INTEGRATING KNOWLEDGE-BASED SYSTEM AND NEURAL NETWORK TECHNIQUES FOR ROBOTIC SKILL ACQUISITION

David A. Handelman, Stephen H. Lane, and Jack J. Gelfand*
Human Information Processing Group
Department of Psychology
Princeton University
Princeton, New Jersey 08544

ABSTRACT

This paper describes an approach to robotic control that is patterned after models of human skill acquisition. The intent is to develop robots capable of learning how to accomplish complex tasks using designer-supplied instructions and self induced practice. A simulation is presented in which a rule-based system supervises the training of a neural network and controls the operation of the system during the learning process. Simulation results show the interaction between rule-based and network-based system components during various phases of training and supervision.

INTRODUCTION

Neural networks have been shown to be very efficient at learning from experience. However, there area number of problems to be overcome before they become useful components of truly autonomous learning systems. Presently, training data must be supplied by an outside operator who must closely supervise the learning process. Also, if the system containing the neural network is required to perform its task during learning, a provision must be made to control its operation during this period. We have developed an automated approach to solving these problems in which a rule based system supervises the training of a neural network and controls the operation of the system during the learning process. The operation of this system exhibits properties which are similar to the stages of motor learning in humans in which control shifts from an explicit, verbally oriented representation to an implicit reflexive representation and execution.

For a preliminary demonstration of these concepts, a simulation has been constructed in which a two link manipulator is taught how to make a tennis-like swing. A schematic of this approach is illustrated in Fig. 1. The control system first determines how to make a successful swing using rules alone. It then teaches a neural network how to accomplish this task by having the network observe and generalize on the rule-based task execution. Following initial training, a rule based execution monitor evaluates the neural network performance, and re-engages rule-based swing-maneuver control whenever errors due to changes in the manipulator or its operating environment necessitate retraining of the network. The rule-based system thereby ensures proper task completion while neural network re-learning takes place. The simulation shows the interaction betweeen rule-based and network based system components during various phases of training and supervision.

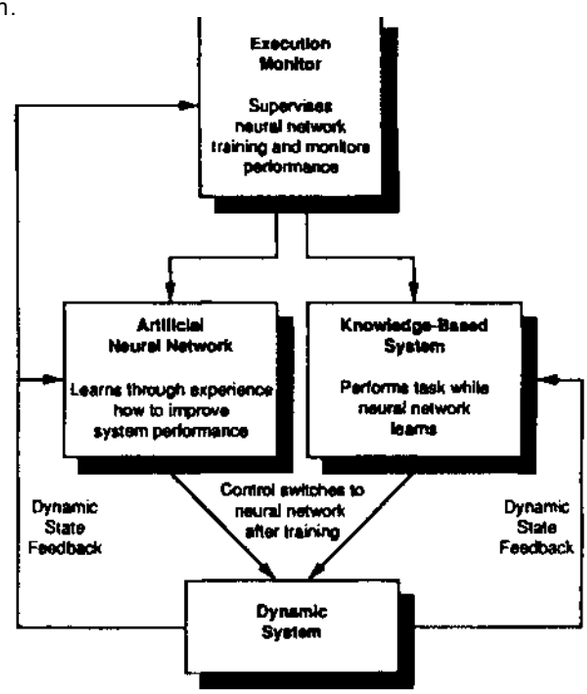

Figure 1. Integration of Neural Network and Knowledge-Based System Techniques for Hierarchical and Modular Control of Dynamic Systems and Learning

PHASES OF SKILL ACQUISITION

It is well known that humans pass through various levels of competence as motor skills are acquired. Fitts and Posner [1967] define three phases of skill learning: 1) the cognitive (early) phase, wherein a beginner tries to understand the task, 2) the associative (intermediate) phase, where patterns of response emerge and gross errors are eliminated, and 3) the autonomous (final) phase, when task

* This work was supported by the James S. McDonnell Foundation.

execution requires little cognitive control. Adams f1971] defines two stages of motor learning: 1) the verbal-motor stage, where corrections are based on verbal descriptions of how well the task is being accomplished, and 2) the motor stage, where conscious behavior eventually becomes automatic, and attentional mechanisms pick out only those channels of information relevant to learning. Brooks [1986] points out that there are numerous structures in the brain associated with motor learning. Centers involved in emotion, attention, memory and motor control all participate. Within this system the cerebellum is particularly concerned with coordination and skill acquisition. Although it is not well understood at this time whether control of motor learning specifically passes between various structures in the motor system it is known, however, that fine motor control cannot be learned or executed without the participation of the cerebellum.

