557864 P6

N90-20939

RULE-BASED MECHANISMS OF LEARNING FOR INTELLIGENT ADAPTIVE FLIGHT CONTROL

David A. Handelman* and Robert F. Stengel**

Princeton University
Department of Mechanical & Aerospace Engineering
Princeton, New Jersey

ABSTRACT

This paper investigates how certain aspects of human learning can be used to characterize learning in intelligent adaptive control systems. Reflexive and declarative memory and learning are described. It is shown that model-based systems-theoretic adaptive control methods exhibit attributes of reflexive learning, whereas the problem-solving capabilities of knowledge-based systems of artificial intelligence are naturally suited for implementing declarative learning. Issues related to learning in knowledge-based control systems are addressed, with particular attention given to rule-based systems. A mechanism for real-time rule-based knowledge acquisition is suggested, and utilization of this mechanism within the context of failure diagnosis for fault-tolerant flight control is demonstrated.

INTRODUCTION

Adaptability is an essential feature of any control system designed to interact effectively with the real world. Uncertainty motivates much of the need for adaptability, directly affecting control system stability and performance. Sources of uncertainty are many, representing inadequacies in knowledge of the system to be controlled, or in the environment within which the system must operate. For example, uncertainty can result from corruption of incoming information due to sensor noise or failure. It can also result from changes in control effectiveness due to failure or unanticipated changes in the operating environment, changes in system dynamics due to environmental factors or structural failure, and unanticipated or poorly-modeled external disturbances. Control law design must address to some degree these issues.

Fortunately, effective control techniques capable of accommodating certain types of uncertainty exist. Robust stochastic optimal estimation and control methods, for example, perform well in the presence of Gaussian sensor noise and state disturbances [1]. Characteristic changes in the dynamics of the controlled system often can be accommodated using parameter estimation and adaptive control techniques [2]. Under certain circumstances, self-organizing controllers may be used to perform non-trivial tasks given little prior information about the kinematics and dynamics of the system being controlled [3,4,5,6].

Even the most accommodating numerical control techniques, however, be they robust, adaptive, or self-organizing, eventually reach limits of performance when deficiencies in knowledge of the plant or its environment exceed certain thresholds. One limiting factor relates to the ability of the control system to learn about novel, important relationships and events in the world, and how to respond properly to them. Machine learning is an active area of research in the field of Artificial Intelligence (AI) [7,8,9,10,11]. However, although problem-solving techniques of AI are finding their way into various phases of control system design and implementation [12,13], little work has addressed the issue of learning in demanding real-time applications such as aircraft and spacecraft flight control.

This paper investigates how certain aspects of human learning can be used to characterize learning in "intelligent" control systems. Two types of memory and learning are described. It is shown that model-based adaptive control methods are particularly well suited for implementing one type, whereas the problem-solving capabilities exhibited by knowledge-based systems of artificial intelligence make them naturally suited for implementing the other type. Issues related to learning in knowledge-based control systems are addressed, with particular attention given to rule-based systems. A mechanism for rule-based knowledge acquisition is suggested, and utilization of this mechanism within the context of failure diagnosis for fault-tolerant flight control is described.

REFLEXIVE AND DECLARATIVE MEMORY AND LEARNING

Learning relates to knowledge acquisition, memory to its storage. Various methods of classification are used by psychologists to describe different types of memory and learning exhibited by humans. One classification scheme is based on how learned information is encoded and recalled, distinguishing between what some authors term <u>reflexive</u> and <u>declarative</u> memory and learning [14]. With regard to control, maneuvers indicative of reflexive mechanisms may be characterized as automatic, requiring little or no thought. Maneuvers involving declarative memory and learning, on the other hand, require conscious effort. Evaluation, comparison, and inference characterize declarative thinking. Moreover, whereas reflexive learning relates specific responses to specific stimuli, declarative learning provides insight into not only how something is done, but why. Any complex task attempted for the first time involves some form of declarative reasoning.

* Graduate Student
** Professor
Presented at the 1988 American Control Conference, Atlanta, June 1988.

