or removing states. For example, a given movement may be overshooting a particular location during the course of its movement, indicating that the velocity is too high during the previous portion of the movement. A modified schema would reflect this by substituting a smaller velocity in the state description just prior to the failure point. After several such revisions, the schema may be at a point where no adjustment to the velocity will further improve the position of the arm at the time of the failure point. At this point, a completely new state description could be added that would help guide the arm through the proper locations at the appropriate times.

To review our representation, each state description specifies a time value and a set of 3-tuples, each of which consists of a joint identifier, a position vector, and a velocity vector. In principle, any of the values in a state description may be modified except the joint identifier. The current model only considers adjusting the values of velocity vectors and, in regards to structural changes, only considers adding state descriptions. Furthermore, MAGGIE considers modifying only the two data points that delimit the segment of the schema containing the time of failure. That is, for the throw schema of Figure 1 in Chapter 4, if the selected failure point was at time 7, then the second and third state descriptions would be said to 'contain' the failure point and would be considered for modifications. However, selecting among real-valued modifications still leads to an infinite branching factor, so we require some simplifying assumptions to help reduce the effective search space. We employ a constrained generate-and-test method to select among the alternative modifications generated.

For two state descriptions  $S_i$  and  $S_j$  that contain the failure point, the amount of adjustment  $A$  applied to each is inversely proportional to their respective distances (in time) from the failure point. That is, the closer the failure point is to  $DP_i$ , the larger the adjustment made to  $DP_i$ 's velocity. Although this does not guarantee an optimal modification, it provides a reasonable alteration based upon the limited information available from the motor buffer. The amounts of adjustment that are

---

20. Recall that OXBOW serves as MEANDER's (and therefore MAGGIE's) interface to long-term memory.



considered are  $A_i = E m_i$  to  $DP_i$  and  $A_j = E m_j$  to  $DP_j$ , where  $m_i$  and  $m_j$  are computed by

$$m_i = \frac{t_F - t_i}{t_k - t_i} \quad \text{and} \quad m_k = \frac{t_k - t_F}{t_k - t_i}$$

for failure point  $t_F$ , error vector  $E$ , and the associated time values for  $DP_i$  and  $DP_j$ ,  $t_i$  and  $t_j$ .

Based on this calculation, MAGGIE considers all four possible ways of pairwise incrementing and decrementing the two data points discussed above by their respective amounts. Because the failure point may overshoot or undershoot based upon the velocity values at either (or both) of the containing state descriptions, any one of these four critics may yield the most improvement.

The remaining critic suggests adding a state description in the joint-centered schema as outlined above. The new state description is generated using the time of the failure point and the inverse kinematic transform of the desired positions and velocities of the joints as given by the viewer-centered schema. This new state description is inserted appropriately into the sequence that comprises the joint-centered schema. Given this revised schema and the four based upon velocity adjustments, the evaluation function chooses among them as described below.

#### 5.4.3 Selecting the Modified Schemas

The selection of the candidate motor schemas is based on the predicted performance of each at the time of the failure point. MAGGIE estimates predicted performance by generating a partial motor program for each choice and evaluating the error at the specified time. The candidate that minimizes error at this time is selected for further processing as described below.<sup>21</sup> However, because states specified in the schema are generally guaranteed to be reached at their respective times, this simple scheme would always favor the creation of new points when comparing the new partial motor program, with the result of adding a completely new state description.

As mentioned above, adding a new state is a more drastic modification to the schema than simply modifying the velocity values, and it should be avoided if alternatives can suffice. Moreover, in the context of memory storage through OXBOW described below, adding a new point may sometimes decrease performance. For this reason, we have included a *bias* against this choice. As long as the best of the four possible velocity modifications results in an improvement that is greater than the bias factor, the modification is preferred. That is, if the bias factor is set at one-half, and a modification to the velocities can correct 70% of the detected error, then MAGGIE will prefer this modification over the addition of a new state. Only when none of the modifications considered can sufficiently improve the error (at the time of failure) will the system add a new state to the schema. As mentioned above, modifications to velocities may have a limited improvement. MAGGIE's use of the bias factor can effectively knock the system out of local minima, which can lead to improved search through the space of joint-centered schemas.

21. Another method would involve executing all four revised schemas in their entirety and comparing their resulting overall deviations. However, this would be very expensive computationally and we find it unlikely that humans carry out such computations unconsciously.

#### 5.4.4 Modifying Long-term Memory

Once MAGGIE has selected the best of the candidates from among those proposed by the critics, there still remains the need to alter memory for producing future movements. As we stated before, OXBOW is MÆANDER's sole interface to the movement hierarchy. In Chapter 4, we saw how observed movements could be parsed and incorporated into the movement hierarchy to allow more accurate prediction of path trajectories. In this chapter, we have focused on joint-centered schemas and how movements are actually generated, rather than recognized.

MÆANDER improves its motor skills by passing the best candidate produced by MAGGIE's critics, in conjunction with the viewer-centered schema that it originally retrieved, to OXBOW as an "observed" instance. These two schemas are given together to OXBOW. Recall that the two types of schemas are kept distinct, but they are stored together under the same skill concept. When an observed movement is parsed and the resulting viewer-centered schema is incorporated by OXBOW, the information represented in the joint-centered portion of the skill is unaffected. However, when both the revised candidate schema and the viewer-centered schemas are incorporated into the movement hierarchy, the information for both schemas in the skill concept is modified. Typically, the viewer-centered information will be sufficient to classify the combined movement structure to the same place from which it was taken; in such cases the viewer-centered schema will be reinforced, because the means were used when extracting the viewer-centered schema. Occasionally, misclassifications will occur and the viewer-centered schema stored in a node of the movement hierarchy may become degraded. After considerable experience, any single misclassification will have a vanishingly small impact on the viewer-centered schema. Of course, this leads to predictions about learning rates and the effect of practice prior to acquiring a good viewer-centered schema on the learning of joint-centered schemas. We will return to this prediction in Chapter 8.

Finally, we should note the distinction between the learning method described above and mental practice. MAGGIE takes an actual execution with monitoring information and produces a candidate schema to be stored in memory. In all probability, the candidate joint-centered schema that is passed to OXBOW has never been observed or executed. This should not be misconstrued as mental practice, which is an observable phenomenon that results in improved performance (Stelmach, Kelso, & Wallace, 1975; Gallway, 1974). Mental practice involves imagining the execution of a movement and comparing the imaginary movement to the desired movement. Changes can be made based on detected errors, but naturally the quality of the "feedback" is not as good as when physically practicing the movement. In MÆANDER, there is currently no provision for mental practice. Therefore, our model cannot account for the differing benefits from these two practice schemes. In the final chapter, we briefly return to this issue and describe what would be necessary for MÆANDER to account for this phenomenon.

### 5.5 Discussion

In this chapter we addressed the second half of our primary research goal – the generation of movement skills. Throughout the discussion, we touched upon constraints and issues related to

what is known about the generation of motor skills by humans. But predominantly we described MAGGIE, our computational model of skill generation.

After defining the problem, we reviewed MÆANDER's general representations for movements, schemas, and motor skills. Here we formally introduced the joint-centered schema, which is the memory structure that MAGGIE uses to control its jointed limb. One of MÆANDER's important contributions is the flexible representation it uses to represent observed and generated behavior through the two coordinate frameworks. Furthermore, because it stores schemas simply as sequences of state descriptions, the representation supports movements of quite different levels of complexity.

Next we described MAGGIE's performance and learning mechanisms. The former consists of a straightforward feedback control system, but the latter represents one of MAGGIE's contributions as a computational model of motor learning. By employing a set of critics to suggest revisions and relying on OXBOW to store the changes, we have developed a unique combination of supervised and unsupervised learning mechanisms.

In closing, we observe that OXBOW and MAGGIE, the two major subsystems of MÆANDER, each call the other for some aspect of their associated tasks. Again, note that the separation between these components is more complete when looking at the tasks instead of the subsystems. The task of acquiring representations of observed movements is handled entirely by OXBOW. However, the comparison of an observed movement and the movement specified by a concept in memory requires that the points in the viewer-centered schema be expanded by MAGGIE's interpolation mechanism. The task of improving the ability to perform a given skill is mostly the responsibility of MAGGIE, but again, OXBOW is necessary to access and update the hierarchy of movement concepts.



## Evaluating Movement Recognition in Mæander

---

### 6.1 The Experimental Method

As we saw in Chapter 4, OXBOW provides a method both for representing jointed limb movements and for acquiring a concept hierarchy of movement concepts. Naturally, before we can make conclusions about the usefulness of such a system, we must know how well the system operates and improves with respect to some performance task. In this chapter we evaluate OXBOW's behavior on the performance and learning tasks defined at the beginning of Chapter 4. We first present our performance measure, followed by a number of experiments. These demonstrate that OXBOW can recognize observed movements and improve this ability with experience.

#### 6.1.1 The Tasks and a Metric

The performance task for OXBOW is to *classify* a newly presented movement with respect to the current state of the movement knowledge base. As discussed in Chapter 4, this involves associating the new instance with a node in the concept hierarchy that represents previously observed movements similar to the new movement. We have implemented OXBOW to let classification occur without modifications to the concept hierarchy. That is, we use a trimmed version of the learning algorithm that does not consider tree modification operators and that does not alter the contents of the nodes in the tree.

A general metric for evaluating a system's ability to classify instances is *predictive accuracy* (Fisher, 1987; Gennari, Langley, & Fisher, 1989). For movement recognition, the metric we use compares an observed movement trace to the movement trace stored with the concept chosen for classification. We evaluate the system's performance by comparing an idealized test movement to the movement described by the node of the schema hierarchy at which the test instance is classified; the result of this comparison is a *mean absolute error* over the course of the movement. This measure

indicates how far, on average, the limb was from the desired positions. The error score is computed by finding the Euclidean distance between corresponding joints of the arm at corresponding times for the two movements. We take the absolute value of these distances and average over the joints of the arm and over the time slices occurring during the testing movement. This corresponds closely to the *absolute error* measure used in psychological studies of human motor behavior. The error scores we report in the following experiments reflect this averaging over joints and simulated time slices. The units given are for an arm with two joints operating in a reachable workspace of 200 unit diameter.

As a concept formation system, OXBOW addresses two distinct problems. First, it must determine appropriate groupings of movement instances and, second, it must form useful generalizations of these groupings. The latter is an issue for OXBOW because it must establish a mapping between the structural components of movements. We can easily control the first of these two problems by presenting only a single class of movements, thereby letting us evaluate OXBOW's generalization behavior. That is, we can evaluate how well it characterizes a set of movements that have already been correctly grouped. In the next section we test OXBOW's generalization mechanisms and then move on to its clustering mechanisms in Section 6.3. The following two sections, 6.4 and 6.5, contain the results of additional tests with different tasks, and the chapter closes with several conclusions about OXBOW's behavior.

### 6.1.2 An Artificial Movement Domain

Our experiments with OXBOW have primarily involved an artificial movement domain.<sup>22</sup> We have created artificial templates that roughly correspond to four natural movements – a slap, a throw, a wave, and a salute.

As described in Chapter 4, schemas consist of states describing the positions and velocities for each of the joints in an arm. In our templates, the time, position, and velocity values specify a normal distribution from which values are drawn when generating a new movement instance. The values (time, position, and velocity) each have their own distributions with independent variances. Table 6.1 lists the four templates used to generate our artificial movements. The notation corresponds to that used in Chapter 4 when we introduced the motor schema. Because these are joint-centered schemas, the vectors (in square brackets) have only one component specifying the joint rotation and rotational velocity in polar coordinates. The two arm segments are both 50 units long and would be the  $\rho$  value for polar coordinate pairs if we had shown them in the table; we have left this out of the table since they remain constant. The values for time, rotation, and rotational velocity are given as means, with the standard deviation shown as the subscript ( $\mu_\sigma$ ).

In our experiments with this domain, observed movements were produced by motor schemas instantiated from the templates. Each value of an instantiated motor schema was generated as a random sample from the normal distribution having the appropriate mean and standard deviation. That is, each place holder in the template has its own distribution from which values were drawn

---

22. However, we also present initial studies of the system applied to actual movement data from cursive letter generation. Here we describe our artificial domain and delay discussion of handwriting until Section 6.5.

Table 6.1. The artificial movement templates for the four movement types. Values are denoted as means with subscripted standard deviations ( $\mu_\sigma$ ).

---

<b>salute</b>	
(10.0, {	$\langle J_0, [0.0_{0.05}], [0.0_{0.01}] \rangle, \langle J_1, [0.0_{0.05}], [0.0_{0.01}] \rangle$ )
(30.3, {	$\langle J_0, [1.5_{0.1}], [0.0_{0.01}] \rangle, \langle J_1, [-3.0_{0.15}], [0.0_{0.01}] \rangle$ )
(50.0, {	$\langle J_0, [0.7_{0.05}], [0.0_{0.01}] \rangle, \langle J_1, [0.0_{0.05}], [0.0_{0.01}] \rangle$ )
<b>throw</b>	
(10.0, {	$\langle J_0, [-1.5_{0.05}], [0.0_{0.01}] \rangle, \langle J_1, [-1.5_{0.05}], [0.0_{0.01}] \rangle$ )
(20.2, {	$\langle J_0, [0.0_{0.15}], [0.115_{0.05}] \rangle, \langle J_1, [0.0_{0.15}], [0.115_{0.05}] \rangle$ )
(40.0, {	$\langle J_0, [1.5_{0.05}], [0.0_{0.01}] \rangle, \langle J_1, [1.5_{0.05}], [0.0_{0.01}] \rangle$ )
<b>slap</b>	
(10.0, {	$\langle J_0, [0.0_{0.05}], [0.0_{0.01}] \rangle, \langle J_1, [-1.0_{0.05}], [0.0_{0.01}] \rangle$ )
(20.0, {	$\langle J_0, [1.57_{0.05}], [0.1_{0.05}] \rangle, \langle J_1, [0.0_{0.05}], [0.25_{0.1}] \rangle$ )
<b>wave</b>	
(10.0, {	$\langle J_0, [1.5_{0.05}], [0.0_{0.01}] \rangle, \langle J_1, [0.0_{0.05}], [0.0_{0.01}] \rangle$ )
(25.2, {	$\langle J_0, [0.0_{0.15}], [-0.09_{0.02}] \rangle, \langle J_1, [-3.0_{0.15}], [0.0_{0.02}] \rangle$ )
(50.0, {	$\langle J_0, [-1.5_{0.05}], [0.0_{0.01}] \rangle, \langle J_1, [0.0_{0.05}], [0.0_{0.01}] \rangle$ )

---

when instantiating motor schemas. The resulting schema was executed by MAGGIE (without error correction) and the movement was observed and parsed (in Cartesian coordinates) by OXBOW.

We can adjust the variance of the distributions by a scale factor to produce sets of movements that contain different amounts of variability. We use the term *variability level* in the following experiments to refer to the value of this scalar, which adjusts the individual distributions used to determine the values of a newly generated schema. That is, for a given level of variability  $k$  and a place holder in the template  $\mu_\sigma$ , we sample the random numbers from the modified distribution having a mean of  $\mu$  and a standard deviation of  $k\sigma$ . The motor schema generated in this fashion is executed as described above, but the resulting behavior will have either less or more variation from the prototype, as defined by the means of the template.

## 6.2 Learning Single Movement Concepts

By considering only movements of a single type during a given training run, we can control for clustering errors, as described above. However, even with this control, there are still two potential sources for error. One is from the process that incorporates an observed movement into the hierarchy of motor skills (generalization); this process involves finding a best match between state descriptions in an instance and a stored concept. A second potential source of error is the process that classifies an observed movement (recognition); this process amounts to retrieving a schema from memory that is most similar to the observed movement. In this section, we first examine the issue of incorporating a new motor schema and then turn to the issue of retrieving a motor schema from memory.

### 6.2.1 Constructing the Appropriate Schema

Recall that the learning algorithm treats schemas in two passes – first as a set of individual states, in order to find the best match to a particular schema concept, and then as a complete sequence of states, to find the most similar schema among the siblings at the current level of the hierarchy. One of the first things to verify is that the inner treatment – the determination of the PART-OF structure for the movement concept at large – is behaving appropriately. We predict that the structure and values of an abstract schema concept, acquired from instances of a single type (assuming a uniform sample from the class of movements), would closely reflect the structure and mean values of the prototype for the class. Therefore, in our first experiment we isolate and evaluate the task of forming an abstract schema (skill concept) from a set of observed movements. This lets us control for possible confusions between movements of different types, and lets us determine how sensitive the generalization process is to variance in the observed data.

To this end, we first trained and tested OXBOW on instances sampled from only a single movement type. In this experiment, we tested each movement type in isolation over 20 runs, with 40 learning trials in each run. A single learning trial consisted of presenting OXBOW with a parsed movement generated at random. We repeated runs at four different levels of variability (0.25, 0.5, 0.75, and 1.0) for each of the four movement types (slap, throw, wave, and salute) and measured the system's performance after every other learning trial. The performance metric used to evaluate OXBOW in this experiment compares the prototype with the schema stored at the root node of the schema hierarchy. Because there is only one movement type presented in a run, and the root represents the summary over all the observed instances, this comparison lets us control for possible retrieval problems. Figure 6.1 shows four learning curves, summarizing the reduction of error as a function of experience and variability level. Each learning curve represents the decrease in error for a single level of variation, averaged over the four different movement types.

We can draw two conclusions from this figure. First, the learning rate decreases as the amount of variation increases. We would expect the system to require more samples in high variability domains before it could form a satisfactory summary description. Note that after the first few instances, error has decreased drastically at all four levels.<sup>23</sup> However, at the lowest level error drops to its asymptote after two training instances, and at the highest level it requires several more training instances. Second, we see that the asymptotic levels increase with the variability level. These results indicate that OXBOW has trouble finding the central tendency in domains with high variance. Because the data comes from a single prototype, we would expect that the prototype would be recoverable. This effect of variability on asymptote level could either be due to problems determining the values within the states of the learned motor schemas, problems finding the correct structure of the states in a schema, or a combination of both.