The distinctions between various stages of motor learning and control are also found in more general cognitive skill acquisition and memory. Kupfermann [1985] differentiates between reflexive and declarative memory and learning. Task execution based on reflexive memory and learning relate specific responses to specific stimuli. Learned functions appear automatic and require little or no thought. Declarative memory and learning can usually be expressed in verbal statements. Execution of tasks based on declarative memory usually requires evaluation, comparison and inference. Most complex tasks attempted for the first time usually require some form of declarative reasoning.

Anderson [1982] has modelled the transition between the cognitive and associative stagesofFittsand Posnerf 1967] as a knowledge compilation process whereby a more general declarative form of task knowledge is converted into a specific procedural representation. During the transition from declarative to procedural processing, operational production rules become more condensed and task specific. In Anderson's model the final shift to the most autonomous form of processing is accomplished by fine tuning the selectivity of the problem solving search procedures.

A quasi-neural model for the development of automatic processing accompanying skill acquisition has been proposed by Schneider [1985] and Schneider and Detweiler [1987]. In this, model a shift in the control mechanism changes the gating of information vectors between visual, lexical, semantic, and motor processing units. In Schneider's model the initial stage of cognitive processing involves direct control of information between units. During practice, associative learning enables direct association of input/output vectors pairs and priority learning determines the transmission strength of these vectors between processing modules. A gradual transition from controlled to automatic processing occurs with practice. In the final and most automatic stage of skill acquisition, communication between processing units is not gated by the controller butisdetermined soley by the transmission strengths learned during the priority learning process.

In the model presented by Anderson, knowledge representations and functional architectures remain the same as the system goes through various stages of skill acquistion. In Schneider's model, the interconnections and processing modules remain the same, but a change in the control of information flow between units causes the execution to shift from a serial to a more parallel processing mode. In this paper we explore the properties of a system in which both control and learning are shifted between two differentfunctional architectures with differentrepresentations of the control problem. We shall use the terminology, declarative and reflexive control, patterned after Kupfermann [1985] to describe the shift between knowledge-based system and neural network control because it most nearly represents the differences between these two approaches.

Because the rule base used for the control of the arm is quite task specific, the system described here corresponds most closely in its operation to the shift in Anderson's procedural phase represented by the rule-based system to an automatic phase represented by the neural network. We plan to implement in the near future a more complex rule-based system which also exhibits the shift from a declarative representation to procedural processing in Anderson's terminology. In the work reported here, we made no attempt to optimize the design or performance of the individual knowledge-based system or neural network components but rather used simplified versions of both to highlight the properties of the integrated system.

It is important to note that the shift of processing from controlled to automatic is a central component of the allocation of resources in human processing. Tasks initially learned declaratively become reflexive through repetition. In this way tasks performed most often are processed in a way that requires the smallest allocation of processing resources. However, when familiar tasks are attempted in novel situations, there must be a shift back into a declarative form of processing in order to accomplish the task. For example, although one may become adroit at tying one's own necktie, tying someone else's necktie requires some thought due the change in perspective. The intent of the present research is to endow robots with a similar capability, combining reflexive and declarative forms of information processing for highly adaptive and dexterous robotic control.