Reflexive and declarative memory and learning are closely related. Tasks initially learned declaratively often become reflexive through repetition. Conversely, when familiar tasks are attempted in novel situations, reflexive knowledge must be converted back into declarative form in order to become useful. 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. By drawing analogies to such human information processing mechanisms, adaptive control systems might benefit from the incorporation and integration of both reflexive and declarative forms of learning [15].

A simplistic example of learning pertinent to aircraft flight control demonstrates that distinctions between reflexive and declarative learning can affect aircraft stability and performance. Consider what happens when a student pilot is taught how to recognize and recover from a wings-level approach-to-landing stall. The student is shown how sluggish control response and aircraft buffeting indicate low airspeed and impending stall, and that recovery includes pushing the control stick forward. Reflexive learning would encode knowledge similar to the following.

```
If      control response is sluggish
        and
        buffeting is encountered
then    push stick forward
```

Conversely, declarative learning would result in the acquisition of knowledge encoding more causal detail, such as the following.

```
If      control response is sluggish
        and
        buffeting is encountered
then    decrease magnitude of angle of attack

If      magnitude decrease in angle of attack
        is required
then    push stick forward
```

The type of knowledge acquired by the student, reflexive or declarative, has a large impact on the student's ability to apply this knowledge in novel situations (and hence to adapt). For example, during aerobatic flight, the student notices significant differences between required control inputs for inverted and non-inverted flight.

```
If      aircraft is upright
        and
        magnitude decrease in angle of attack
        is required
then    push stick forward

If      aircraft is inverted
        and
        magnitude decrease in angle of attack
        is required
then    pull stick aft
```

How will the student respond when pre-stall conditions (control sluggishness and airframe buffeting) are encountered during inverted flight? A reflexive response would be based on the relationship between pre-stall warnings and forward stick movement learned during non-inverted flight. Such a response would aggravate the stall. Declarative thinking, on the other hand, would recognize the need for a reduction in the magnitude of angle of attack, and that when inverted such a reduction is accomplished with aft stick movement.

The distinction between reflexive and declarative memory and learning suggests what roles existing systems-theoretic adaptive control techniques and proposed artificial intelligence methodologies can play in adaptive control systems. Most existing adaptive control techniques are based (for good reason) on mathematical models of the dynamic system being controlled. The analytical functions representing the adaptive control law ultimately calculate specific control commands in response to specific sensor measurements. In this sense, these model-based techniques can be viewed as implementing reflexive knowledge. Conversely, the inferencing capability of knowledge-based systems can be viewed as implementing declarative knowledge. Many model-based adaptive control techniques benefit from the availability of closed-form analytical expressions dictating how model parameters should be modified. Learning often proceeds in a stable and optimal fashion, and these algorithms should be used when possible. Unfortunately, with knowledge-based systems, no such rigorous guidelines for learning exist.

In general, a learning controller should be able to identify important new information, decide whether this new information should augment or replace existing knowledge, and transfer this information into the existing knowledge base. Possible schemes for knowledge acquisition include rote learning, learning by example, and learning by trial and error. As mentioned above, much AI research addresses learning in knowledge-based systems, and the field of adaptive control most likely will benefit from its advances. Any knowledge-based application, however, is based upon a specific form of knowledge representation. The following sections suggest that rule-based expert system techniques provide a sound representational basis for declarative learning in time-critical adaptive control systems.

REAL-TIME DECLARATIVE LEARNING THROUGH RULE RECRUITMENT

The knowledge representation and problem-solving features of rule-based systems make them particularly well-suited for implementing the causal relationships characteristic of declarative learning. Moreover, as will be demonstrated, knowledge acquisition can be made to occur in the computationally efficient manner required for real-time control. Note that the discussion below focuses on mechanisms enabling automated rule-based knowledge acquisition, not on how this new knowledge is identified.

Within a certain class of forward- and backward-chaining rule-based systems [16], the knowledge base is composed of parameters and rules. Parameters represent symbolic information. Each parameter may acquire one of a list of allowed values, or its value may be considered unknown. Information expressing relationships and dependencies between parameter values is contained in rules. Each rule contains a premise and an action. If a rule premise is true when tested, its action is executed, causing the inference of additional information. In control system applications, rule actions also may perform specific control tasks. Rule testing is guided by an inference engine, an operator applied to the knowledge base enabling the process of search. Parameters thereby represent a partial description of the "state of the world", and rule-based search is used to modify this description. Figure 1 defines a symbology useful in graphically depicting such a knowledge base. Rectangles contain all values that the corresponding parameter can acquire. With arcs between parameters representing rules, the resultant AND/OR graph can be used to trace the logic path taken by the search process.