To help clarify this issue, Figure 6.2 shows the same data in a different format, graphing the asymptotic error levels for each movement type separately as a function of the structural complexity inherent in the data. We define complexity as the number of states in a parsed description of an

---

23. Prior to any learning, we can define error to be the prototypical movement compared to a stationary arm, but we do not show this in Figure 6.1. We have arbitrarily defined the *no-knowledge* condition to leave the arm in the initial position of the prototype.



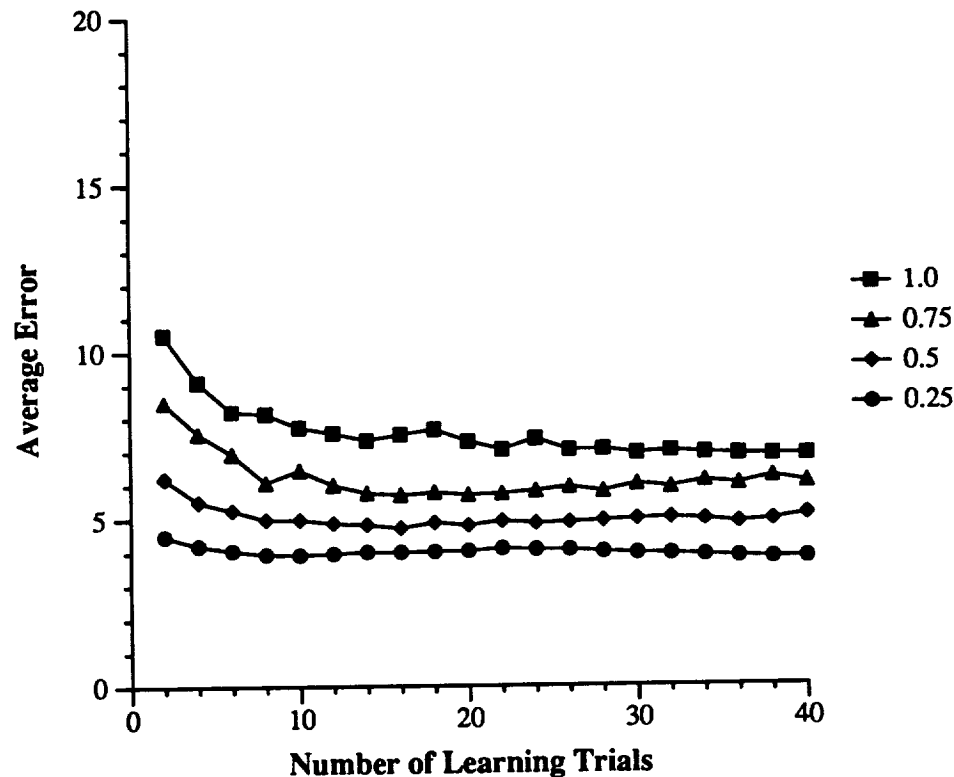


Figure 6.1. Average learning curves when trained separately at four levels of variation in the data.

observed movement. For a given movement type and a single level of variability or noise, we computed the average complexity over 20 randomly generated movement instances. This graph shows a number of interesting points. First, it is apparent from the differing asymptotic levels for the four movement types that the artificial movements we are using are not of equal difficulty. Additionally, it shows how the asymptote and complexity changes for the different levels of variation in the data.

From Figure 6.2 we see that changing the variability in the generated movements does not cause large changes in the structure or complexity of the parsed movements. That is, the number of zero crossings detected by the parser is roughly uniform for the different levels of variability. For example, with the “slap” movement, as the variability of the observed movements increases, the asymptotic error increases, but the structural complexity of the learned schema changes only minimally. The “wave” and “salute” movements do show some increase in structural complexity, but these have relatively little increase in asymptotic error. This stability of the parsed structures with respect to variability suggests that the increased asymptotes in Figure 6.2 do not result from failure to determine the appropriate PART-OF structure for the movement concept, but rather from problems in determining the correct values within the states.

This figure also reveals a surprising result – that increasing complexity tends to decrease asymptotic error level. This non-intuitive result is not without precedent; for instance, vision researchers found that more complexity in the environment makes things easier to disambiguate (Waltz, 1975).

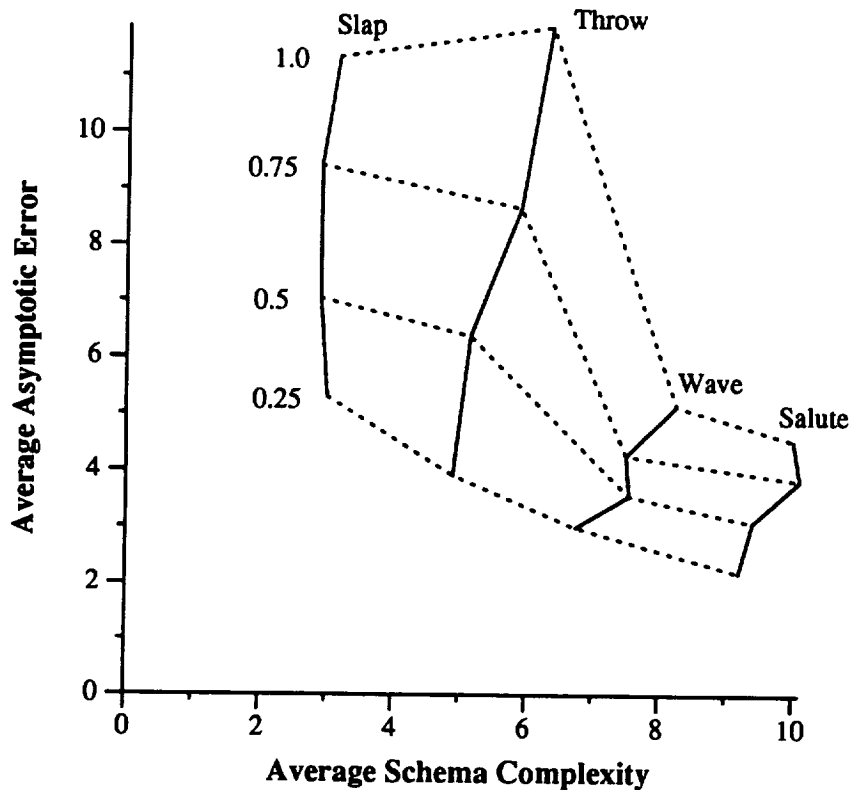


Figure 6.2. A comparison between asymptotic error rates and complexity of input data at different levels of variation.

This suggests that OXBOW should scale up to more complex environments and movements. In future work we intend to study this particular result and evaluate the extensibility of our methods.

Overall, this first experiment indicates that OXBOW captures the PART-OF structure found in observed movements when faced with only a single type, but that its ability to form accurate state descriptions is hampered by increased amounts of variability in the movement data presented during training. That is, we showed that the variability level affects asymptotic error rate but not movement complexity. Furthermore, the results indicate that greater complexity in the training movements leads to improved asymptotic performance.

### 6.2.2 Retrieving the Appropriate Schema

In Chapter 4 we saw that OXBOW relies on its retrieval mechanism to locate a stored concept that is similar to an observed schema. In the previous subsection we used the root of the concept hierarchy as the source for comparison and measurements of error. For an initial study of learning single movement concepts, this was appropriate because we only presented instances of a single type and the root should provide the best “average” or summary of all the observed movements. Retrieving a more specific concept would be considered *overfitting* and should yield a higher error score. However, we predict that there are situations in which performance is actually improved by

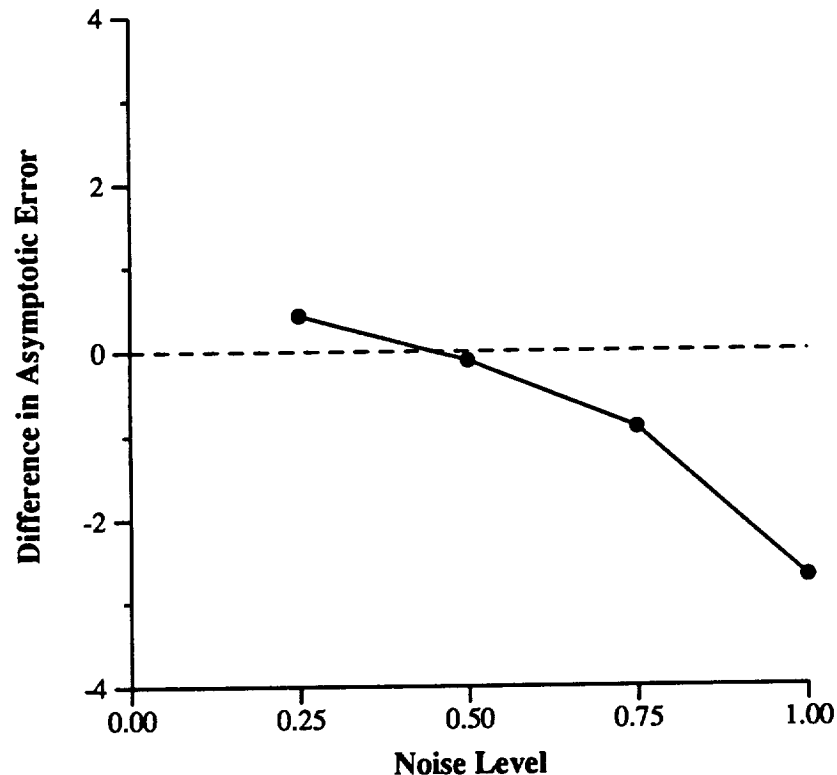


Figure 6.3. A plot of the difference between asymptotic error in the root retrieval condition and the standard retrieval condition (*root - regular*) for the four levels of variability considered.

retrieving a more specific concept than the most general summary description. The reason involves the nature of the abstraction or characterization process. Whenever heuristic search is involved in the generalization process (matching structural components, and especially partial matching), mistakes in the search can lead to non-optimal concept representations. This leaves open the possibility that the root node, as the "complete summary" over both component structure and attribute values, may lead to larger errors than more specific nodes in the schema hierarchy. Given the nature of OXBOW's partial-matching mechanism for state descriptions within a schema, we predict that such will be the case here.

To test this prediction, we repeated the first experiment but instead used OXBOW's standard mechanism to retrieve the schema used for computing error scores (see Chapter 4). In the current context of single movement domains, this would usually be expected to suffer from overfitting and perform more poorly than observed in the previous experiment. We refer to this as the *regular* condition, and in this experiment compare its results to the previous *root* condition, where we simply used the root node as the best classification. We are not particularly interested in learning rates in this case, since this study only varies the retrieval mechanism.<sup>24</sup> Therefore, Figure 6.3 shows the *difference* between asymptotic error levels in the root condition and the regular condition. Negative

24. That is, the same instances were presented in the same order and the same classification choices (during learning) were made in both conditions.

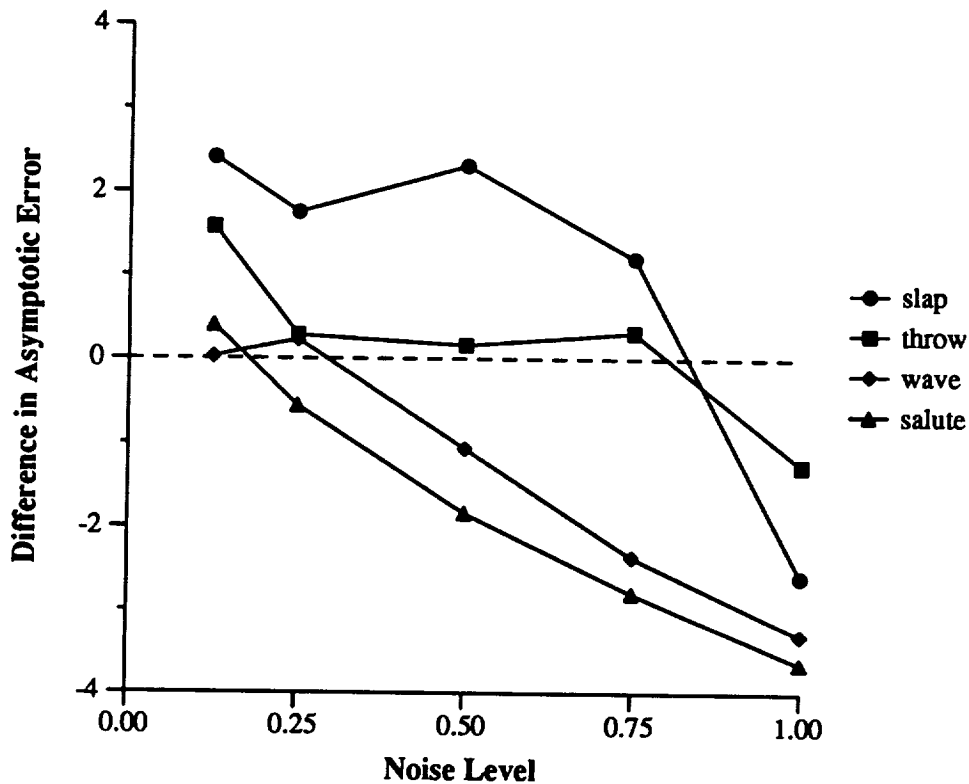


Figure 6.4. Plots of the differences in asymptotic error levels for the root and standard retrieval conditions (*root - regular*) for each individual movement type. Results are displayed for a variability level of 0.125 in addition to the levels from Figure 6.3.

values indicate that the standard retrieval in the regular condition is doing worse than the method of selecting the root node; this is the standard notion of overfitting. The asymptotic values are given for the four levels of domain variability averaged over the four movement types. The results support our prediction. Although the overfitting condition generally does worse than the root condition, the degradation decreases with the amount of variability in the domain and, at the 0.25 level of variability, performance in the regular condition exceeds the root condition averaged over the four movement types. From this trend, we hypothesize that at levels of noise lower than 0.25, even greater advantages are gained over the root condition.

To test this hypothesis, we ran OXBOW under both conditions of retrieval at a lower level of movement variability. As expected, we found that the regular condition outperformed the root condition to an even greater extent than shown in Figure 6.3. However, the results for the individual movement types reveal another interesting characteristic. Figure 6.4 presents the asymptotic differences as computed for Figure 6.3, but for each of the movement classes plotted independently. From this graph we see that each movement type reaches the cross-over point, where standard retrieval begins to deteriorate performance, at different levels of noise. Furthermore, there appears to be a correlation between schema complexity and the trade-off point similar to that found in Figure 6.2.

In summary, we have established a baseline of error levels to which we can compare later results. Furthermore, we have found that increased variability in the domain leads to greater asymptotic error levels and that individual movement complexity correlates inversely with asymptotic error level. We also compared two retrieval methods and found that what would normally be thought of as "overfitting the data" actually produced better results in some cases. Although the root condition was shown to be superior for most variability levels, this retrieval method was only applicable because the system was learning a single concept. In general, we are interested in evaluating OXBOW's ability to appropriately form multiple classes present in the observed data, and this requires that we rely upon the regular retrieval mechanism. Having collected the results in the regular condition for single movements, we can compare them to OXBOW's results on the problem of acquiring movement concepts drawn from a domain with multiple movement classes.

### 6.3 Concept Formation for Multiple Movements

If we had first tested OXBOW on acquiring multiple concepts simultaneously, we would not have known whether performance errors were caused by confusions between categories when classifying an observed movement, problems identifying the appropriate PART-OF structure for a particular node in the hierarchy, or both. The previous study established a baseline for comparison. We can expect that errors above and beyond those reported in the previous section are a result of problems distinguishing between movements of different types. In particular, we predict that having more concepts to learn at a time will slow down learning (require more training instances to reach asymptote) because instances of each individual concept will be observed less frequently than in the separate training condition. Additionally, we predict that the asymptotic levels should not be significantly affected, even though the learning rate should be.

To study these predictions, we ran an experiment in which OXBOW observed movements from all four of the classes, each with an equal likelihood. We presented 40 training instances, from which the system constructed its hierarchy of movement concepts. After every other training instance, we stopped learning and tested the system's performance as described above. We repeated this process at the same four variability levels as before. Figure 6.5 shows the average error (over the four movement types), again as a function of experience and noise level. The errors are averaged over 20 runs with different training orders of the movement types.

The results support our main prediction; that increasing the number of concepts decreases the rate of learning. Comparing Figures 6.1 and 6.5, it appears that OXBOW reaches asymptote at between two and four instances in the separate training condition, and after about 20 instances in the mixed condition.

Since the movement types are selected randomly, more instances are required in order to reliably have observed three or four of each type. In this case, the 20 trials to asymptote is what we might expect given that there are four movement classes and that, individually, three or four trials are needed. As it appears that misclassifications are not a significant problem, this slowdown of learning rate gives some indication that OXBOW accurately distinguishes between observed movements of different types.