INTEGRATING KNOWLEDGE-BASED SYSTEMS AND NEURAL NETWORKS

Artificial neural networks and knowledge-based systems represent the desired reflexive and declarative forms of processing, respectively. Artificial neural networks exhibit characteristics of associative memory, pattern matching, generalization, and learning by example [Rumelhartand McClelland, 1986, Fahlman and Hinton,1987]. Insofar as they direcdy compute specific outputs in response to specific inputs, neural networks can be viewed as imple-

meriting reflexive task knowledge. Knowledge-based systems have demonstrated the ability to encode and exercise expert knowledge within limited domains, providing a hierarchical software organization and explainable solutions [Charniak and McDermott,1985]. The inferencing capabilities of these systems effectively implement declarative task knowledge.

The control task chosen to illustrate the integration technique involves teaching a two-link manipulator how to make a tennis-like swing. The swing eventually will become part of a higher-level task that involves striking a ball to hit a specific place on a wall. Presently, only the swing is considered. The manipulator, shown in Fig. 2, is composed of two cylindrical links, each with a length of 1 m and a radius of 0.1 m. The inner link, located between the shoulder and elbowjoints, has a mass of 1 kg, and the outer link mass varies between 0.5 and 1 kg in the simulations described below. Rotation of the shoulder and elbowjoints occurs in a vertical plane, against gravity. The overall task of the control system is to vary joint torques over time such that the outer link moves from its vertical resting position (pointing straight down) through an imaginary contact point. The outer link is to strike the contact point at the link's mid-point, and at a specified angle (slope). A swing is deemed successful when the two swing-maneuver performance measures "Contact Position Error and Contact Slope Error" fall within specified bounds. A more complete description of the system and its performance is given by Handelman et. al. [1989].

Figure 3 shows a block diagram of the controller used to accomplish this task. Three levels of a controller task hierarchy are defined: low-level reflexes, a reflex modulator, and an execution monitor. On the lowest level (the innermost loop in the block diagram), each joint is endowed with simple reflexes that command torque as a function of angular position and velocity. The term "reflex" is borrowed from human motor control and is meant to represent cither a simple position or force servo system. It is believed that many human voluntary motions involve the modulation of reflexes [Stein and Capady,1988]. Therefore the middle loop of Fig. 3 is used to provide the low-level reflex loops with time-varying gains and commands needed to carry out the swing maneuver. Finally, the execution monitor supervises various phases of

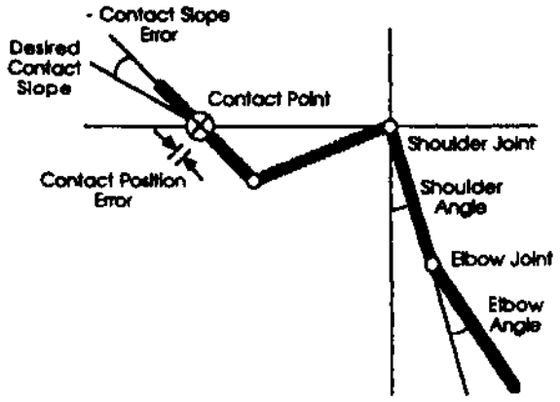

Figure 2. Two-Link Manipulator

reflex modulation learning.These levels of the controller hierarchy are described in detail below.

Rule-Based Swing Reflex Modulator

On the lowest level, each joint reflex acts as a hybrid position/torque servomechanism. The two available reflexes are 1) spring-like position control with joint angle and angular velocity feedback, and 2) constant torque control. Reflex gain and command modulation is used to generate the manipulator swinging motion. Implemented in the middle feedback loop of Fig. 3, modulation is accomplished using either rules, a neural network, or both, depending on the mode of operation. Table 1 presents the rules used to execute a rule-based swing. The rules are invoked by a cyclic goal-directed search through a knowledge base of parameters, rules, and procedures in much the same way that aircraft emergency procedures are executed by Handelman and Stengel [1988,1989]. The rules were obtained by the control system designer through trial and error. No explicit knowledge of the dynamics was used in this process. A single base torque is used and is scaled for each joint by an appropriate parameter.