Various forms of search may be applied to the knowledge base. The key to search is the manner in which rules are linked through parameters. Lists of rule names associated with each parameter provide this link. For example, the purpose of a goal-directed (backward-chaining) search is to infer a value for a specified parameter. To this end, each parameter has attached to it a list identifying which rules are capable of modifying the value of the parameter through the rule action. This list is consulted by the inference engine during goal-directed search when a parameter value must be inferred.

In general, knowledge acquisition within such a rule-based system involves three steps. These three steps are depicted graphically in Fig. 2 to 4 with reference to the knowledge associated with the stall recovery scenario given above. First, parameters capable of representing the "state space" of knowledge to be learned are collected as shown in Fig. 2. In the second knowledge acquisition step, rules associating parameter states are constructed as in Fig. 3. Finally, rules are linked by updating the rule-name lists associated with relevant parameters. The resultant knowledge base is depicted in Fig. 4.

Following incorporation into the knowledge base, these new rules may be utilized for control system problem solving. A goal-directed search on the knowledge of Fig. 4, for example, would begin with the question, "How should the stick be moved?" or more specifically the instruction, "Determine the value of parameter DESIRED_STICK_MOVEMENT." Rules 1 and 2 are capable of supplying this information. Rule 1 is tested first, with its premise initially checking aircraft attitude. Assume the aircraft is inverted. In this case, Rule 1 fails, and Rule 2 is tested. The premise of Rule 2 eventually needs to know whether or not a magnitude decrease in angle of attack is required. Rule 3 is capable of supplying this information; therefore it is tested at this time. If control response is sluggish and buffeting is encountered, the action of Rule 3 determines that a change in angle of attack is required, finally allowing the action of Rule 2 to determine that aft stick movement is appropriate.

Rules represent executable code. Knowledge acquisition as specified above involves the automatic generation and execution of code during control system operation. List-based programming languages such as LISP can be used to accomplish such feats. Problems can arise, however, when symbolic programming languages and hardware must be integrated with

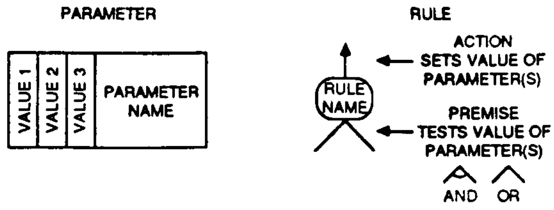

Fig. 1. Graphical Representation of Rule-Based Knowledge.

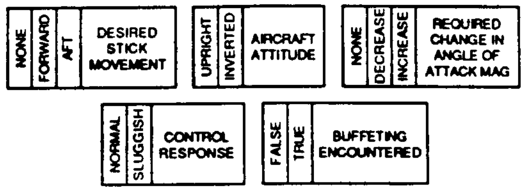

Fig. 2. Parameter Representation of Acquired Sample Knowledge.

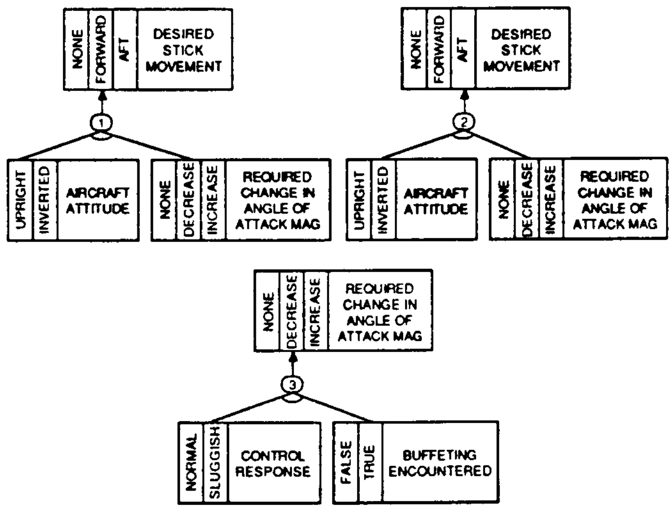

Fig. 3. Rule Representation of Acquired Sample Knowledge.