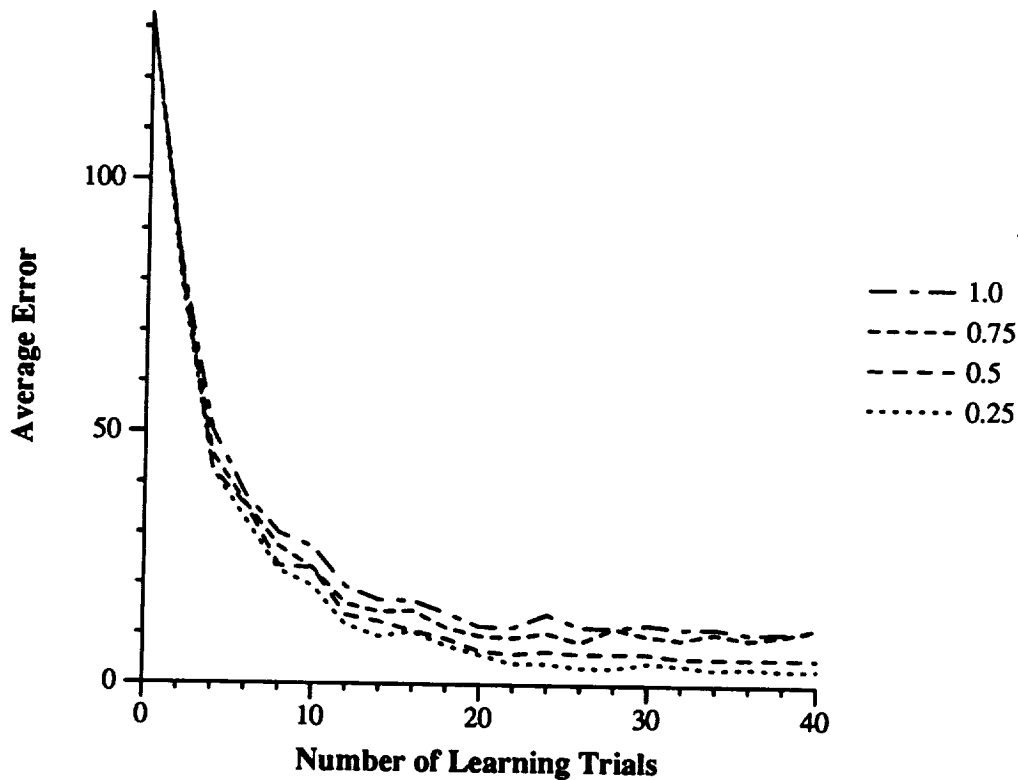


Figure 6.5. Learning curves when trained on multiple classes simultaneously (mixed training condition) at four levels of variation.

The evidence for the second part of our prediction – that asymptotic error should not be significantly affected – is less clear. Figure 6.6 compares the asymptotic performance from Figure 6.5 under the mixed training condition to the asymptote levels found for learning single concepts under the regular retrieval condition. The corresponding asymptotes are plotted for each of the noise levels. The curves indicate a small but definite increase in asymptotic error levels between the mixed and separate training regimes. An analysis of variance indicates that this difference is statistically significant at the  $p = 0.031$  level, but there is no significant interaction effect between noise in the input and the number of concepts being learned. Although this difference was statistically significant, we do not believe that it represents a strong relation between the number of concepts and the asymptotic error level. Additionally, the difference between the conditions was very small – approximately a single percentage point.

We carried out an additional study to help identify the strengths of the previous findings. We predicted that the number of trials to asymptote would vary significantly with the number of concepts learned, but that the asymptotic levels should not vary. This experiment evaluated OXBOW's learning rate and asymptote for learning two and three concepts at a time. Because our earlier experiments on learning single movement types indicated that the difficulty of the four artificial movements varied, we considered all possible ways of choosing two and three concepts out of the four. This led to four sets of runs for the three-concept condition and six sets for the two-concept

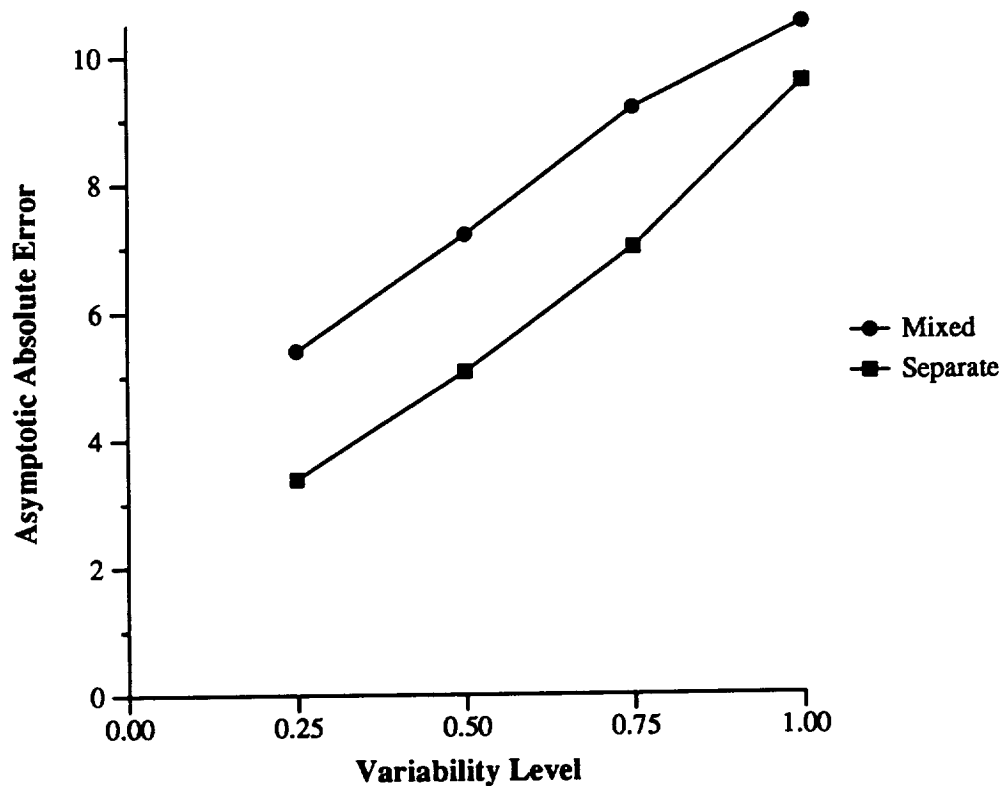


Figure 6.6. A comparison between asymptotic error for the separate and mixed training conditions, plotted as a function of variability level.

condition. In each case, the selected movements were equally likely to be observed. We ran 15 training sequences for each possible combination of two and three concepts, then averaged the results. The results given in Figure 6.7 support our predictions. The number of trials needed to reach asymptote increases regularly with the number of concepts being learned. More important, the level of the asymptote appears unaffected by the number of concepts in the domain. This suggests that OXBOW's recognition performance is robust with respect to increasing the number of concepts.

#### 6.4 Predicting Unseen Movement

In the previous experiments, the performance measure corresponded to what has been termed *recognition* in the psychological literature. That is, the complete prototype of a particular movement class was classified and a comparison was made across the entire duration of the movement. In real life, one would more likely observe a partial movement and need to predict the continuation of the movement. Observing a portion of a movement and predicting future movement corresponds to the task of *recall* in the psychological literature. If we ignore issues of learning, varying the amount of a test movement that is observed provides a method for adjusting the difficulty of OXBOW's retrieval task, thereby allowing a more direct assessment of its contribution to error.

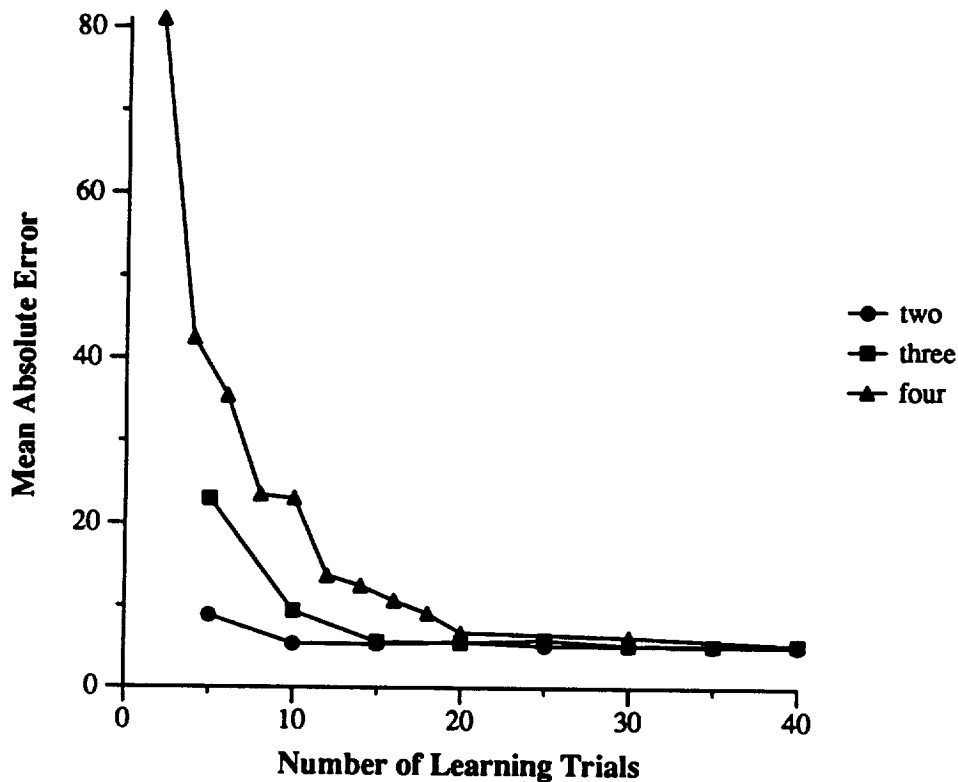


Figure 6.7. Three separate learning curves for learning two, three, and four concepts at a time.

Thus, in a third experiment, we trained OXBOW as described before, but we altered the performance task as alluded to above. When testing, we presented only a portion of the prototypical movement and then measured error over the remaining unobserved movement. Note that complete movements were given during training and only when evaluating system performance did we limit the extent of the observed prototype. We can compare errors among different lengths of predicted movements because we average the total error by the number of time slices compared during prediction. Any differences in errors can be attributed to classification problems during retrieval, because the knowledge base is the same for each level of observation at a given point in training.

This formulation of the task suggests a prediction: as less of the movement is observed, classification should become more difficult and mistakes should lead to greater measured error. Simply stated, the more one is able to observe, the more one should know about what will happen next. Figure 6.8 shows the learning curves from an experiment in which we varied the portion of the movement to be predicted. We fixed the variability level at 0.5 and averaged the results over ten runs of 30 training instances each.

The figure shows that when OXBOW is predicting 80% of the movement (observing only the first 20% of the movement), the errors are consistently the highest (except very early in training, when not all the movement types have yet been seen). However, there is little difference between predicting 50% of the movement and only 20%. This result suggests that the system is not severely affected by having less information available for classification, except in extreme cases like the



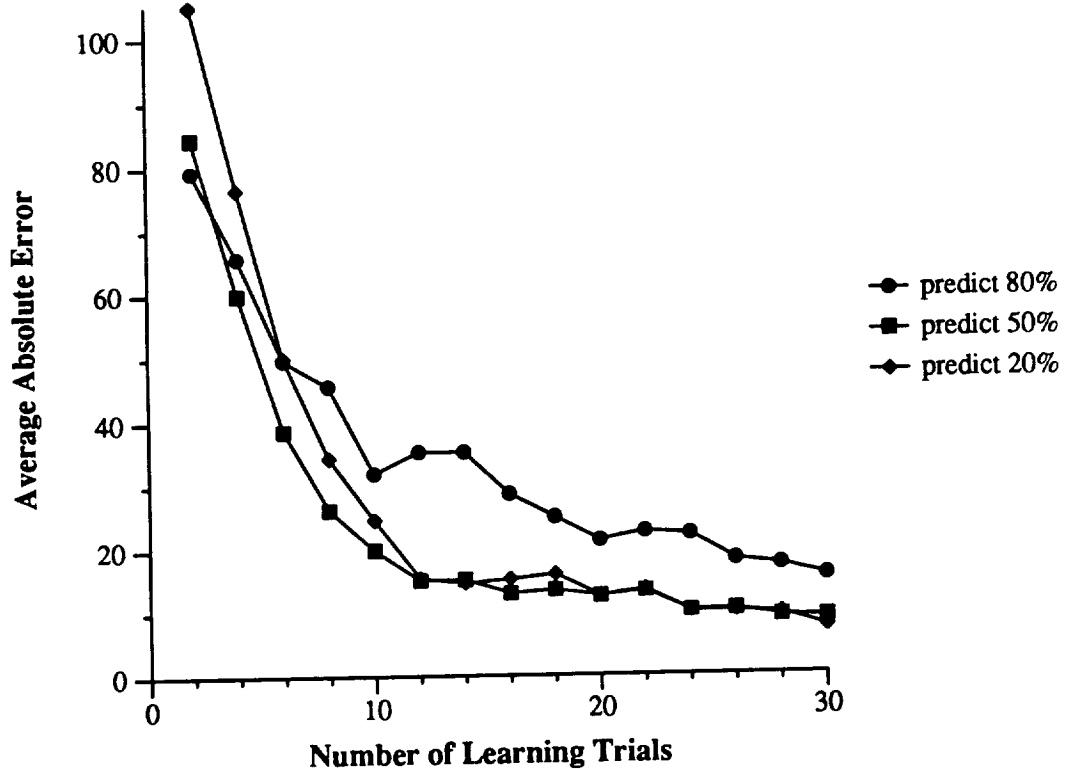


Figure 6.8. Learning curves showing error as a function of the amount of the test movement that is missing and that must be predicted.

80% condition. It follows that there must be some point at which classification accuracy begins to significantly suffer. From previous experiments we know that increasing the variability in the domain increases the asymptotic error levels. We have supposed that these raised error levels occur because the increased noise makes it more difficult to construct high-quality generalizations for the concepts. It seems reasonable to suppose that poor representations in memory make correct classifications of new instances more difficult. Above we showed that observing less of a test movement makes classification more difficult and eventually leads to increased error (attributed to misclassifications). When two factors influence the same mechanism – in this case noise and observation level both making classification more difficult – the factors' influences may interact in a multiplicative fashion. This leads to another prediction: as the training data becomes more variable, the system should require larger portions of the test movement in order to prevent the error from increasing.

To test this prediction, we ran OXBOW in partial prediction mode while training on data with different levels of variability. In a single experimental run for a given level of noise, we trained OXBOW on 60 observed movements and tested predictive performance after every four training instances. We considered four levels (80%, 60%, 40%, and 20%) of the portion of movement that was observed and available to the classification mechanism. As before, the remainder of the movement was predicted using the node retrieved from the schema hierarchy. For each condition

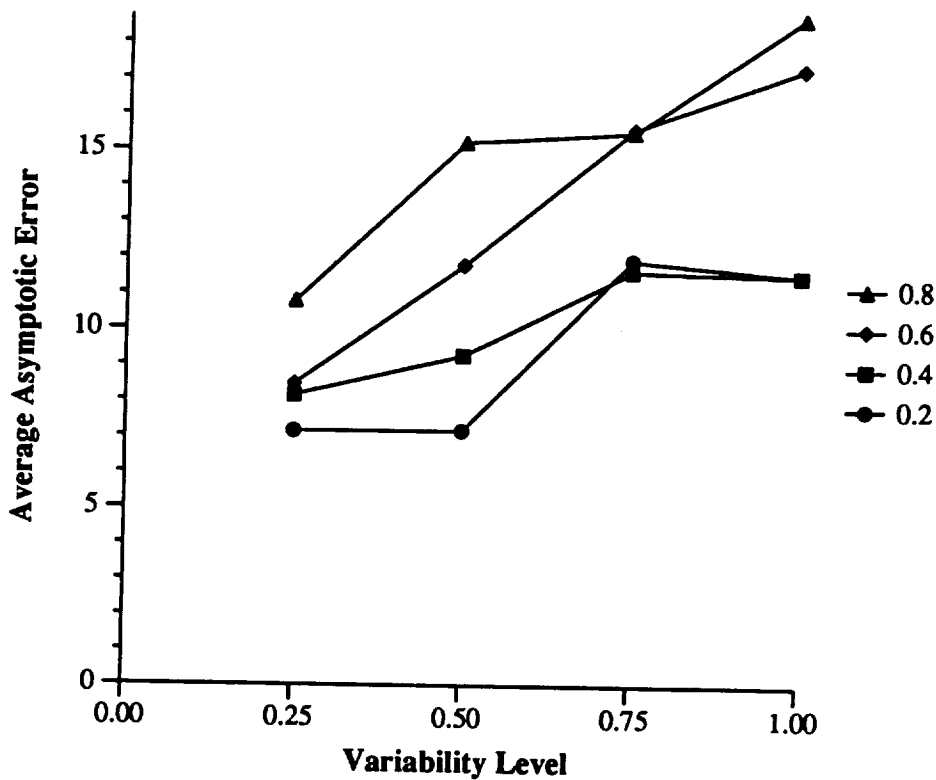


Figure 6.9. Asymptotic error levels after 60 training instances for four levels of domain variability and four levels of the portion of test instance to be predicted.

of noise and observation level, we averaged the results over 20 different training orders to control for order effects.

In this experiment, we again were only interested in asymptotic error levels because we had already considered the affects of variability upon learning rate (shown in Figure 6.7). Altering the performance task in this way should not affect learning rates. Figure 6.9 shows the asymptotic error rates for the four levels of noise as a function of the portion of each test movement to be predicted. The graph indicates similar asymptote levels for the 0.25 variability condition but a wide range of asymptotes for 1.0 level. Separate analyses of variance for these two variability conditions reveal a statistically significant difference in 1.0 condition ( $p < 0.001$ ) but no difference in the 0.25 condition ( $p > 0.1$ ). This would seem to support our prediction of an interaction between noise and portion observed. However, an analysis of variance over all the data shows a significant main effect of the portion to be predicted, but no significant interaction between the two factors.<sup>25</sup> Although our prediction was not strongly supported, the results indicate a relative robustness of the system's retrieval mechanism with respect to noise; that is, when learning from highly variable data, the system is no more adversely affected by incomplete data than when learning from very regular data.

25. An analysis of variance for a design containing only high and low levels of noise (removing the 0.5 and 0.75 noise levels) indicates a significant interaction with  $p < 0.05$ .

More important, the above experiments hold the learning system constant while varying the amount of information in the test movement, thus indicating the sensitivity of the classification process. The results suggest that OXBOW is not making misclassifications when given partial structures in the input. This provides supporting evidence that the increase in error observed in conjunction with increased variability in the domains is due to problems in the generalization process when incorporating new experience. Understanding and reducing these errors remains a topic for future research.

## 6.5 Recognizing Handwritten Letters

The artificial movements introduced above served a useful purpose for evaluating our method of movement acquisition through observation. They were defined by an explicit prototype from which a class of similar movements was generated. However, it is sometimes possible to lose complexities inherent in real-world domains when constructing artificial domains in order to evaluate a particular system or theory. Testing a model on a "real-world" domain helps support a claim that the model's methods are generally useful. In this section we present experiments testing OXBOW's recognition of handwritten letters of the alphabet. Note that this is not the recognition of letters themselves, but rather recognition of the movements that generate letters.