Network-Based Swing Reflex Modulator

Neural-network-based reflex modulation is performed with a CMAC (Cerebellar Model Articulation Controller) network module developed by Albus [1975]. A CMAC is used to encode implicitly, through learning, the sensorimotor relationships represented explicidy by the swing-maneuver rules of Table 1. The CMAC module is a

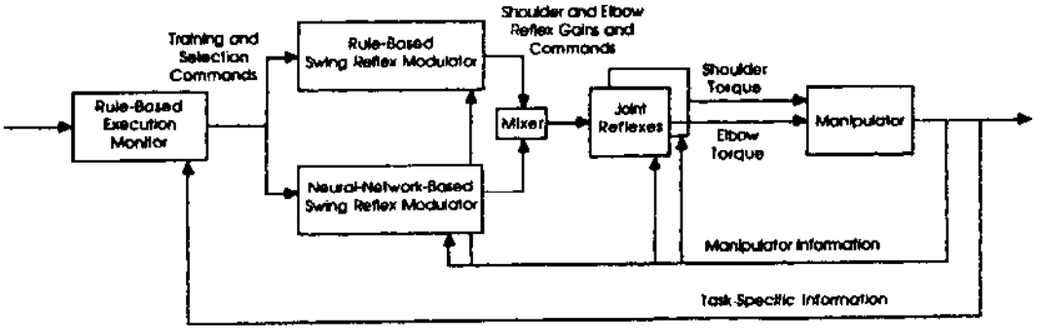

Figure 3. Block Diagram of Manipulator Control System

Swing Step 1

Stiffen shoulder about 20 deg with gain 0.5 and
stiffen elbow about 0 deg with gain 0.1
until shoulder angle >= 10 deg

Swing Step 2

Torque shoulder with gain -1.0
until shoulder angle <= -65 deg

Swing Step 3

Torque shoulder with gain 0.5 and
stiffen elbow about -60 deg with gain 0.2
until shoulder angle >= -45 deg

Swing Step 4

Torque shoulder with gain -1.3 and
stiffen elbow about 0 deg with gain 0.1
until shoulder and elbow velocities are below 50 deg/sec

Swing Step 5

Stiffen shoulder and elbow about 0 deg with gain 1.0
until magnitudes of shoulder and elbow velocities are
below 20 deg/sec

Table 1. Rule-Based Swing Maneuver Steps

perceptron-like table look-up technique for reproducing functions with multiple input and output variables over particular regions of the function space. A CMAC can learn by example, and its mapping scheme provides automatic generalization (interpolation) between input states, so that similar inputs produce similar outputs.

The CMAC network modules are used to implement a reflexive swing maneuver. With the CMAC properly trained, input values representing the manipulator state cause the CMAC to output gains and commands to the low-level reflexes that ultimately move the manipulator into the next task-dependent area of the manipulator state space. The resultant manipulator state then defines new input values which, in turn, produce CMAC outputs that again "push" the manipulator into the nextdesired state. Thus, after the CMAC network has been trained, rules initiate a network-based swing by "throwing" the manipulator into an area of the manipulator state space recognized by the network. The network then reflexively executes the swing maneuver. When the manipulator finally enters a configuration recognized by rules as concluding the swing maneuver, the rules disengage the CMAC network and once again take over reflex modulation responsibility. Training and supervision of CMAC reflex modulation are performed by the execution monitor.

Rule-Based Execution Monitor

The execution monitor is entirely rule-based, and implements three Swing Modes that correspond to various phases of reflex modulation learning. The Swing Modes are termed Rule-Based, Rule-and-Network-Based, and Network-Based. Under Rule-Based Swing Mode, reflex modulation is carried out by rules only. When certain conditions are met, the CMAC network modules concurrently attempt to duplicate the rule-based reflex modulation commands, learning by example. Under Rule-and-Network-Based Swing Mode, rule-based and CMAC-based reflex modulation suggestions are generated in parallel, and CMAC learning continues. CMAC-based outputs are fed directly to the reflexes when they are deemed close enough to the "optimal" rule-based command suggestions. Finally, under Network-Based Swing Mode, CMAC learning ceases, and CMAC outputs direcdy modulate shoulder and elbow joint reflexes throughout the entire swing maneuver. Switching between Swing Modes is dictated by the rules shown in Table 2.