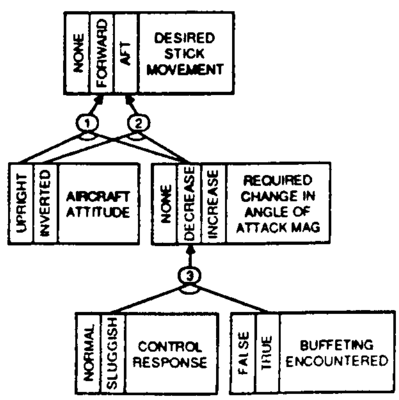

Fig. 4. Knowledge Base Representation of Acquired Sample Knowledge.

numeric control system processing chores [17]. Fortunately, by sacrificing some flexibility, real-time performance can be obtained. For example, by encoding rules in a structured procedural language such as C or Pascal, and integrating them with numerical routines in a multiprocessor system, symbolic processing capability remains limited, yet powerful [18,19]. Furthermore, knowledge acquisition can be enabled in such a system through a process called <u>rule recruitment</u>.

Rule recruitment involves the manipulation of a stack of <u>dormant rules</u>. A rule is defined as dormant if it cannot be referenced during search. This situation will occur if the rule's name does not appear on any parameter's rule-name list. For example, before the rules depicted in Fig. 3 are linked into the knowledge base through their respective parameters, they remain dormant and inaccessible to the inference engine. Simply by manipulating parameter lists, rules may be transferred into and out of dormancy.

Rules on the dormant rule stack serve as rule "templates". Each retains a fixed premise and action structure but refers to parameters and parameter values indirectly through pointer-type references. For example, the following rule template exhibits a structure capable of encoding the rules of Fig. 3.

```
If     <parameter pointer> is <value pointer>
       and
       <parameter pointer> is <value pointer>
then   set <parameter pointer>
       to  <value pointer>
```

By manipulating rule template pointers, rules may be built automatically as required. The process of rule recruitment, therefore, involves pulling a rule template off the dormant rule stack, initializing its pointers so that it encodes the desired chunk of knowledge, and modifying the appropriate parameter lists so that the new rule becomes an active part of the knowledge base. Figure 5 depicts the process of rule recruitment.

The major drawback with this knowledge acquisition mechanism is the inability to build arbitrarily complex rules at run-time. All desired parameters must be pre-defined as well. However, by forcing a system designer to formalize the structure of knowledge to be learned by the controller, these limitations may prove beneficial in the long run. A restricted set of unique rule templates encourages modular construction of more complex rules. Furthermore, a large set of unique rule templates may be designed into the dormant rule stack if needed. This remedy is memory intensive, not computation intensive, and memory is inexpensive.

The major advantage associated with rule recruitment is that it is fast. The execution time incurred during pointer assignment and parameter list updating is negligible. Moreover, the recruitment of rules can be overseen by other rules dedicated to knowledge acquisition, in much the same way that meta-rules can be used to guide rule selection during search [16]. Consequently, the knowledge acquisition mechanism of rule recruitment fits neatly into the existing computationally efficient rule-based control system environment.

APPLICATION OF RULE RECRUITMENT TO FLIGHT CONTROL SYSTEM DESIGN

A demanding adaptive control application was chosen as a testbed for some of the rule-based learning ideas presented above. The Rule-Based Flight Control System (RBFCS) is designed to combine analytical redundancy and expert system concepts for fault-tolerant flight control [19,20]. The software and hardware architectures of the RBFCS provide for real-time integrated symbolic and numeric processing. Within this setting, rule-based learning has been used in conjunction with model-based simulation to facilitate certain phases of control system design.