For the following studies, we consider the letters *m*, *a*, *g*, *i*, and *e*. The author generated 63 instances of each letter with his non-dominant hand using a computer mouse. Each letter instance was generated by dragging the mouse, which controlled the endpoint of the arm, and collecting the positions and velocities of the hand during the generation of the letter. For a two-jointed arm with an initial configuration and a fixed base, the movement of the elbow joint is determined by the movement of the hand. This procedure resulted in 315 raw movement traces, which were then parsed as described in Chapter 4 and handed to OXBOW. These letter movements were divided into a training set of 210 instances (42 of each letter) and a test set of 105 instances (21 of each letter). In the following experiments, training letters were randomly drawn (with replacement) from the training set of 210 instances and the system was tested on the entire set of 105 test instances.<sup>26</sup>

In the previous studies we compared the prototypical movement with the movement stored at the node of the schema hierarchy where the prototype was classified. In this way we quantified the error introduced by OXBOW. However, in this case we have no such prototype for comparison. Instead we have fallen back to a simpler task – that of letter-type prediction. In this context, the letter name (e.g., "a") of a training movement is stored at each node in which the movement is incorporated during the classification process. That is, the letter-name attribute is updated as if it were just another attribute in the instance description, but this particular attribute is not used to calculate category utility when determining the quality of competing classifications. The recognition "accuracy" of a given test letter is then computed by considering the letter names of the instances stored at the node of the schema hierarchy where the newly observed movement is classified. The most frequently occurring letter at the node is compared to the observed letter. If

---

26. We should note that this data set is extremely noisy due to two factors: the movements were generated by the non-dominant hand, and they were recorded using a Sun 3/60 workstation running Unix.

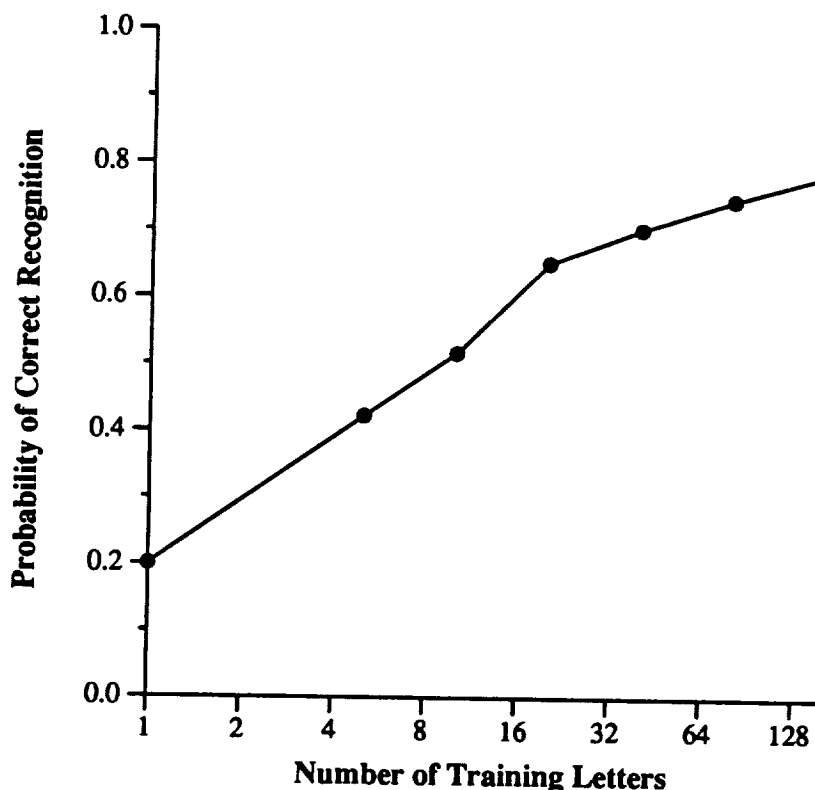


Figure 6.10. Letter recognition accuracy on a test set of 105 hand-drawn script letters plotted as a function of the number of training letters observed.

they are the same, then the letter has been correctly recognized. If  $n$  letters are equally the most frequent and if the label of the observed letter is one of these, then (under a random selection scheme) the letter is said to be correctly recognized at the  $1/n$  level. Otherwise, this test letter is incorrectly recognized. From this method we obtain the percentage of correct classifications over a set of test instances at a given stage of training.

Our first study with recognizing letter movements considered the main effect of improved letter recognition as a function of observation experience. Just as we saw error decrease in the artificial movement domain, we predict that classification accuracy should increase from an initial level of 20% (chance in the case of five letters). Figure 6.10 shows the learning curve averaged over 15 runs of 160 training instances each, and uses a logarithmic scale for the number of training instances. We evaluated OXBOW's performance after 5, 10, 20, 40, 80, and 160 instances on each run. The curve in the figure shows the average classification accuracy at each of these training levels. As predicted, the scores increase as a function of experience but the equal increments in recognition accuracy require successively greater amounts of training experience. However, a question remains about whether the learning rate for letter recognition is affected by the number of letters being learned, as we saw in Figure 6.7 with artificial movements.

As a further test, we partially replicated the earlier experiment in which we varied the number of concepts to be learned. Our prediction is that, as in the artificial movement domains, the number

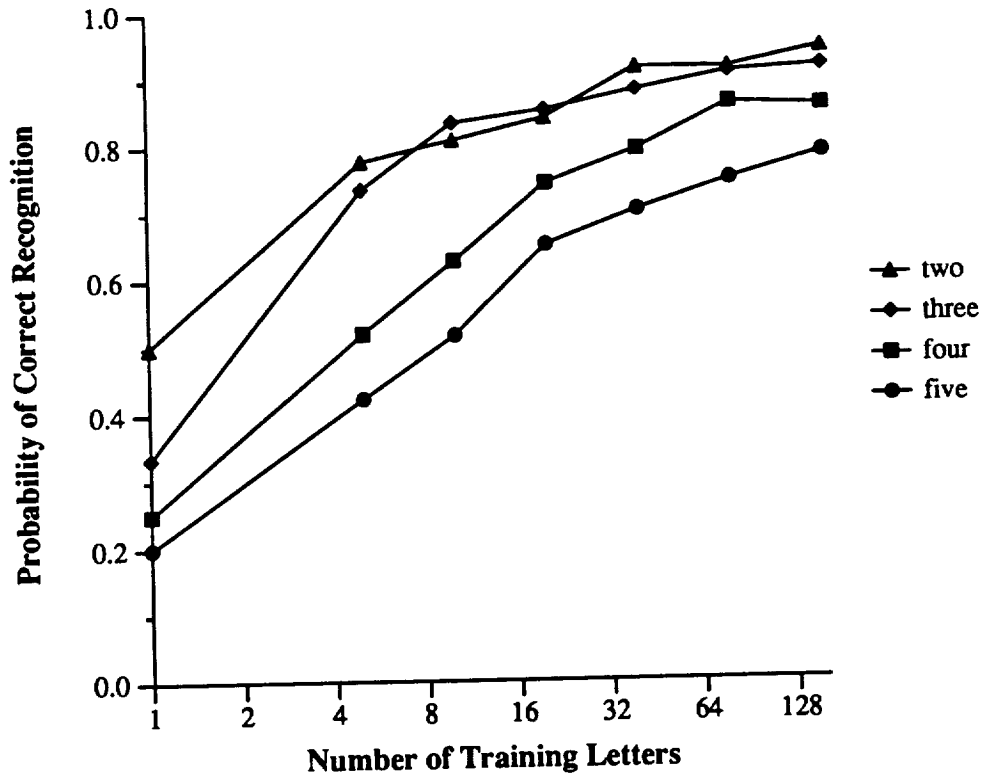


Figure 6.11. Learning curves for training (and test) alphabet sizes of two, three, four, and five letters as a function of training experience plotted on a logarithmic scale.

of letters in the alphabet should affect learning rate but not asymptotic level. As a test of this prediction, we ran OXBOW with two, three, and four letter training and test sets as described above.<sup>27</sup> However, in this case it was not practical to test all possible combinations of two, three, and four letters out of five. Instead we selected single sets of two, three, and four letters to represent the different numbers of concepts learned simultaneously. The appropriate 42 training and 21 test instances for the selected letters were collected into new training and test sets. Figure 6.11 shows the results from two, three, and four letters at a time superimposed upon the results from Figure 6.10, which shows five letters at a time. As in our earlier experiments, we see that reducing the number of concepts to be learned – in this case letters – increases the learning rate.

We mentioned that one drawback of natural domains was the difficulty of quantifying the “desired” conceptual structure. However, one significant advantage is that we can easily compare the results produced by a fabricated system to the results produced by humans on the same type of task. In this case, we can consider the types of mistakes OXBOW makes during letter classification and see whether they correspond to the types of errors that people make.

In this study, we slightly modified the evaluation procedure. Instead of recording the prediction of a letter’s label as correct or incorrect, we stored the actual letter predictions in a confusion

27. Given our performance task and our metric for the letter movement domain, learning a single letter at a time would always yield 100% predictive accuracy.

Table 6.2. Confusion matrix for observed letters (left-hand column) and their classifications (row) after 160 trials.

	<i>m</i>	<i>a</i>	<i>g</i>	<i>i</i>	<i>e</i>
<i>m</i>	0.768	0.060	0.003	0.104	0.064
<i>a</i>	0.106	0.803	0.041	0.053	0.006
<i>g</i>	0.000	0.009	0.962	0.028	0.000
<i>i</i>	0.044	0.022	0.009	0.744	0.179
<i>e</i>	0.073	0.084	0.000	0.197	0.646

matrix. For each of the five possible letters given as a test instance, a set of cells stored the number of times each respective letter was given as the test letter's classification. The resulting  $5 \times 5$  array gives us a picture of the types of confusions made by the classification system. The natural prediction is that similar letters, such as *i* and *e*, will be readily misclassified as each other and have low individual classification scores, but that distinctive letters, such as *g*, will have few confusions and will have high classification scores.<sup>28</sup>

Table 6.2 shows the confusion matrix for the set of runs in Figure 6.10. The values in each cell are averaged over the 15 runs. Inspection of the table reveals that *e* and *i* are the most frequently confused of the letters. This agrees with our prediction that *i* and *e* are the most similar of *m*, *a*, *g*, *i*, and *e*, and should therefore be the most difficult to identify and to discriminate. Furthermore, we see that *e*'s are more frequently mistaken as *i*'s than *i*'s are for *e*'s. Also, we see that the letter *g*, the only letter of the set that descends below the line, is the most accurately recognized. These three observations support a claim that OXBOW is making the same types of error that we would expect humans to make. We might further expect that *e*'s and *i*'s are located close to one another in the schema hierarchy, giving further explanation for the confusions. This is an issue of tree structure and is beyond the scope and intent of the current work, but this study points to confusion matrices as a possible method for understanding the behavior of concept formation systems.

## 6.6 Conclusions

The experiments described in this chapter were intended to evaluate the claim that OXBOW provides a viable mechanism for the storage and organization of motor schemas. Taken together, they provide strong support for this view.

In particular, we argued four important points. First we claimed that the partial matching mechanism finds appropriate correspondences between the temporal structure in instances and concepts.

28. Keep in mind that similarity is determined in the space of handwritten letters. The letters *i* and *e* are similar in shape and the letter *g* is the only descender in the group of chosen letters.

The partial matcher represents one of OXBOW's advances over other concept formation systems, so the results in Figure 6.2, which showed an actual improvement with increasing movement complexity, were especially significant. A second point is that the learning and recognition mechanisms seem robust with respect to the number of concepts present in the domain. This is an important point to establish if we expect our system to scale up to more complex applications. Third, we showed that OXBOW could recognize partially observed movements, and that classification accuracy was not critically sensitive to the amount of the movement observed. Any real-world setting would seem to require some analogous capability that lets an agent predict future events based on current ones. Finally, we showed that OXBOW handles a real-world domain that involves recognizing cursive letter movements. We also noticed that OXBOW made the same types of mistakes that we would expect humans to make; this amounts to a prediction that could be empirically tested in the laboratory. Our prediction emphasizes a point that has been largely ignored in this chapter; the majority of motor phenomena reported in the literature have addressed generation rather than recognition. In the future, connecting this part of our research to psychological phenomena will be a high priority.

At the beginning of this dissertation, we stated that recognition of movements and learning through observation was only the first part of our goals. We expect the same mechanisms to participate in the generation of movements and the improvement of such generation through practice. In the next chapter, we evaluate MEANDER in the context of generating movements that have been previously acquired through observation.





## Evaluating Movement Generation in Mæander

---

### 7.1 Introduction

In the previous chapter we demonstrated that MÆANDER, using OXBOW, could learn to recognize classes of movements through observational learning. However, we set out to construct a computational model that addressed not only the recognition of movements, but also the generation of movement skills acquired through observation. MAGGIE serves this role in MÆANDER by taking a skill concept from OXBOW and using the joint-centered schema to perform a movement that is as close as possible to the one described by the viewer-centered schema. Furthermore, we noted in Chapter 5 that the quality of the model's generated movements should improve through practice. Accordingly, MAGGIE refines the joint-centered schema when it notices errors during performance, and asks OXBOW to store the revised schema with the corresponding viewer-centered schema. In this chapter we evaluate MÆANDER's ability to achieve these goals using MAGGIE's generation capabilities and OXBOW's mechanisms for memory organization and retrieval.

The tests described below follow the experimental methodology developed in the previous chapter. All the movements considered occur in the plane with the two-jointed arm described in Chapter 3. Here we use the same set of artificial movement classes introduced in Chapter 6, as well as the handwritten letter set. Recall that we view motor skills as being first acquired through observation, and then improved through practice. In this vein, we first primed MÆANDER's knowledge base of movements by having OXBOW construct an initial concept hierarchy by observing 120 randomly selected instances from the artificial movement domain. (We will discuss the handwriting domain later.) We generated these instances at the 0.5 level of variability and sampled the four concepts in a random order. This initial hierarchy had a mean absolute error of 6.09 during recognition of the prototypical test instances; this is close to the average asymptotic values found in Figure 6.7.<sup>29</sup> MÆANDER started with this initial knowledge for all of the experiments using the artificial movement domain that we report in this chapter.

---

29. As in Chapter 6, the units given in this chapter reflect a two-jointed arm with a reachable work space of 200 unit diameter.

In testing MÆANDER's ability to generate acquired movement skills, the system first retrieved a movement and then attempted to generate it. The retrieval was done as before, using prototypes as probes, and the retrieved node was then used to generate the behavior. Prior to learning through practice, no joint-centered information was available. In this case, the retrieved viewer-centered information was used to create a schema that holds the arm motionless at the initial position. The resulting error was not the worst possible, but it was still quite large. Once practice has caused joint-centered information to be stored at the retrieved node, an improved joint-centered schema was available for recall and execution. In either case, the generated behavior was compared to the movement described by the viewer-centered schema at the retrieved node.

In Chapter 2 we discussed a number of phenomena that have been observed in human motor behavior. In the following section we address the behavior of MAGGIE's movement generation component with respect to those phenomena pertaining to performance. Next, we evaluate the system's learning operators and their behavior, and consider MÆANDER's behavior both as a computational model and as a psychological one. We conclude with a summary of the results from these experimental studies of MÆANDER's generation and improvement of motor skills.

## 7.2 Behavior of the Performance System

To review from Chapter 5, MAGGIE's performance task is to generate motions that are similar to movement concepts acquired through observation. As usual, the learning task is to improve behavior on the performance task through experience and, in this case, to modify the representation based on errors detected during practice. In this section we ignore learning and focus on factors that influence the quality of generated movements at a given level of generative expertise. These factors include the parameters that control the performance mechanism and the speed of execution.

### 7.2.1 Parameters affecting performance

Our description of MAGGIE in Chapter 5 introduced several system parameters that could influence various aspects of the overall behavior. In general, we want the system's performance to be robust with respect to particular settings of those parameters. That is, the system's behavior should not change radically as a result of small changes in any of the parameters. In our first experiments we evaluate MAGGIE's sensitivity to changes in those parameters that might affect performance. In the case of each parameter, we predict that, at worst, the system's behavior will reflect a graceful degradation with changes in the parameter.

Recall that when MAGGIE detects an error its default response is to generate an error correction that exactly compensates for the current error. Frequently, the model detects an error as the deviation is becoming progressively greater, and radical corrective action is in order. However, such a remedy can also result in overcompensation, leading the model to 'overshoot' the desired position or trajectory. The *compensation* parameter controls how much the system overcorrects or undercorrects by scaling the magnitude of the error correction in response to a detected error.

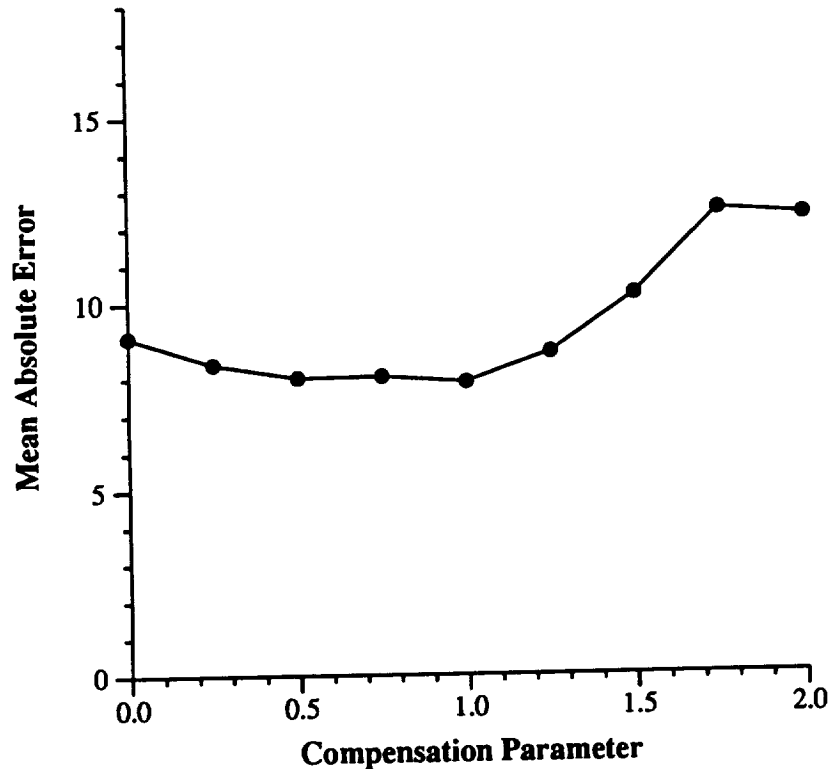


Figure 7.1. Schema execution performance averaged over the four movement types plotted as a function the compensation parameter, a scalar controlling the magnitude of error corrections.