SIMULATION RESULTS

Operation of the integrated swing-maneuver controller of Fig. 3 was simulated on a personal computer equipped with a 10-MHz 80286 processor, an 8-MHz 80287 math coprocessor, and 640 KBytes of Random-Access Memory (RAM). Source code for the CMAC modules was written in Pascal, whereas the rule-based system components (including 77 rules) were translated automatically from LISP to Pascal as described Handelman and Stengel [1988]. Following a discussion of simulation results detailing the operation of the controller, real-time performance issues will be addressed.

Figure 4 depicts the activity of the rule-based system components during various phases of the swing maneuver. Each data point in the top plot corresponds to the number of rules tested by the controller inference engine during a goal-directed search cycle. Adjacent data points are separated by the amount oftime required to complete a search cycle, in this case forced to coincide with a controller sample rate of 20 per sec. At the start of the simulation, the manipulator outer link mass is set at 0.5 kg, and the Base Torque command is set at 3 Nm.

The controller performs swings at 5 sec intervals. It begins swinging at t= 5 sec while in Rule-Based Swing Mode. Peaks and valleys in the number of rules tested reflect execution of the various swing steps of Table 1. As shown by the plot of Swing Contact Error in Fig. 4, errors associated with the first 3 swings remain high. The outer link never reaches the contact point because the applied torques do not overcome gravity by a sufficient amount, prompting successive increases in Base Torque. By t =20 sec, a Base Torque command of 9 Nm results in acceptable contact error, enabling training of the CMAC neural network modules.

During the 15 swings between t = 25 sec and t = 100 sec, the rule-based system components teach the CMAC network how to modulate reflexes in order to accomplish a successful swing. The first 5 swings arc performed in Rule-Based Swing Mode, with the CMAC network learning, but not contributing to, modulation of the shoulder and elbow joint reflexes. By t=50 sec, CMAC command suggestion errors have dropped low enough for the controller to enter Rule-and-Network-Based Swing Mode. At t = 80 sec, the controller momentarily switches to Network-Based Swing Mode, but the resulting swing falls short of the contact point. Finally, by t = 100 sec, four consecutive

Rule-Based Swing Mode

```
if
  swing mode was rule-based and
  required base torque has not been found and
  adequate contact was not made
then
  adjust base torque

if
  swing mode was rule-based and
  required base torque has not been found and
  adequate contact was made
then
  mark base torque as found and
  begin network training

if
  swing mode was rule-based and
  required base torque has been found and
  adequate contact was made but
  network-based command suggestions were not
  consistent with rule-based commands
then
  make no swing mode change

if
  swing mode was rule-based and
  required base torque has been found and
  adequate contact was made and
  network-based command suggestions were
  consistent with rule-based commands
then
  set swing mode to rule-and-network-based
```

Rule-and-Network-Based Swing Mode

```
if
  swing mode was rule-and-network-based and
  adequate contact was not made
then
  set swing mode to rule-based and
  mark required base torque as not found and
  stop network training

if
  swing mode was rule-and-network-based and
  adequate contact was made but
  network-based command suggestions were not
  consistent with rule-based commands
then
  make no swing mode change

if
  swing mode was rule-and-network-based and
  adequate contact was made and
  network-based command suggestions were
  consistent with rule-based commands
then
  set swing mode to network-based
```

Network-Based Swing Mode

```
if
  swing mode was network-based and
  adequate contact was not made
then
  set swing mode to rule-and-network-based

if
  swing mode was network-based and
  adequate contact was made
then
  make no swing mode change
```

Table 2. Sample of Execution Monitor Rules

network-based swings can be accomplished. Note in Fig. 4 that the number of rules tested during network-based swings drops considerably, reflecting a shift from a declarative to a reflexive form of processing.