The RBFCS is intended to detect, identify, and reconfigure for a wide range of aircraft failures. The overall job of failure accommodation is broken down into five main tasks. The Executive Control Task provides continual dynamic state estimation, feedback control calculations, and synchronization of the remaining tasks. The Failure Detection Task monitors aircraft behavior and detects significant abnormalities. The Failure Diagnosis Task finds a set of probable causes of the problem, and the Failure Model Estimation Task generates a mathematical model of the aircraft dynamics considered to reflect changes arising from the assumed failure. Finally, the Reconfiguration Task determines what action should be taken to correct the situation. Automated learning has been used to generate rules relevant to failure diagnosis.

At the core of the Failure Model Estimation Task is a numerical algorithm that chooses from among a group of failure hypotheses the one most likely (in a probabilistic sense) to represent the actual failure. The number of hypotheses considered by the algorithm must be kept low, and this is the job of the Failure Diagnosis Task, which performs initial failure candidate screening. The intent is to use expert system techniques to emulate in real-time the reasoning of pilots, engineers, and mechanics familiar with the aircraft in order to make informed judgments as to what did or did not fail [21].

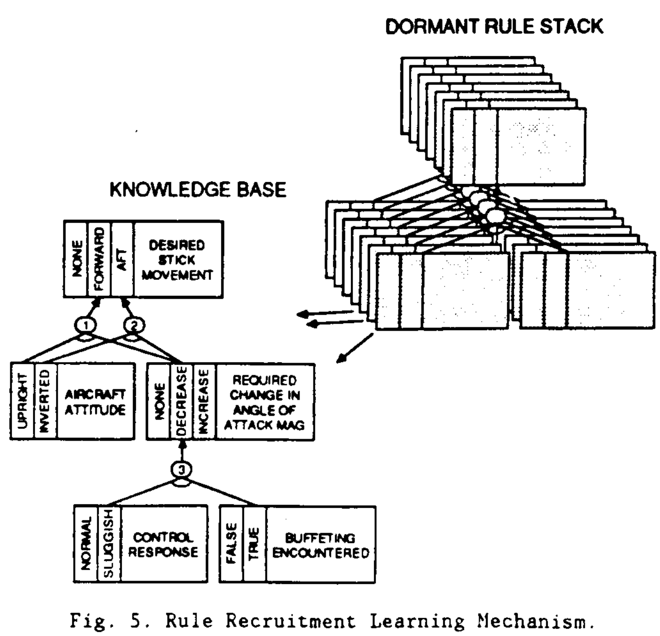

Fig. 5. Rule Recruitment Learning Mechanism.

In addition to containing explicitly specified heuristic knowledge, the Failure Diagnosis Task also has been given the capability of learning through simulation how the Failure Detection Task will respond to various failures. The impetus behind this capability is the intent to accommodate eventually an extremely large number of possible failure modes. The RBFCS presently is configured to accommodate a biased or stuck sensor or control in a U. S. Army CH-47 tandem-rotor helicopter travelling at 80 knots airspeed and sea level altitude. However, accommodation of structural failures affecting aircraft dynamics, as well as multiple and intermittent failures in sensors and controls, is included in the design goal of the RBFCS. By using analytical models of these failures, stochastic Monte Carlo type simulations can be used to characterize the effect that such complex failures have on failure detection. Learning by example off-line, the Failure Diagnosis Task generates a set of rules that approximates the effect of each failure mode. Utilizing these rules on-line, the task bases its initial screening of failure candidates, in part, on similarities between available failure-time information and effects known to be caused by specific types of failures.

This simulation-based learning by example is accomplished using rule recruitment. Presently, seven rules are used to approximate the average effect each failure mode has on 16 indicators. Windowed average and root-mean-square values of the residuals of an 8-state estimator are used as indicators. The 24 possible failure modes correspond to abrupt bias and stuck failures in 8 sensors and 4 controls. Sensors measure body-axis longitudinal velocity, lateral velocity, vertical velocity, roll rate, pitch rate, yaw rate, pitch angle, and roll angle. Controls include the two actuators of each rotor: forward cyclic pitch, forward collective pitch, aft cyclic pitch, and aft collective pitch. Each rule has the following form.

```
If     indicator_01 is near x.x
       and
       indicator_02 is near x.x
       and
                  :
       and
       indicator_16 is near x.x
then   there is good chance that
       forward collective pitch control is
       biased from nominal by an amount near x.x
       and
       failure detection delay is near x.x
```

The definition of "near x.x" in these rules is defined by fuzzy functions [22]. Failure detection delay is the time difference between detection and occurrence of the failure. Heuristics are used to distribute the 7 rules per failure mode throughout the expected failure mode range.