To study the effect of this parameter on performance, we ran the system on all four movement types at nine different compensation settings. In this experiment (and all the parametric studies to follow) we primed MÆANDER's knowledge of movements with 60 practice trials after the 120 observed movements mentioned above. For the compensation parameter, the initial practice prevents a bias toward overcorrections in response to a schema describing a motionless arm. This scheme does not confound performance and learning, in that the knowledge base is held constant for the different settings of the parameters.

Figure 7.1 presents the effects on the model's behavior as one alters the value of this parameter. We see a shallow U-shaped curve, indicating that error increases gradually with over- and under-compensations. This supports our prediction of a graceful performance degradation. One thing the graph does not show is the nature of the movements generated with the different settings of the parameter. Although the mean absolute error does not increase rapidly until above 1.75, the characteristics of the movements change noticeably even at the 1.25 level. For instance, instead of a movement with smooth corrections as necessary, the hand may follow a jagged line that cuts back and forth across the desired path. This effect becomes quite significant at the 1.75 level, even though absolute error is still relatively low. Although we did not plan the model to behave in this fashion, we believe it makes sense. A high setting for the correction parameter will cause the system

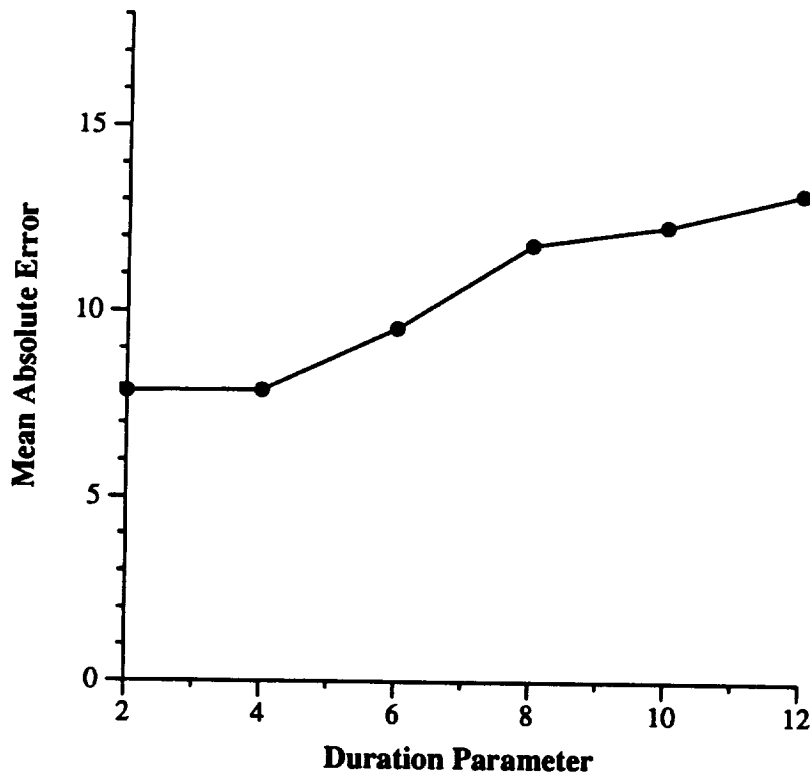


Figure 7.2. Average movement error as a function of the duration, a scalar parameter controlling the length of time a correction is applied.

to overcompensate, and this can lead to oscillations.<sup>30</sup> This characteristic behavior is frequently observed in humans, especially when performing novel or difficult tasks.

Another of MAGGIE's parameters determines the duration of an error correction. That is, an error correction of a given magnitude can be applied all at once with a great burst of force, or over a longer period of time with a more gentle force. As long as this *duration* parameter is less than the monitoring frequency, changes to the duration should have little or no effect. However, when the duration extends beyond a single cycle, we would predict that the effect should be similar to that observed for over-compensation. This should result because the longer duration stretches the correction over a long period. When it is time to monitor again, only part of the original error has actually been corrected, and therefore the remaining portion will be counted twice. This should cause the next error correction to be artificially large, as the extra error would have been corrected eventually by the previous cycle of monitoring and error correction.

Figure 7.2 shows the results of varying this parameter over a range of settings, from correcting all of the error in two time slices to stretching the correction out over a total of three monitoring cycles (four time slices each). In agreement with our prediction, we see the mean absolute error increase as the duration parameter increases. The amount of increase corresponds closely to the increment in error when increasing the compensation parameter. However, in this case there is a way to avoid

30. The default value of the compensation parameter is one.

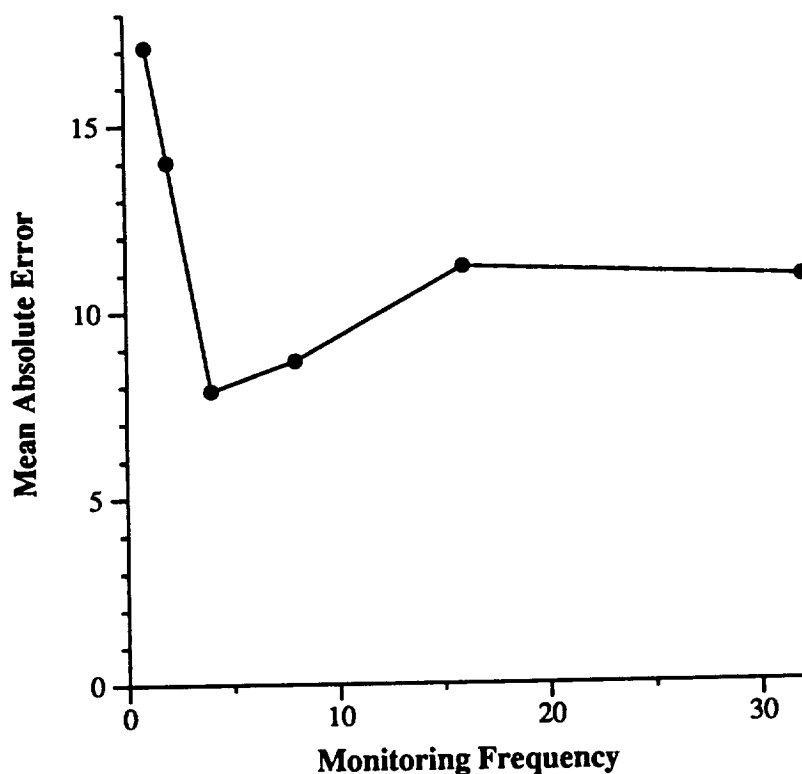


Figure 7.3. Plot of absolute error as a function of different frequencies for monitoring and error correction.

this performance degradation. Pew (1974) suggests that information about corrections in progress, is shared across between monitoring cycles. This approach avoids the multiple avoids the multiple correction problem we observe here.<sup>31</sup>

A final parameter that we should consider is the *frequency* of monitoring during a movement. This parameter controls how often the error-correction mechanism has the opportunity to improve a movement. We view the frequency of monitoring as a parameter that should be under the conscious control of the acting agent. This is related to the issue of attention. When the monitoring frequency is small (i.e., the agent is paying close attention and monitoring frequently), errors are quickly detected and corrected before they become large and significantly degrade performance. We predict that, for a given movement at a fixed skill level, the larger the monitoring frequency (the fewer actual opportunities to make corrections), the larger the error.

Figure 7.3 shows MAGGIE's performance over a range of values for this parameter. At first glance, the results appear to contradict our prediction, instead showing the most severe errors when monitoring very frequently. However, as we discussed above in the context of the duration parameter, this is not surprising because we held the duration parameter at its default of four time

31. We have chosen not to implement this type of mechanism because we are focusing on the integration of skill acquisition and skill improvement. Nothing precludes such a mechanism, and we intend to follow up on this issue in future work. Our default value of the duration parameter is the same as the default monitoring frequency, which is set to four.

slices. Accordingly, when the monitoring frequency is less than the correction duration, we can expect such amplified errors. Again, the mechanism suggested by Pew would correct this situation. However, the results for settings above four would not be altered by this proposed mechanism and they currently confirm our original prediction.

In summary, each of the parameters considered here shows some effect on performance, but none of them indicates a brittleness in the system that would be considered undesirable. One caveat is the high error rates resulting from low values of the monitoring frequency parameter; we have accounted for this behavior and suggested how it can be corrected within MEANDER's framework. Also, if we consider the data for the movement types individually, we see consistent behaviors with respect to changes in the parameters. Therefore, we may assume that the default values for these parameters are not required in order for MAGGIE to perform reasonably. Now let us turn our attention to how well the performance model accounts for psychological phenomena.

### 7.2.2 Human Performance Phenomena

In Chapter 2, we discussed a number of phenomena in the psychological literature that constrain plausible models of human motor behavior. We noted that one of the most robust findings in human performance involved a tradeoff between the speed at which a movement is generated and the accuracy of the resulting movement. Although we presented two different versions of this tradeoff, our task most closely corresponds to the time-matching tasks in which the linear tradeoff holds (Schmidt et al., 1979; Write & Meyer, 1983). Since MAGGIE can run motor schemas at different speeds, we can test the model's ability to account for this tradeoff. We predict not only that error will increase as execution speed increases, but that the rate of increase should be linear.

To test this prediction, we primed the knowledge base with the 120 observed movements as before. In this case, the movements generated were based upon a naive joint-centered schema that holds the arm motionless at the initial position. We varied the speed by multiplying the movements by a scalar factor. Figure 7.4 shows a scatter plot of the errors for each of the four movement types at differing execution speeds. Clearly, executing the schemas at higher speeds leads to greater errors, thereby confirming the first part of our prediction. This effect emerges naturally from the inherent delay in error corrections. The more quickly the system runs a joint-centered schema, the farther the arm will travel during the fixed delay. Note that this is not a sufficient explanation of the tradeoff in psychological terms, as ballistic movements of shorter than the delay for humans (200 msec.) also display this tradeoff. We acknowledge that other mechanisms contribute to the phenomenon (Schmidt, 1985; Meyer et al., 1990).

The second part of our prediction, the linearity of the tradeoff, is less clear from our results. Applying a linear regression to the data produces a good fit ( $r = 0.8903$ ), but one that is not extremely strong by psychological standards. One explanation for the weakness of this fit is that each movement type is inherently different in character and difficulty. The psychological experiments used to study this phenomenon compare results from a single movement pattern at different speeds and distances. Viewed in this light, our regression is comparing apples and oranges, and the reasonably good fit we obtained is surprisingly good! If we follow this idea and perform regressions on the data

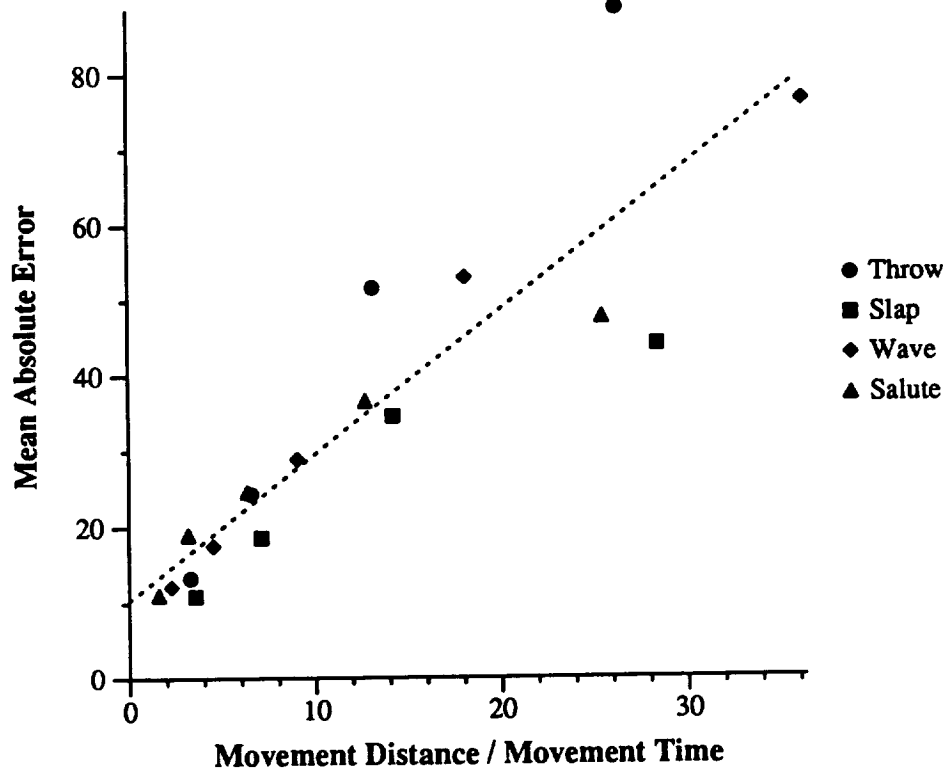


Figure 7.4. The best linear model fit to the execution data showing mean absolute error as a function of movement speed.

from the individual movement types at varied speeds, we get much stronger correlation coefficients ( $r = 0.9960, 0.9550, 0.9851, \text{ and } 0.9668$ , respectively). In terms of Schmidt et al.'s (1979) tradeoff function,  $S = A + B(D/T)$ , we can interpret the different slopes of the regression lines (not shown) as a reflection of movement difficulty.<sup>32</sup> This notion is supported by Figure 6.2, which revealed that the asymptotic error levels for each of the movement types differed considerably.

We believe that this tradeoff demonstrates the continuum between open-loop and closed-loop behavior (Stelmach, 1982), which reflects the amount of monitoring that occurs during movements. When performing a skill slowly, one can make frequent adjustments, thus operating in a closed-loop mode. As the speed of the skill is increased, the performer monitors less often, thereby moving performance towards the open-loop end of this continuum. We address a number of other issues elsewhere (Iba & Langley, 1987).

Our model also provides an account for the transfer of motor skill between limbs (Raibert, 1976). This phenomenon concerns the qualitative similarities between stylized movements performed using different appendages. MEANDER stores each joint-centered schema without reference to the particular limb involved. Thus, the system could take a schema designed for shoulder, elbow, and wrist joints and execute it on a different arm or even on a hip, knee, and ankle. However, to the

32. Recall from Chapter 2 that  $S$  is the standard deviation (variable error),  $D$  is the distance traveled,  $T$  is the length of time taken by the movement, and  $A$  and  $B$  are constants.

extent that learning has fine tuned the schema for a given set of joints, performance will degrade drastically when it is run on limbs with different physical characteristics. However, the overall qualitative characteristics inherent in the schema would still be present. We have not yet run tests of this sort, but we predict this behavior would follow naturally; this is one of our priorities for future research.

One final performance phenomena (not discussed in Chapter 2) that we mention here is the longer reaction times necessary to initiate more complex movements (Fischman, 1984). This "set-up" time is explained in *MEANDER* as the longer time required to classify a complex motion. Again, we define complexity relative to the number of state descriptions in a schema. Retrieving a joint-centered schema from an indexed concept node takes an amount of time proportional to the number of state descriptions in the probe.

In summary, *MAGGIE* explains a number of well-known phenomena relating to motor performance. However, our main concern is with *learning*. In the following section we describe the model's empirical behavior on this dimension and its relation to human motor learning.

### 7.3 Behavior of the Learning System

In addition to *MAGGIE*'s performance characteristics, which we considered in the previous section, we are naturally interested in how the system improves its performance as a result of practice. However, in the context of learning, recall that *OXBOW* serves as *MAGGIE*'s sole memory interface. Therefore, in order to evaluate improvement, we consider *MEANDER* as a complete system made up of *OXBOW* and *MAGGIE*. We assumed this in our experimental studies of performance, but here we make this explicit: in order for improvements to be realized, *OXBOW* must properly store and retrieve the modified schemas that *MAGGIE* generates. Where appropriate, we view our experimental studies in the light of those psychological phenomena that pertain to learning.

#### 7.3.1 Improvement Through Practice

Naturally, we would expect that, as *MEANDER* gains experience through practice, its performance will improve on later executions. Furthermore, we would expect improvements to be significant early on but that performance should approach an asymptote with later practice. To test this main learning effect, we again primed *OXBOW* with 120 observed movements sampled randomly from the four artificial movement types. These training instances were generated at the 0.5 variability level. With the resulting hierarchy of viewer-centered schemas, we had *MAGGIE* practice the four movement types (in random orderings) for 100 practice trials. We measured performance by comparing the executed behavior to the viewer-centered schema that was retrieved by a probe of one of the four prototypes. Figure 7.5 shows the reduction in *MAGGIE*'s absolute error over the course of practice. These values are averaged over the four movement types and over ten different training orderings. The figure indicates that, as expected, the system's performance improves quite rapidly after initial practice, but then improves more slowly and levels off altogether with subsequent practice.