In order to demonstrate the inherent adaptability of the integrated controller, the outer link mass is increased from 0.5 kg to 1.0 kg at t = 120 sec. As shown in Fig. 4, insufficient Base Torque causes the manipulator to miss the contact point over the next 6 swings. During this time the Swing Mode regresses from Network-Based, to Rule-and-Network-Based, to Rule-Based, and the Base Torque command is steadily increased. By t = 155 sec, a Base Torque of 19 Nm results in a successful rule-based swing. Between t = 160 sec and t = 205 sec, the CMAC network re-learns by rule-based example how to modulate properly the low-level joint reflex gains. By t = 205 sec, four successful network-based swings once again can be performed.

Utilizing the personal computer hardware and software described earlier, processing requirements for the various controller operations included the following. While in Rule-Based Swing Mode with no CMAC learning, a typical controller goal-directed search cycle testing 23 rules required 0.014 sec. The CMAC network (4 inputs, 6 outputs) required approximately 64 KBytes of memory (RAM). The reading (recall) of a single set of CMAC network outputs required 0.023 sec, and the updating (learning) of a set of CMAC input-output points required 0.047 sec. Although the present controller implementation configuration cannot support the simulated sample rate of 20 per sec, implementation of the integrated controller on multiple state-of-the-art microprocessor boards should achieve real-time performance.

CONCLUSIONS

Knowledge-based systems and artificial neural networks offer very different capabilities concerning control system design, implementation, and performance. The approach taken in this paperwas to develop a robotic control system that utilizes the strengths of each processing technique where appropriate. A rule-based system initially finds acceptable first-cut solutions to the given control objective, then teaches a neural network how to accomplish parts of the task which are amenable to the network's learning and generalization abilities. The network learns on-line by examples provided by rule-based task execution. The resulting integrated system enables a natural division oflabor; strategics for task execution (embodied by the execution monitor of the swing maneuver described above) remain within an expressive rule-based representation, whereas specific tactics (such as reflex gain modulation) are delegated to the most appropriate form of processing. In addition, by replacing rule groups with neural networks in a modular fashion, integration of the two computational paradigms occurs within the convenient hierarchical software architecture provided by knowledge-based programming techniques.

When integrated, knowledge-based systems and artificial neural networks merge declarative and reflexive forms of information processing. For the swing-maneuver controller described above, shifts between declarative and reflexive processing are traced by variations in the number of rules being tested and the errors associated with the network outputs during learning. Furthermore, a shift from declarative to reflexive processing implies changes in the controller's focus of attention and allocation ofcompu-

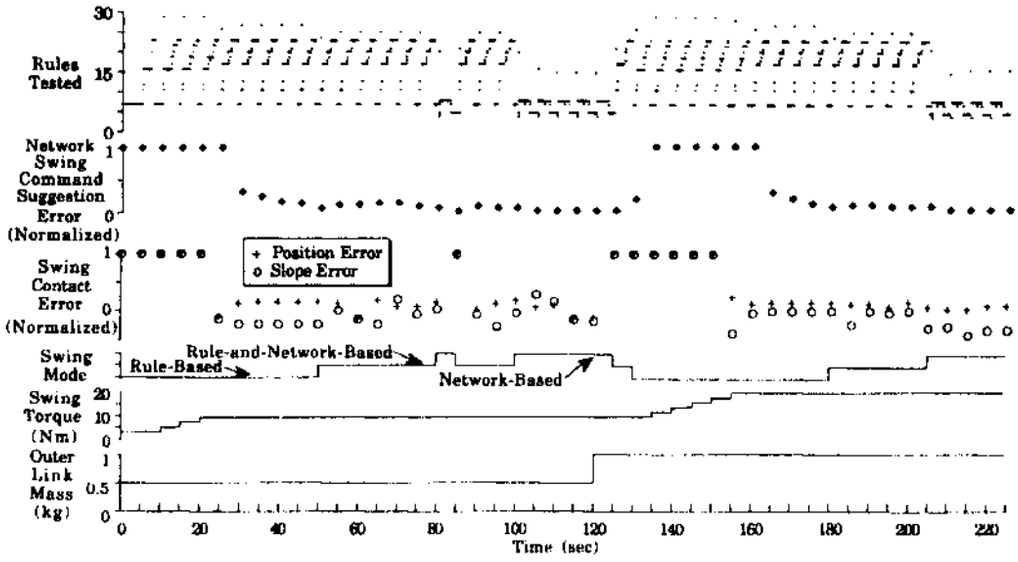

Figure 4. Rule-Based System Activity During Integrated Rule-Based/Network Based Swing Maneuver

lational resources. By integrating aspects of declarative knowledge-based systems and reflexive artificial neural networks in real time, the stage may be set for the development of highly adaptive and dexterous robotic control systems.