When recruited into the knowledge base, the 168 failure-effect rules integrate features of function approximation and pattern recognition with the remaining heuristics of the Failure Diagnosis Task. They help estimate at failure detection time the relative likelihood of each failure mode, failure mode magnitude and direction, and failure detection delay. The relative likelihood of a failure mode depends on the validity of its rule's premises, and it is used to narrow the initial set of failure mode candidates down to a reasonable size.

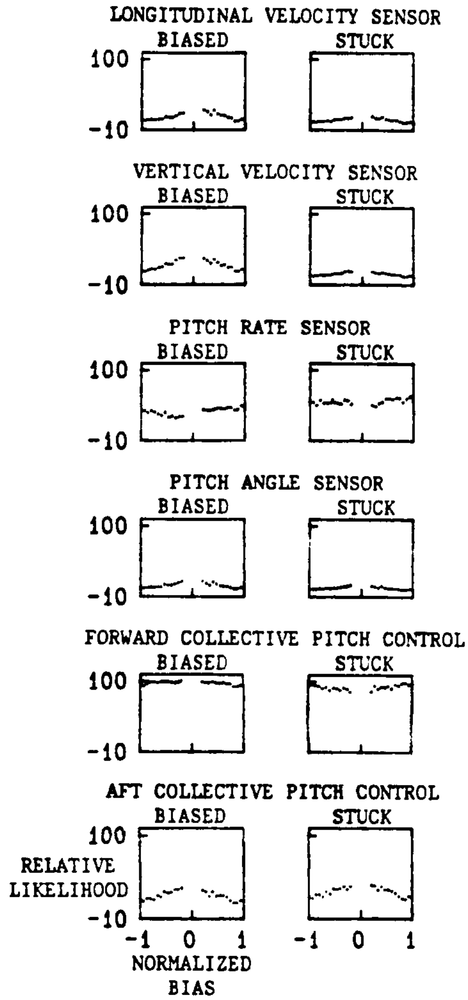

Fig. 6. Rule-Based Failure Diagnosis Performance: Failure Mode Likelihoods Given Biased Forward Collective Pitch Control.

Figure 6 illustrates some of the on-line information generated by these off-line recruited rules. For this figure, bias failures in forward collective pitch control were simulated using a stochastic fully-coupled 8th-order linear model of aircraft dynamics. Failure Diagnosis Task processing typically required less than 3 seconds on a computer equipped with an 8-MHz 80286 CPU and an 8-MHz 80287 math coprocessor. The abscissa of each plot in the figure corresponds to the amount of failure bias, normalized for a range of +/- 7.5 cm. Each data point in a plot corresponds to the average relative likelihood of the associated failure mode candidate over 20 simulated failure runs. Failure detection delays for these bias failures varied between 0.2 sec and 2 sec. Failures remaining undetected 2 sec after their occurrence were considered undetectable, explaining the gap in data associated with near-zero bias failures. It can be seen from the plots in Fig. 6 that likelihoods associated with the actual failure mode described as "forward collective pitch control biased" usually were the highest. Furthermore, other failure mode candidates capable of strongly affecting longitudinal dynamic state variables had significant likelihoods, as expected. Likelihoods associated with failure modes strongly affecting lateral/directional states (not shown) remained low.

By applying other heuristics in concert with this type of information, the Failure Diagnosis Task can quickly narrow the number of failure mode candidates from 24 down to 6 or less. Note that due to a rule-based implementation, the failure-effect knowledge generating this information can be used on a conditional basis if desired. For example, rule premises can be made sensitive to previously identified failures. The rule-based technique thereby exhibits in this case certain advantages over standard pattern matching techniques. Additionally, although the recruited failure diagnosis rules of the RBFCS were obtained off-line as part of control system design, the same mechanism of learning could be utilized during on-line control system operation.