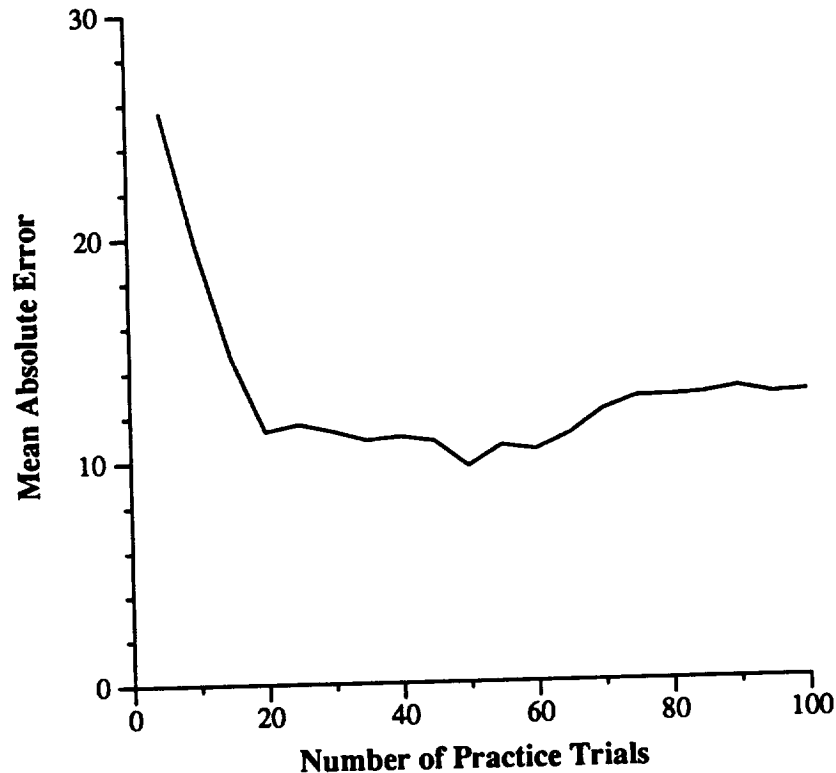


Figure 7.5. A basic learning curve showing the reduction in execution error as a function of practice experience, averaged over the four artificial movement types.

In Chapter 5 we introduced MAGGIE's two learning critics and the bias parameter that determines which one is applied in a given situation. In another experiment we examined the model's learning behavior for different values of the bias parameter. As with the parameters considered in the previous section, we predict that behavior – in this case improvement over practice – will not be seriously affected by moderate changes in this parameter. We tested five levels of the bias factor from zero to one. For each level, we started the system with an initial hierarchy of 120 viewer-centered schemas. A single run consisted of 50 practice movements with performance evaluation after every five trials. Each parameter setting was tested in this fashion over ten runs of different schema orderings. The results (not shown) were fairly uninteresting. An analysis of variance indicated no significant differences in either the learning rates or asymptotes for any of the levels. On a closer look, we noticed that the velocity-modifying critic was rarely used. Even at the zero level, in which the system prefers to make velocity adjustments if *any* improvement is anticipated, this critic was selected less than eight percent of the time. Over most of the range, MAGGIE always preferred to add points, and learning behavior was identical for each of those conditions ( $0.25 \leq \text{bias} \leq 1.0$ ).

To explain this finding, we hypothesized that, because the initial knowledge base given to the system was only observed schemas (no joint-centered schemas), the velocity critic was at a severe disadvantage. Recall that when no joint-centered information is available, a single state description describing a motionless arm is used to generate the "action". In this case, adjusting the velocities

(defined to be zero) may make things worse when evaluating the arm positions at the time of the error point. Therefore, we tested the system with an analogous procedure in which the initial knowledge base also had 60 practice trials incorporated, but again we found no significant difference between parameter levels. Furthermore, this variation caused no noticeable increase in the frequency of use for the velocity critic. Although this indicates that, as we predicted, our system is not overly sensitive to changes in this parameter, it also indicates that we could simplify our model by deleting the parameter and the critic responsible for modifying velocity values. There are two possible reasons that the velocity critic is being largely ignored. Either the critic itself is suggesting modifications that are not improvements, or the evaluation function is not evaluating the critic's suggestions properly. However, previous studies suggested that the velocity critic was significantly useful for at least one type of movement (Iba & Langley, 1987), and we intend to focus attention on this issue as part of our future research.

### 7.3.2 Human Learning Phenomena

Above we considered some performance characteristics of MAGGIE and how they relate to the phenomena presented in Chapter 2. Now let us consider MÆANDER in the context of phenomena that describe human motor learning. As we mentioned before, improvement over time is not sufficient for a psychologically plausible model of motor learning. The nature of MAGGIE's learning mechanism, as described in Chapter 5, theoretically leads to power law improvements in mean absolute error. This should arise from attending to the largest errors first, causing the most dramatic improvements in performance during early stages of practice. However, our preliminary results about improvement through practice are inconclusive. Figure 7.5 certainly shows a decreasing reduction in absolute error, suggestive of a power function. However, it is relatively easy to fit a power function to any data and so we remain hesitant. An added problem is that the reported human learning curves have measured performance either as the number of units produced per time, or as the average time to completion of task. We must find new ways to test MAGGIE, since our studies measure the quality of the trajectories. Although we cannot make strong claims at this time, the results displayed in the figure are not discouraging.

In section 7.2.2, we showed how our performance model accounted for the speed-accuracy tradeoff. However, it seems natural to expect learning to affect this phenomenon. We predict that as the skill level increases, the severity of the speed-accuracy tradeoff should decrease; that is, the slope of the best fit line in Figure 7.4 should become more level as a function of practice. We tested this prediction by stopping MÆANDER at several points during practice and testing the various movements at a range of speeds. A single learning run consisted of practicing 28 movements and measuring errors at speedup factors of 0.25, 0.5, 1.0, 2.0, and 4.0 after every four practice trials. Again, we averaged our results over ten runs with different orderings of training instances.

Figure 7.6 shows that MAGGIE's speed-accuracy tradeoff changes with practice averaged over the four movement types. As the skill level improves, the tradeoff curve becomes flatter. That is, modifications to the schema let the system's behavior rely less heavily upon monitoring and error correction. This means that MAGGIE can execute the schema at a higher speed – even though

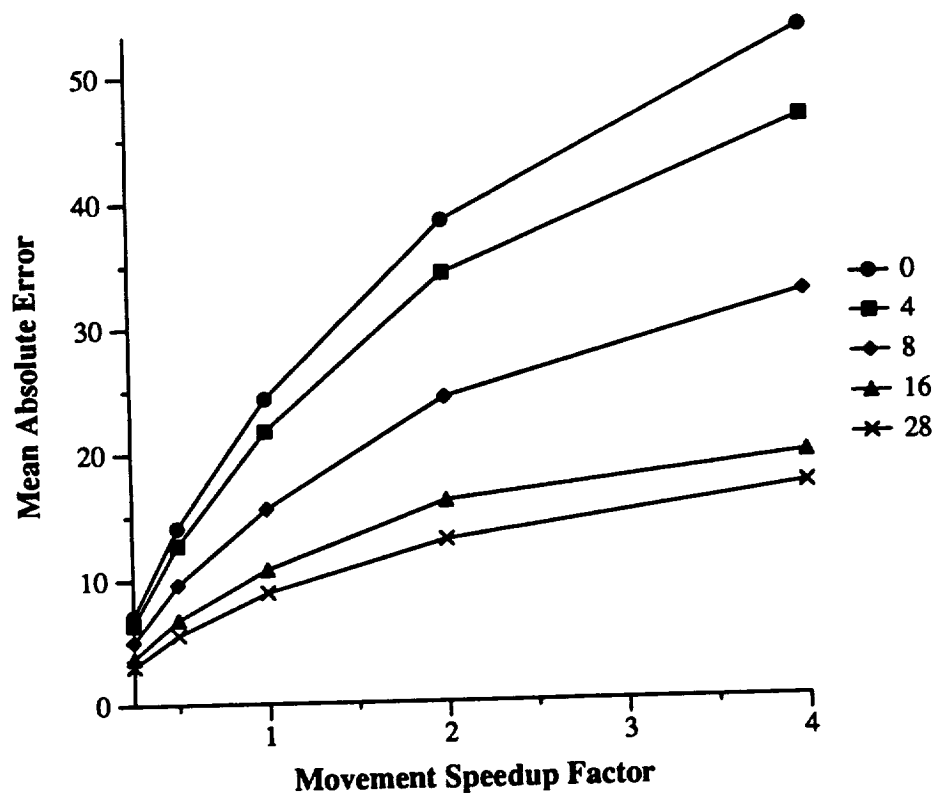


Figure 7.6. Speed-accuracy tradeoff curves measured prior to any practice and after 4, 8, 16, and 28 learning trials.

there are fewer chances for monitoring – without seriously decreasing its accuracy.<sup>33</sup> After making this prediction and carrying out our experiments, we found evidence that suggests this holds for human behavior. Sugden (1980) showed that the index of difficulty for identical tasks decreased with the average age in the groups. Regardless of the particular form of the tradeoff (log, linear, or power function), this implies that errors will decrease as various skills are improved, which is usually coincident with getting older.

Although we have shown that execution speed affects performance error, we would also predict that it should affect the learning rate. As movement speed is increased, not only are there fewer occasions for error correction but also fewer opportunities to learn. MAGGIE focuses its attention on a single error point; thus, as long as at least one error is detected, there is the opportunity for improvement regardless of speed. However, the quality or representativeness of the detected error point will not be the same in all cases. We predict that slower execution allows more representative error sampling and leads to more *effective*, or rapid, learning. Since both conditions have access to the same data in the long run, there is no reason to expect that the asymptotes will differ.

33. These results suggest another prediction: learning should produce a transition in skills from closed-loop processing to open-loop mode, in which feedback is unnecessary and a motor skill can be carried out accurately with little attention.

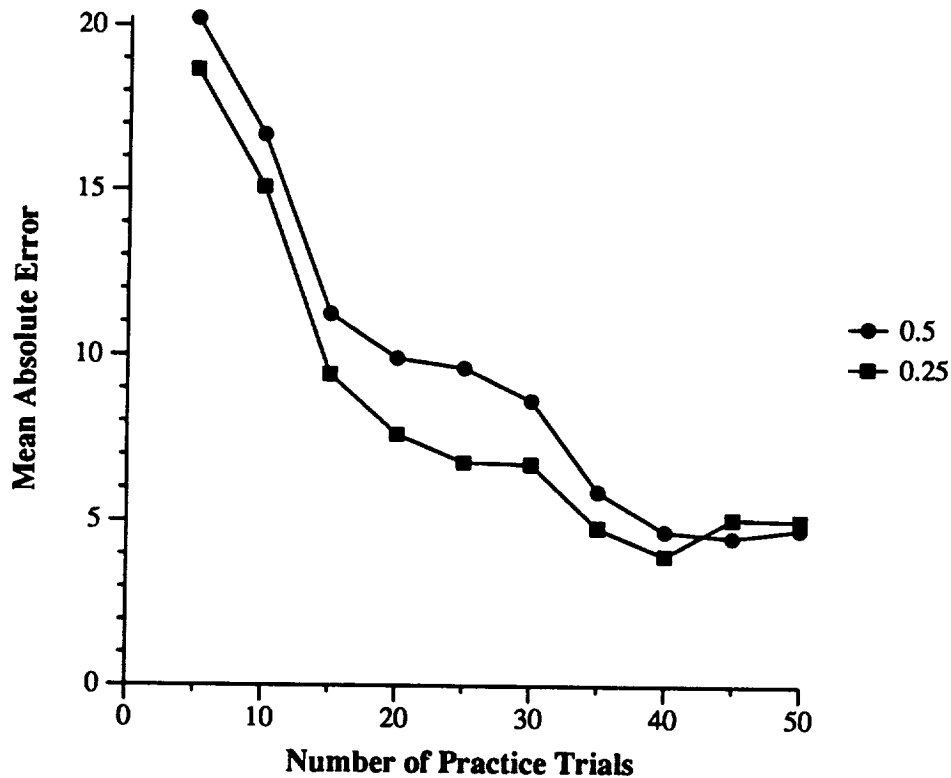


Figure 7.7. Two learning curves showing the rate of improvement in performance for practice speeds of 0.25 and 0.5.

To test this prediction, we ran another experiment in which we varied the practice speed during training. As before, we started *MÆANDER* with the initial hierarchy of observed schemas, but we slowed down the practice movements during training by different amounts in two conditions. We evaluated performance for both cases by running the schemas at the standard rate and measuring errors as before. Figure 7.7 shows the results of this experiment over ten random practice orderings of 50 practice trials each. Clearly, our prediction was borne out, as the slower practice condition improved more rapidly than the 0.5 slow-down condition. Also notice that both learning curves achieve the same asymptotic levels. An analysis of variance indicates a significant difference between the conditions at the  $p = 0.002$  level.

In this section, we showed that *MÆANDER*, using both *OXBOW* and *MAGGIE*, gradually improved its performance as a function of practice. Additionally, we examined the effect that the velocity modification critic has on learning and found it to be seldom used. In future work, we will either replace the critic or modify the evaluation function. We also demonstrated the richness of our framework by exploring two predictions of the model's behavior. One of these, the effect of practice on the speed-accuracy tradeoff, was later found to be supported in the literature, and both predictions could be tested on human subjects. In summary, the previous sections have shown that *MAGGIE*'s performance and learning mechanisms are effective and robust, that *MÆANDER*'s behav-

ior conforms to the behavior observed in humans, and that the model provides potential insights into unexplored human phenomena.

#### 7.4 Generating Script Letters

In Chapter 6, we demonstrated OXBOW's ability to recognize handwritten letters of the alphabet, giving evidence of MÆANDER's applicability to real movement data and "real-world" domains. A natural second step is to have MÆANDER attempt to generate the letters it has learned. Below we describe our efforts in this direction. As before, we first presented a sequence of observed letter movements; in this case we used a sequence of 160 letters drawn randomly from our set of 210 letters. As described earlier, the training procedure involved selecting a movement to generate using a probe letter, practicing the movement described by the retrieved concept, and storing the revised joint-centered schema together with the retrieved viewer-centered schema. However, our evaluation metric was more complex than in the previous study. During testing, OXBOW retrieved a movement concept based on a given probe. The retrieved movement was then executed and the resulting action was presented to OXBOW as an "observed" movement. This latter movement was classified with respect to the initial hierarchy (i.e., the concept memory prior to any practice). In this way we could quantitatively measure the "recognizability" of the letters that MÆANDER generated.

We first ran the system over a single ordering, measuring *cumulative* classification error for the "observed" movements generated by MAGGIE. The results (not shown) revealed no improvement. Thinking the problem resulted from the high noise in the data set, we created a smaller data set by filtering out letters that were considered poor quality. On a second run using this data set, MAGGIE's performance improved to 60% classification accuracy but then degraded to 40% (random guessing would yield 20%). Although performance still failed to reach the ideal, this study revealed the nature of the problem.

In both runs, the probe letters were usually correctly classified, even those that we considered of poor quality. The problem was that the indexed concepts lacked joint-centered information, even after considerable training. Recall that a probe is a skill concept that consists of a viewer-centered schema, which describes the movement that MÆANDER is intended to generate, and an *empty* joint-centered schema. OXBOW is supposed to take the probe and perform pattern completion over the joint-centered schema based on prior practice with MAGGIE. That is, given a probe with a missing joint-centered schema, we wanted OXBOW to retrieve the joint-centered schema from long-term memory that is associated with the closest match to the viewer-centered information present in the probe. This point is important, and we will return to it later in this section.

For example, when given the letter *g* as a probe, OXBOW should retrieve a skill concept from memory in which the joint-centered schema summarizes one or more practice trials on the letter *g*. Instead, the system retrieved a skill concept with a very similar observed *g* but without any joint-centered schema. Therefore, MAGGIE started from scratch and revised the initial motionless joint-centered schema. But when MÆANDER tried to store the combined retrieved viewer-centered and revised joint-centered schemas, OXBOW stored the new pair at a place in the skill hierarchy

where it had previously stored *complete* instances – those having both viewer-centered and joint-centered information. Although OXBOW might have been constructing a very good joint-centered schema in this portion of memory, the next time a *g* probe was presented (with the missing joint-centered information), the same observed viewer-centered schema was retrieved without the benefit of the prior practice. In short, MÆANDER was losing access to portions of its long-term memory.

Naturally, we want to know why this behavior is occurring. Filling in a missing component of a two-component concept should be no more difficult for OXBOW than predicting unobserved movement based on an initial phase of the movement, as we considered in Chapter 6. Both involve completing missing structure based on partial information, and OXBOW performed quite well on predicting unseen movement. However, in the prior study there were no concepts in memory that represented partial movements, and here there are concepts that have missing joint-centered components. Ironically, it seems OXBOW is doing its job too well. An instance with a missing component is more similar to a stored concept missing the same component than to a complete concept (even one that has an identical first component). It is important to note that MAGGIE's learning critics improve the joint-centered schemas to the point where generated letters are recognizable; the problem lies in how OXBOW stores and retrieves this information. Actually, that is only the surface problem.

The real problem lies in the discrepancy between our task design for this study and the formal problem statements in Chapter 4. That is, what we *wanted* OXBOW to do was not what we *designed* it to do. Originally, we stated that, given an observed instance, OXBOW should retrieve a concept from memory that is most similar to the given instance. Instead, we are essentially asking it to find the best component that is associated with the given instance. The emphasis here is placed on completing or filling in missing information in the instance, rather than matching the instance, in its current form, to concepts in memory.

There are several classes of responses to this situation. The first involve "hacks" to the retrieval or storage mechanisms that directly address the desired behavior (which we did not specify). One idea is to let MAGGIE's selected critic modify the joint-centered information of the retrieved skill concept. This would change the joint-centered schema in long-term memory without having to reclassify the viewer-centered and joint-centered schema pair. Another approach involves altering the category utility function to evaluate matches only on the basis of the viewer-centered information in a concept. Both of these proposals implicitly modify the original goals of our system, and make intrusive changes to MÆANDER's mechanisms.