Many additional issues related to the integration of neural networks and knowledge based systems for autonomous control will be investigated in future research. These include: I)Ifwe were to use special purpose parallel architectures for the neural network and rule-based system control, would neural network control be significantly more efficient for the automatic control phase as in humans ? 2)How should optimal learning strategies used by the execution monitor be generated ? 3) In the preliminary example we investigated, we were able to transfer control smoothly between the knowledge-based system and neural network. What stability problems might arise in more complex systems ? 4) Given a group of neural networks assigned to perform a complex, multi-faceted task, how should the execution monitor knowledge base be set up to optimally allocate knowledge among networks and determine their training order ? 5)Can the transfer of knowledge among system components be bi-directional, that is, can the knowledge-based system infer new rules or modify existing ones based on neural network performance ?

Acknowledgements

The authors acknowledge many helpful discussions with Gregory Clark, Carl Olson and Paul Thagard of the Princeton University Psychology Department.

REFERENCES

[Albus,1975] J. Albus, "A New Approach to Manipulator Con trol: The Cerebellar Model Aniculation Controller (CMAC)",J. Dyn. Syst. Meas. Control, 97:270-277, 1975.

[Anderson,l 982] J. R.Anderson, "Acquisition of Cognitive Skill", Psychological Review, 89:369-406, 1982.

[Brooks,19861 V. Brooks, The Neural Basis of Motor Control, Oxford University Press, New York, 1986.

fCharniak and McDenriott,1985] E. Charniak and 1). McDerrnott, Introduction to Artificial Intelligence, Addison-Wesley Publishing Company, Reading, Mass., 1985.

[Fahlman and Hinton,1987] S. Fahlman and C. Hinton, "Connectionist Architectures for Artificial Intelligence", IEEE Computer, Jan. 1987.

[Handelman and Stengel,1987] D. Handelman and R. Stengel "An Architecture for Real-Time Rule-Based Contror, Proc. 1987 American Control Conference, Minneapolis, June 1987.

f Handelman and Stengel,19881 D. Handelman and R. Stengel, Perspectives on the Use of Rule-Based Control", Proc. IFAC Workshop on Artificial Intelligence in Real-Time Control, Swansea, UK, Sept. 1988, Pergamon Press, London, 1989.

[Handelman el. al.,1989] D. Handelman, S. Lane and J. Gelfand, "Integrating Neural Networks and Knowledge-Based Systems for Robotic Control", Proceedings of the 1989 IEEE Conference on Robotics and Automation, Scottsdale, Arizona, May, 1989, IEEE Computer Society Press, Los Angeles, 1989.

[Kupfermann,1985] I. Kupfermann, "Learning*', in E. Kandel, and J. Schwartz, Eds., Principles of Neural Science, Elsevier Science Publishing Co., Inc., New York,1985.

[Rummelhart and McClelland,1986] D. Rumelhart and J. McClelland, Eds., Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 1: Foundations, The MIT Press, Cambridge, Mass., 1986.

[Schneider, 1985] W. Schneider, "Toward a Model of Attention and the Development of Automatic Processing", in M. Posner and O. Marin, Eds. Attention and Performance XI, Erlbaum, Hillsdale, NJ.,1985.

[SchneiderandDetweiller,1987] W.Schneidcrand M.Detwciler, "A Connectionist/Control Architecture for Working Memory", in G. Bauer, Ed., Vol. 21, Academic Press, New York, 1987.

[Stein and Capady,1988] R. Stein and C. Capaday "The Modulation of Human Reflexes during Functional Motor Tasks", Trends in Neural Science, Vol. 11, No. 7,1988.