CONCLUSIONS

The problem-solving capabilities of numeric model-based systems and symbolic knowledge-based systems can be used to implement various forms of automatic learning. The concept of learning through rule recruitment described above serves as an extension to work originally designed to integrate such symbolic and numeric processing for real-time control. Rule recruitment provides a mechanism whereby knowledge may be acquired automatically in a timely manner, allowing rules to generate additional rules. It can be used as a representational vehicle through which more fundamental issues of control system learning may be addressed, such as the acquisition and maintenance of general knowledge for highly adaptive aircraft and spacecraft flight control.

ACKNOWLEDGMENT

This project was sponsored by the U. S. Army Research Office under Contract No. DAAG29-84-K-0048.

REFERENCES

1. Stengel, R., Stochastic Optimal Control: Theory and Application, John Wiley & Sons, New York, 1986.

2. Goodwin, G., and Sin, K., Adaptive Filtering Prediction and Control, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1984.

3. Saridis, G., Self-Organizing Control of Stochastic Systems, Marcel Dekker, Inc., New York, 1977.

4. El-Fattah, Y., and Foulard, C., Learning Systems: Decision, Simulation, and Control, Springer-Verlag, New York, 1978.

5. Narendra, K., ed., Adaptive and Learning Systems: Theory and Applications, Plenum Press, New York, 1986.

6. Miller, W. III, "Sensor-Based Control of Robotic Manipulators Using a General Learning Algorithm," IEEE J. Robotics and Automation, Vol. RA-3, No. 2, April 1987.

7. Barr, A., Cohen, P., and Feigenbaum, E., The Handbook of Artificial Intelligence, William Kaufmann, Los Altos, California, 1982.

8. Michalski, R., Carbonell, J., and Mitchell, T., Machine Learning: An Artificial Intelligence Approach, Tioga Publishing Company, Palo Alto, 1983.

9. Michalski, R., Carbonell, J., and Mitchell, T., Machine Learning: An Artificial Intelligence Approach Volume 2, Morgan Kaufmann Publishers Inc., Los Altos, California, 1986.

10. Michalski, R., Carbonell, J., and Mitchell, T., Machine Learning: A Guide to Current Research, Kluwer Academic Publishers Inc., Boston, 1986.

11. Widrow, B., Winter, R., and Baxter, R., "Learning Phenomena in Layered Neural Networks," Proc. First Internat. Conf. on Neural Networks, San Diego, June 1987.

12. Hankins, W., Pennington, J., and Barker, L., "Decision-Making and Problem-Solving Methods in Automation Technology," NASA-TM-83216, May 1983.

13. James, J., "A Survey of Knowledge-Based Systems for Computer-Aided Control System Design," Proc. 1987 American Control Conference, Minneapolis, June 1987.

14. Kandel, E., and Schwartz, J., Principles of Neural Science, Elsevier Science Publishing Co., Inc., New York, 1985.

15. Hutchison, W., and Stephens, K., "Integration of Distributed and Symbolic Knowledge Representations," Proc. First Internat. Conf. on Neural Networks, San Diego, June 1987.

16. Buchanan, B., and Shortliffe, E., Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project, Addison-Wesley Publishing Company, Reading, Mass., 1984.

17. Birdwell, J., Cockett, J., and Gabriel, J., "Domains of Artificial Intelligence Relevant to Systems," Proc. 1986 American Control Conference, Seattle, June 1986.

18. Heudin, J., Burg, B., and Zavidovique, B., "A Flexible Operating System for Real-Time Process Control," Applications of Artificial Intelligence IV, Proc. SPIE, Vol. 657, April 1986.

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

20. Handelman, D., and Stengel, R., "Combining Expert System and Analytical Redundancy Concepts for Fault-Tolerant Flight Control," AIAA J. Guidance, Control, and Dynamics, Vol. 11 (to appear), 1988.

21. Gilmore, J., and Gingher, K., "A Survey of Diagnostic Expert Systems," Applications of Artificial Intelligence V, Proc. SPIE, Vol. 786, May 1987.

22. Tong, R., "A Control Engineering Review of Fuzzy Systems," Automatica, Vol. 13, No. 6, Nov. 1977.