A second class of responses involves explicitly changing the nature of the task addressed in our current study to one that corresponds to the intended purposes of OXBOW. This approach also seems unsatisfactory, as the task we have outlined here really is quite reasonable. A third response involves augmenting the probe data that is given to OXBOW when retrieving a skill concept. For example, instead of an empty joint-centered component, we might present the "naive" joint-centered schema, which consists of a single state description. This would encourage OXBOW to classify the probe with a skill concept that has at least some joint-centered information. The last two approaches will require further research, and we feel the first set of responses are inappropriate. In the final

chapter, we return to the issues discussed in this section and outline the approach we intend to follow toward correcting this problem.

## 7.5 Conclusions

The studies presented in this chapter were designed to demonstrate MÆANDER's overall ability to generate movements previously acquired through observation, and to improve its generation based upon practice. The results from these studies certainly demonstrated this ability, although they also revealed a few problems.

In summary, we made four general claims in this chapter that were supported by the experimental results. First we argued that MAGGIE's system parameters are not overly sensitive to particular settings. That is, the model is not dependent upon one particular combination of values in order to function properly. Second, we showed that MAGGIE exhibits a speed-accuracy tradeoff that is consistent with the appropriate results reported in the literature on human motor behavior. We also showed how the learning critics in MAGGIE, in conjunction with OXBOW's concept formation mechanisms, reduced error in movement trajectories with increasing experience. This is crucial to a claim of improvement through practice. Finally, we demonstrated MÆANDER's richness as a psychological model of skill learning through comparisons to, and predictions about, human learning phenomena. Unfortunately, our last experiment indicated that MÆANDER could not achieve a high level of competence in the real-world domain of drawing cursive letters. However, the results did show some improvements, and we outlined our assessment of the problem and a number of possible solutions. All things considered, we view MÆANDER as a success, based especially on the results supporting the first four claims.

In Chapter 1 we stated our goal as the construction of a computational model of motor behavior that possessed several characteristics. Without going into details, the results reported in this chapter and the previous one certainly satisfy these goals. In our final chapter we return to these original issues and their implications for future research.





## CHAPTER 8

# Discussion

---

### 8.1 Introduction

In Chapter 1 we set our goal as the development of a computational model of human motor behavior that possessed certain characteristics. The most important characteristics were that the model should learn to recognize movements through observation and that it improve its generation of movements through practice. At this level of specification, we can say MÆANDER satisfies our goal. That is, in Chapter 6 we demonstrated that OXBOW learned to recognize various movements, and in Chapter 7 we showed that MAGGIE could generate and improve stored motor skills. However, we specified several characteristics in Chapter 1, and we should consider MÆANDER's accomplishments and weaknesses with respect to these characteristics.

In this chapter, we close our discussion of MÆANDER by reviewing the contributions and advances made by the model, and the shortcomings that became apparent. At the same time, we consider why MÆANDER fails to fully meet our expectations in some cases. This serves as a natural springboard for an outline of possible directions to take this work in the future. We discuss several of the many extensions and improvements that could be made to our system, and we close with a final evaluation of the model, its behavior, and its significance.

### 8.2 Contributions of Mæander

The research reported in this dissertation holds significance for the study of both machine learning and human motor behavior. The model builds on both fields and contributes to both in one way or another. In this section, we consider the major contributions of the research, particularly in the context of our initial goals outlined in Chapter 1.

An implicit requirement of our model of motor behavior is that it be formulated in computational terms, and MÆANDER certainly satisfies this requirement. But more importantly, we stated that a model of motor behavior should address both the recognition and generation of movement skills.

We have demonstrated this quality through OXBOW and MAGGIE, and we have integrated these modules in MÆANDER. Although MÆANDER does not consist of a single mechanism that handles both recognition and generation, neither are its two components tailored to individual tasks that have been spliced together. OXBOW handles all memory management tasks such as the storage, organization, and retrieval of movement knowledge encoded in the form of motor schemas. MAGGIE handles monitoring and error correction and it suggests changes to the memory structures managed by OXBOW. The rest of MÆANDER consists of an interface between these modules, and the system's sensors and effectors. This includes the parsing and interpolation mechanisms that are necessary to convert movements to schemas and visa versa. In order to evaluate the two facets of our primary goal – the recognition and generation of movement – we separated the tasks and issues and, consequently, we emphasized OXBOW and MAGGIE separately. More appropriately, MÆANDER should be thought of as a single computational architecture.

A second contribution of our model is that the representation of skills are sufficiently rich to describe both very simple and very complex movements. The simplest movement (i.e., a motionless limb) can be represented as a single state description, and an arbitrarily complex movement can be represented as a sequence of states that indicate zero crossings in velocity or acceleration. The artificial movements used in the experimental chapters reveal some of this continuum. The SLAP movement is very simple and short, consisting of three states in its parsed form on average, whereas the SALUTE movement averages around ten. Both movement concepts reside in memory at the same time without serious interference. This shows even more strongly that MÆANDER's representation and learning mechanisms are robust and flexible.

Another issue related to flexibility is that of generality. We described the domain our model would address as containing those movements in which the form of the trajectory was of primary importance. In contrast, much of the psychological work on human motor behavior concerns ballistic aiming movements. Such tasks are easy for the experimenter to control and vary in the laboratory, but they may have limited applicability to more complex skills. Likewise, a fair amount of work has been done in artificial intelligence on control problems like the pole-balancing task. Both of these approaches are useful for studying some issues, but they are not very interesting with respect to many real-world tasks, such as playing a violin or performing martial arts. The class of trajectory-following movements we have addressed should allow considerable breadth in the types and complexities of skills that MÆANDER can learn and perform. Although we do not claim that this class subsumes the others, we view MÆANDER as an important contribution in terms of the tasks addressed by computational models.

Finally, as one of our initial goals we wanted the behavior of our computational model to conform wherever possible to phenomena observed in humans. In the preceding chapter, we compared MÆANDER's behavior to a number of these phenomena and, in some cases, found that the match was quite good. MAGGIE accounted quite well for the speed-accuracy tradeoff, and a quantitative comparison of our model's behavior to the psychological model was quite strong. The model also accounted for the qualitative phenomenon of transfer of skill between limbs. Based on the structure of our model, we made a number of predictions about phenomena that are not widely reported but that might be observable in humans. One of these, the change in the speed-accuracy tradeoff with

practice, was later found to be supported in the literature. These successes give us confidence that MÆANDER represents an interesting computational model of complex human motor behavior.

In addition to providing a viable model of motor learning and performance, MÆANDER has made at least one other contribution: the extension of previous techniques for concept formation. When starting this research, we were not explicitly interested in issues of concept formation and looked for an off-the-shelf conceptual clustering system that could be used with our representation of schemas. As we found that there were none and considered various adaptations of existing methods to meet our needs, we confronted some fundamental problems in this subfield of machine learning. One issue involved finding partial matches between components in a new instance and those in a stored concept – particularly when the instance and concept may have different numbers of components. We believe our approach to this problem is an elegant one and that it has revealed an interesting correspondence between PART-OF and IS-A relationships in structured domains. In summary, it is the *collection* of contributions described above, intended or otherwise, that represents MÆANDER's most significant contribution. It is the first computational model to address such a range of tasks and issues that are relevant to researchers from several fields.

### 8.3 Limitations of the Model

Although MÆANDER makes a number of important contributions, like any theory or model, it is not without its faults. We see a number of issues or areas in which the model is lacking. One of these drawbacks involves OXBOW's generalization mechanism. In Chapter 6, we pointed out that this process appeared to be sensitive to the level of noise in the domain. Although the system found good matches between state descriptions, it had trouble finding the correct values for the individual states. This was not a significant problem and only increased error by a few percentage points (approximately five units on a 130 unit improvement). However, we did not expect this behavior and should look more closely to determine its cause.

Another issue, more an oversimplification than a weakness, involves the method of arm control. MAGGIE controls its simulated arm by setting the change in position for every time slice of the simulation. We claimed that this was a reasonable design based on supporting psychological results and available computational mechanisms. However, we feel it is important to connect the model to a real robot arm. This requires that we address the issues we ignored via the assumption, and it would provide an opportunity and motivation to have the model itself handle low-level control. We think this could be accomplished within the current framework. One approach would determine the rotational accelerations (and ultimately torques) from the velocity information that is specified and use this information to drive the arm. However, it may be desirable to directly represent the accelerations as part of the skill concepts in long-term memory. Representing the positions and velocities of the joints may be appropriate in the case of viewer-centered schemas, but perhaps joint-centered schemas should be specified in terms of rotational accelerations or torque. We anticipate that the general mechanisms used in MÆANDER will transfer to schemas that specify torques instead of positions, or to a hybrid situation that utilizes both representations.

In the previous chapter, we identified an issue involving the method *MEANDER* uses to associate viewer-centered and joint-centered information. This was exemplified in the system's failure to improve performance on the letter generation task through practice. However, an analysis of the problem showed that *OXBOW* was classifying probes as well as could be expected. Indeed, it was doing exactly what it was supposed to do – finding concepts in long-term memory that were similar to a given probe. The problem involved our formulation of the experimental task for the letter generation study. This task implicitly asked *OXBOW* to complete a pattern rather than find the best match. That is, we wanted *MEANDER* to retrieve the joint-centered information associated with a probe, but *OXBOW* was designed to find the best match to a probe. In Section 8.4, we consider several approaches to resolving this conflict between tasks.

We are also dissatisfied that *MEANDER* has a number of limitations as a model of human motor behavior. For each of the phenomena addressed and exhibited by *MEANDER*, there are many more that it cannot handle. For instance, the current model cannot account for the practice variability effect described in Chapter 2, although this is perhaps the least robust of the phenomena discussed there. Another limitation involves the tasks that *MEANDER* can address. Currently we have not applied the system to tasks that involve manipulating objects in the environment (e.g., shooting basketballs or juggling balls). Although these tasks are not strictly trajectory-following tasks, such as we have addressed, the model should be able to handle them. This is a limitation of the research that has been completed to date, rather than of the model itself. One other limitation in this context is *MEANDER*'s inability to address the many phenomena involving knowledge of results, that is, the qualitative feedback an agent receives after a movement that communicates the success or failure of the goal. Because the model has no goals, it cannot reason about their success or failure. This point brings us to the final limitation that we consider here.

*MEANDER* models movement recognition and generation, but it is independent of a rational agent. That is, recognizing a movement does not inherently provide useful high-level information and generating a movement does not directly allow the accomplishment of some higher-level goal. Instead, these behaviors (recognition and generation) must be merged into a cohesive plan. Constructing useful sequences of motor skills (learned and stored by *MEANDER*) should be handled by a higher-level planning mechanism that interacts with our model. Furthermore, some of the mechanisms in *MAGGIE*, included out of necessity, are more properly the responsibility of a higher-level mechanism. For example, monitoring is part of a more general attention process and should be under the conscious control of an agent attempting to accomplish a goal. If the agent has high confidence that the current action will be completed to its satisfaction, then it should attend to other issues. On the other hand, if an unfamiliar movement is necessary to accomplish one of the agent's goals, then it should pay close attention to the execution and take corrective measures as needed. This situation and the limitations discussed above suggest several directions for improvement.

## 8.4 Future Work

We have reviewed several areas in which *MEANDER* is limited as a useful model of motor control and learning. In discussing these limitations, we have touched on a number of directions for future

work. In this section, we elaborate on our responses to some of these limitations and present additional directions to extend the model. We view further work on MÆANDER as falling into two different areas. One area addresses problems and extends the capabilities of the system as a computational model, whereas the other addresses phenomena and tasks that pertain to MÆANDER as a psychological theory. In this section we consider each area in turn.

#### 8.4.1 Improving the Computational Model

Throughout this dissertation, we identified issues that implied relatively minor modifications to the model, but for one reason or another had not been implemented to date. For example, in Chapter 7 we encountered a problem in which errors were corrected more than once, thereby leading to overcorrections. We introduced mechanism envisioned by Pew (1974) that would share information between monitoring events so that this problem would not arise. There are numerous similar that would improve and clean up the model, but that would not modify its applicability. We also think of the first two limitations in the previous section – the problem with OXBOW's generalization problem and connecting MAGGIE to a real arm – as being of this sort. Both would be implementation changes within the current framework.

A more significant problem relates to MÆANDER's retrieval of joint-centered schemas. Above we discussed how the retrieval task for joint-centered schemas was distinct from the basic task of concept formation. There are several possible approaches one could take. First, MAGGIE's modifications to the joint-centered schema based on practice could be made directly to the long-term memory structure, rather than invoking OXBOW to store it appropriately. This might work in principle, but there would be no sharing of learned knowledge. Each node in the hierarchy would have to be trained separately, losing the benefit of generalizations. Another alternative would explicitly associate joint-centered schemas with particular viewer-centered schemas. This approach would provide greater flexibility by letting more than one viewer-centered schema index a single joint-centered schema. This could save memory space and speed the learning process, but it would require additional mechanisms to determine which joint-centered schema should be associated with a given viewer-centered schema. Finally, we could address the problem by providing different information in the probe. This would avoid OXBOW's current preference for retrieving a skill concept with an empty joint-centered schema. Each of these ideas has some merit and we will pursue them in our ongoing research on MÆANDER.

We see two other important directions to improve MÆANDER as a computational model. The first would extend the flexibility of the schema concepts constructed by OXBOW. Currently, a schema is based on a particular coordinate system (either Cartesian or local polar) and it is described as particular values within that system. There is no provision for specifying arguments to schemas that would let them apply in novel situations or over different ranges than in which they were originally acquired. One approach we will consider would include schema parameters as part of the structure of the skill concept. The parameters would provide a means to specify detailed information and the schema would represent the invariant structure of the movement, independent of the speed or orientation in which it is performed.

The final direction involves broadening the class of skills addressed to include objects in the environment besides the components of the arm itself. For example, we would like the model to learn and represent target skills such as darts. Schemas would represent not only the trajectories of the limbs, but also those of any objects involved in the skill. This would let MÆANDER manipulate objects and move the model closer to functioning in a complex environment.

#### 8.4.2 Improving the Psychological Model

As mentioned earlier, we also want to strengthen the psychological basis of our model, and one priority is to search the literature for phenomena regarding the observation of movements. The phenomena themselves will suggest changes to the model, depending on whether MÆANDER can account for them. This provides an exciting opportunity – finding phenomena that the model was not designed to explain but that are compatible with its behavior. We will also continue to explore the literature for phenomena pertaining to movement generation.

At the same time, we have already made several predictions about MÆANDER's behavior that need to be tested. For example, in Chapter 5 we briefly discussed mental practice and its effects on performance. Currently, MÆANDER has no means of accounting for this behavior. We should extend the model to include a "mind's eye" that could observe the mental rehearsal of a motor skill and provide feedback for MAGGIE to suggest revisions to the schema. The important feature here is that internal feedback is less accurate or useful for schema modifications. We could include a noise signal, but we want to avoid adding unnecessary baggage to the model. Instead, we will look for a principled reason for such degraded feedback. Another prediction was that practice early in the development of a viewer-centered schema could lead to slower learning, due to reinforcement of the partially learned viewer-centered schema. The predictions about MÆANDER's behavior are implicit predictions about human behavior. Testing these on the model may confirm our expectations or cause us to revise them.

In either case, the next step is to test such predictions on human subjects. MÆANDER has already demonstrated behavior that should be viewed as a prediction of human performance. For example, in Chapter 6 we showed that OXBOW made certain characteristic mistakes when classifying handwritten letters. The pattern of these errors was intuitively what we would expect humans to produce, but this has not been explicitly tested. This is an example of how the model can drive further psychological experimentation.

Finally, in Section 8.3 we mentioned the need for a planning mechanism if we wanted to account for phenomena pertaining to knowledge of results. We are currently attempting to integrate MÆANDER with a comprehensive cognitive architecture ICARUS (Langley et al., in press). This architecture includes a planning mechanism, a memory module analogous to OXBOW, and a mechanism that controls and generates drives. The drives provide the top-level goals for the planner, which in turn creates subgoals that are eventually executable by MÆANDER. The architecture is being developed with a simulated environment that supports three-dimensional objects that obey standard laws of physics. Such an integrated architecture would greatly expand the range of motor phenomena that MÆANDER can explain.

## 8.5 Closing

In the previous pages we have described MÆANDER, a computational model of motor performance and learning. The model addresses both the recognition of observed movements and the generation of such movements. Motor skills are acquired in a natural progression, starting with observations of another agent performing a skill and continuing with improvements to this acquired representation through practice.

We evaluated MÆANDER both as a computational model and as a psychological model. We demonstrated both aspects of the system's behavior through numerous experiments, including studies in the domain of cursive lettering. The model accounted for a number of phenomena observed in human behavior, and it made several interesting and testable predictions.

MÆANDER represents a significant contribution to two fields: machine learning and human motor behavior. The system's memory management component, OXBOW, extends the techniques of concept formation in new and interesting ways. As a computational model satisfying the conjunction of characteristics in Chapter 1, MÆANDER serves as an initial bridge between low-level control and high-level planning mechanisms, as well as psychological and computational models of motor control. Much work still remains, but the current system constitutes clear progress in our understanding of motor skills and their acquisition.





## References

---

- Adams, J. A. (1971). A closed-loop theory of motor learning. *Journal of Motor Behavior*, 3, 111-149.
- Aha, D. W. (1990). Doctoral dissertation, Department of Information & Computer Science, University of California, Irvine.
- Anderson, J. R., & Matessa, M. (1991). An iterative Bayesian algorithm for categorization. In D. H. Fisher, M. J. Pazzani, & P. Langley (Eds.), *Concept formation: Knowledge and experience in unsupervised learning*. San Mateo, CA: Morgan Kaufmann.
- Andrae, P. M. (1985). *Justified generalization: Acquiring procedures from examples*. Doctoral dissertation, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA.
- Arbib, M. A. (1982). Modeling neural mechanisms of visuomotor coordination in frog and toad. In S. Amari & M. A. Arbib (Eds.), *Lecture notes in biomathematics* (Vol. 45). Berlin: Springer-Verlag.
- Armstrong, T. R. (1970). *Training for the production of memorized movement patterns* (Technical Report No. 26). Ann Arbor: University of Michigan, Human Performance Center.
- Atkeson, C. G. (1987). *Roles of knowledge in motor learning*. Doctoral dissertation, Department of Brain and Cognitive Sciences, Massachusetts Institute of Technology, Cambridge, MA.
- Bartlett, F. C. (1958). *Thinking: An experimental and social study*. London: Allen & Unwin.
- Cheeseman, P., Kelly, J., Self, M., Stutz, J., Taylor, W., & Freeman, D. (1988). AUTOCLASS: A Bayesian classification system. *Proceedings of the Fifth International Conference on Machine Learning* (pp. 54-64). Ann Arbor, MI: Morgan Kaufmann.
- Connell, M. E., & Utgoff, P. E. (1987). Learning to control a dynamic physical system. *Proceedings of the Sixth National Conference on Artificial Intelligence* (pp. 456-460). Seattle, WA: Morgan Kaufmann.
- Crossman, E. R. F. W. (1959). A theory of the acquisition of speed-skill. *Ergonomics*, 2, 153-166.
- Ells, J. G. (1969). *Attentional requirements of movement control*. Doctoral dissertation, University of Oregon.

- Davis, W. J. (1976). Organizational concepts in the central motor networks of invertebrates. In R. M. Herman, S. Grillner, P. S. G. Stein, & D. G. Stuart (Eds.), *Neural control of locomotion*. New York: Plenum Press.
- Fikes, R. E., Hart, P. E., & Nilsson, N. J. (1972). Learning and executing generalized robot plans. *Artificial Intelligence*, *3*, 251-288.
- Fitts, P. M., & Peterson, J. R. (1964). Information capacity of discrete motor responses. *Journal of Experimental Psychology*, *67*, 103-112.
- Fisher, D. H. (1987). Knowledge acquisition via incremental conceptual clustering. *Machine Learning*, *2*, 139-172.
- Fisher, D. H., & Langley, P. (1990). The structure and formation of natural categories. In G. H. Bower (Ed.), *The psychology of learning and motivation: Advances in research and theory* (Vol. 26). Cambridge, MA: Academic Press.
- Fisher, D., & Pazzani, M. (1991). Computational models of concept learning. In D. H. Fisher, M. J. Pazzani, & P. Langley (Eds.), *Concept formation: Knowledge and experience in unsupervised learning*. San Mateo, CA: Morgan Kaufmann.
- Foley, J. D., & van Dam, A. (1982). *Fundamentals of interactive computer graphics*. Reading, MA: Addison-Wesley.
- Fried, L. S., & Holyoak, K. J. (1984). Induction of category distributions: A framework for classification learning. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *10*, 234-257.
- Frohlich, D. M., & Elliott, J. M. (1984). The schematic representation of effector function underlying perceptual-motor skill. *Journal of Motor Behavior*, *16*, 40-60.
- Gallwey, W. T. (1974). *The inner game of tennis*. New York: Random House.
- Gennari, J. H. (1990). *An experimental study of concept formation*. Doctoral dissertation, Department of Information & Computer Science, University of California, Irvine.
- Gennari, J. H., Langley, P., & Fisher, D. (1989). Models of incremental concept formation. *Artificial Intelligence*, *40*, 11-61.
- Gluck, M., & Corter, J. (1985). Information, uncertainty and the utility of categories. *Proceedings of the Seventh Annual Conference of the Cognitive Science Society* (pp. 283-287). Irvine, CA: Lawrence Erlbaum.
- Hardy, S. (1984). Robot control systems. In T. O'Shea & M. Eisenstadt (Eds.), *Artificial intelligence: Tools, techniques, and applications*. New York: Harper & Row.
- Hollerbach, J. (1979). Understanding manipulator control by synthesizing human handwriting. In P. H. Winston & R. H. Brown (Eds.), *Artificial intelligence: An MIT perspective*. Cambridge, MA: MIT Press.
- Iba, W. (1991). Learning to classify observed motor behavior. *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence*. Sydney: Morgan Kaufmann.

- Iba, W. (1989). *Human motor behavior: A short review of phenomena, theories, and systems* (Tech. Rep. No. 89-34). Irvine: University of California, Department of Information & Computer Science.
- Iba, W., & Gennari, J. H. (1991). Learning to recognize movements. In D. H. Fisher, M. J. Pazzani, & P. Langley (Eds.), *Concept formation: Knowledge and experience in unsupervised learning*. San Mateo, CA: Morgan Kaufmann.
- Iba, W., & Langley, P. (1987). A computational theory of motor learning. *Computational Intelligence*, *3*, 338-350.
- Iba, W., Wogulis, J., & Langley, P. (1988). Trading off simplicity and coverage in incremental concept learning. *Proceedings of the Fifth International Conference on Machine Learning* (pp. 73-79). Ann Arbor, MI: Morgan Kaufmann.
- Kelso, J. A. S. (1982). Concepts and issues in human motor behavior: coming to grips with the jargon. In J. A. S. Kelso (Ed.), *Human motor behavior: An introduction*. Hillsdale, NJ: Lawrence Erlbaum.
- Kibler, D., & Langley, P. (1988). Machine learning as an experimental science. *Proceedings of the Third European Working Session on Learning* (pp. 81-92). Glasgow: Pitman.
- Klapp, S. T., Anderson, W. G., & Berrian, R. W. (1973). Implicit speech in reading, reconsidered. *Journal of Experimental Psychology*, *100*, 368-374.
- Laird, J. E., Hucka, M., Yager, E. S., & Tuck, C. M. (1990). Correcting and extending domain knowledge using outside guidance. *Proceedings of the Seventh International Conference on Machine Learning* (pp. 235-243). Austin: Morgan Kaufmann.
- Langley, P. (1987). Research papers in machine learning. *Machine Learning*, *2*, 195-198.
- Langley, P. (1985). Learning to search: From weak methods to domain-specific heuristics. *Cognitive Science*, *9*, 217-260.
- Langley, P., Gennari, J. H., & Iba, W. (1987). Hill-climbing theories of learning. *Proceedings of the Fourth International Workshop on Machine Learning* (pp. 312-323). Irvine, CA: Morgan Kaufmann.
- Langley, P., Thompson, K., Iba, W., Gennari, J. H., & Allen, J. A. (in press). An integrated cognitive architecture for autonomous agents. In W. V. De Velde (Ed.), *Representation and learning in autonomous agents*. Amsterdam: North Holland.
- Levinson, R. A. (1985). *A self-organizing retrieval system for graphs*. Doctoral dissertation, Department of Computer Sciences, University of Texas, Austin.
- Mason, M. T., Christiansen, A. D., & Mitchell, T. M. (1989). Experiments in robot learning. *Proceedings of the Sixth International Workshop on Machine Learning* (pp. 141-145). Ithaca, NY: Morgan Kaufmann.
- Massaro, D. W. (1975). *Experimental psychology and information processing*. Chicago: Rand McNally.
- McCracken, H. D., & Stelmach, G. E. (1977). A test of the schema theory of discrete motor learning. *Journal of Motor Behavior*, *9*, 193-201.

- Mel, B. W. (1988). Building and using mental models in a sensory-motor domain: a connectionist approach. *Proceedings of the Fifth International Conference on Machine Learning* (pp. 207-213). Ann Arbor, MI: Morgan Kaufmann.
- Melville, D. S. (1976). *A test of motor schema theory: Performance of a rapid movement task in the absence of knowledge of results*. Doctoral dissertation, University of Iowa.
- Meyer, D. E., Abrams, R. A., Kornblum, S., Wright, C. E., & Smith, J. E. K. (1988). Optimality in human motor performance: Ideal control of rapid aimed movements. *Psychological Review*, *95*, 340-370.
- Meyer, D. E., Smith, J. E. K., Kornblum, S., Abrams, R. A., & Wright, C. E. (1990). Speed-accuracy tradeoffs in aimed movements: Toward a theory of rapid voluntary action. In M. Jeannerod (Ed.) *Attention and performance XIII: Motor representation and control*. Hillsdale, NJ: Lawrence Erlbaum.
- Meyer, D. E., Smith, J. E. K., & Wright, C. E. (1982). Models for the speed and accuracy of aimed movements. *Psychological Review*, *89*, 5, 449-482.
- Michie, D. & Chambers, R. (1968). BOXES: An experiment in adaptive control. In E. Dale & D. Michie (Eds.), *Machine intelligence 2*. Edinburgh: Oliver & Boyd.
- Mitchell, T. M., Utgoff, P. E., & Banerji, R. (1983). Learning by experimentation: Acquiring and refining problem-solving heuristics. In R. S. Michalski, J. G. Carbonell, & T. M. Mitchell (Eds.), *Machine learning: An artificial intelligence approach*. San Mateo, CA: Morgan Kaufmann.
- Moore, A. W. (1990). Acquisition of dynamic control knowledge for a robot manipulator. *Proceedings of the Seventh International Conference on Machine Learning* (pp. 235-243). Austin, TX: Morgan Kaufmann.
- Newell, A. & Rosenbloom, P. S. (1981). Mechanisms of skill acquisition and the law of practice. In J. R. Anderson (Ed.), *Cognitive skills and their acquisition*. Hillsdale, NJ: Lawrence Erlbaum.
- Pew, R. W. (1966). Acquisition of hierarchical control over the temporal organization of a skill. *Journal of Experimental Psychology*, *71*, 764-771.
- Pew, R. W. (1974). Human perceptual-motor performance. In B. H. Kantowitz (Ed.), *Human information processing: Tutorials in performance and cognition*. New York: Lawrence Erlbaum.
- Posner, M. I., & Keele, S. W. (1968). On the genesis of abstract ideas. *Journal of Experimental Psychology*, *90*, 1-7.
- Powers, W. T. (1973). Feedback: Beyond behaviorism. *Science*, *179*, 351-356.
- Raibert, M. (1976). A state space model for sensorimotor control and learning (Tech. Rep. No. AIM-351). Cambridge, MA: Massachusetts Institute of Technology, Laboratory for Artificial Intelligence.
- Rosenbaum, D. A. (1991). *Human motor control*. San Diego: Academic Press.
- Rosenbaum, D. A. (1985). Motor programming: A review and scheduling theory. In H. Heuer, U. Kleinbeck, & K.-H. Schmidt (Eds.), *Motor behavior: Programming, control, and acquisition*. Berlin: Springer-Verlag.

- Rosenbaum, D. A. (1977). Selective adaptation of "Command Neurons" in the human motor system. *Neuropsychologia*, *15*, 81-91.
- Rosenbloom, P. S. (1986). The chunking of goal hierarchies. In J. Laird, P. Rosenbloom, & A. Newell (Eds.), *Universal subgoalting and chunking*. Boston: Kluwer Academic Publishers.
- Rosenbloom, P., & Newell, A. (1987). Learning by chunking: A production system model of practice. In D. Klahr, P. Langley, & R. Neches (Eds.) *Production system models of learning and development*. Cambridge, MA: MIT Press.
- Rubin, J. M., & Richards, W. A. (1985). *Boundaries of visual motion* (Tech. Rep. No. AIM-835). Cambridge, MA: Massachusetts Institute of Technology, Laboratory for Artificial Intelligence.
- Sammut, C., & Banerji, R. B. (1986). Learning concepts by asking questions. In R. S. Michalski, J. G. Carbonell, & T. M. Mitchell (Eds.), *Machine learning: An artificial intelligence approach* (Vol. 2). San Mateo, CA: Morgan Kaufmann.
- Schmidt, R. A. (1982a). More on motor programs. In J. A. S. Kelso (Ed.), *Human motor behavior: An introduction*. Hillsdale, NJ: Lawrence Erlbaum.
- Schmidt, R. A. (1982b). The schema concept. In J. A. S. Kelso (Ed.), *Human motor behavior: An introduction*. Hillsdale, NJ: Lawrence Erlbaum.
- Schmidt, R. A. (1975a). *Motor skills*. New York: Harper & Row.
- Schmidt, R. A. (1975b). A schema theory of discrete motor skill learning. *Psychological Review*, *82*, 225-260.
- Schmidt, R. A., & White, J. L. (1972). Evidence for an error detection mechanism in motor skills: A test of Adams' closed-loop theory. *Journal of Motor Behavior*, *4*, 143-154.
- Schmidt, R. A., Zelaznik, H., Hawkins, B., Frank, J. S., & Quinn, J. T. Jr. (1979). Motor-output variability: A theory for the accuracy of rapid motor acts. *Psychological Review*, *86*, 415-451.
- Schmidt, R. A., Sherwood, D. E., Zelaznik, H. N., & Leikind, B. J. (1985). Speed-accuracy trade-offs in motor behavior: Theories of impulse variability. In H. Heuer, U. Kleinbeck, & K.-H. Schmidt (Eds.), *Motor behavior: Programming, control, and acquisition*. Berlin: Springer-Verlag.
- Segen, J. (1990). Graph clustering and model learning by data compression. *Proceedings of the Seventh International Conference on Machine Learning* (pp. 93-101). Austin, TX: Morgan Kaufmann.
- Segre, A. M. (1987). *Explanation-based learning of generalized robot assembly plans* (Tech. Rep. No. UILU-ENG-2208). Urbana: University of Illinois at Urbana-Champaign, Department of Electrical and Computer Engineering.
- Selfridge, O. G., Sutton, R. S., & Barto, A. G. (1985). Training and tracking in robotics. *Proceedings of the Ninth International Joint Conference on Artificial Intelligence* (pp. 670-672). Los Angeles, CA: Morgan Kaufmann.
- Smith, E. E., & Medin, D. L. (1981). *Categories and concepts*. Cambridge, MA: Harvard University Press.
- Snoddy, G. S. (1926). Learning and stability. *Journal of Applied Psychology*, *10*, 1-36.

- Sperling, G. (1960). The information available in brief visual presentations. *Psychological Monographs*, 74 (11, Whole No. 498).
- Stelmach, G. E. (1982). Motor control and motor learning: the closed-loop perspective. In J. A. S. Kelso (Ed.), *Human motor behavior: An introduction*. Hillsdale, NJ: Lawrence Erlbaum.
- Stelmach, G. E., Kelso, J. A. S., & Wallace, S. A. (1975). Preselection in short-term motor memory. *Journal of Experimental Psychology: Human Learning and Memory*, 1, 745-755.
- Sugden, D. A. (1980). Movement speed in children. *Journal of Motor Behavior*, 12, 125-132.
- Sussman, G. J. (1975). *A computer model of skill acquisition*. New York: American Elsevier.
- Sutton, R. S. (1984). *Temporal credit assignment in reinforcement learning*. Doctoral dissertation, Department of Computer and Information Science, University of Massachusetts, Amherst.
- Swartz, N. M. (1984). Arm dynamics simulation. *Journal of Robotic Systems*, 1, 83-100.
- Thompson, K., & Langley, P. (1991). Concept formation in structured domains. In D. H. Fisher, M. J. Pazzani, & P. Langley (Eds.), *Concept formation: Knowledge and experience in unsupervised learning*. San Mateo, CA: Morgan Kaufmann.
- Waltz, D. (1975). Understanding line drawings of scenes with shadows. In P. H. Winston (Ed.), *The psychology of computer vision*. New York: McGraw-Hill.
- Wiener, N. (1948). *Cybernetics: Control and communication in the animal and the machine*. New York: John Wiley.
- Wylie, C. R. (1975). *Advanced engineering mathematics* (4th ed.). New York: McGraw-Hill.
- Zelaznik, H. N. (1977). Transfer in rapid timing tasks: An examination of the role of variability in practice. In D. M. Landers & R. W. Christina (Eds.), *Psychology of motor behavior and sport* (Vol. 1). Champaign, IL: Human Kinetics.











**FIA-91-27**

*Constraint-Based Scheduling*

MONTE ZWEBEN

September 1991

The GERRY scheduling system developed by NASA Ames with assistance from the Lockheed Space Operations Company, and the Lockheed Artificial Intelligence Center, uses a method called constraint-based iterative repair. Using this technique, one encodes both hard rules and preference criteria into data structures called constraints. GERRY repeatedly attempts to improve schedules by seeking repairs for violated constraints. The system provides a general scheduling framework which is being tested on two NASA applications. The larger of the two is the Space Shuttle Ground Processing problem which entails the scheduling of all the inspection, repair, and maintenance tasks required to prepare the orbiter for flight. The other application involves power allocation for the NASA Ames wind tunnels. Here the system will be used to schedule wind tunnel tests with the goal of minimizing power costs. In this paper, we describe the GERRY system and its application to the Space Shuttle problem. We also speculate as to how the system would be used for manufacturing, transportation, and military problems.

---

**FIA-91-28**

*Introduction to IND and Recursive Partitioning*

WRAY BUNTINE AND RICH CARUANA

October 1991

This manual describes the IND package for learning tree classifiers from data. The package is an integrated C and C shell re-implementation of tree learning routines such as CART, C4, and various MDL and Bayesian variations. The package includes routines for experiment control, interactive operation, and analysis of tree building. The manual introduces the system and its many options, gives a basic review of tree learning, contains a guide to the literature and a glossary, lists the manual pages for the routines, and instructions on installation.

---

**FIA-91-29**

*Acquisition and Improvement of Human Motor Skills: Learning Through Observation and Practice*

WAYNE IBA

November 1991

Skilled movement is an integral part of the human existence. A better understanding of motor skills and their development is a prerequisite to the construction of truly flexible intelligent agents. We present MÆANDER, a computational model of human motor behavior, that uniformly addresses both the acquisition of skills through observation and the improvement of skills through practice. MÆANDER consists of a sensory-effector interface, a memory of movements, and a set of performance and learning mechanisms that let it recognize and generate motor skills. The system initially acquires such skills by observing movements performed by another agent and constructing a concept hierarchy. Given a stored motor skill in memory, MÆANDER will cause an effector to behave appropriately. All learning involves changing the hierarchical memory of skill concepts to more closely correspond to either observed experience or to desired behaviors. We evaluate MÆANDER empirically with respect to how well it acquires and improves both artificial movement types and handwritten script letters from the alphabet. We also evaluate MÆANDER as a psychological model by comparing its behavior to robust phenomena in humans and by considering the richness of the predictions it makes.

---

