A Survey of Some Implementations of Knowledge-Based Systems for Real-Time Control

John R. James  
Department of Electrical Engineering  
United States Military Academy  
West Point, NY 10996-1787

Gregory J. Suski  
Lawrence Livermore National Laboratory  
P.O. Box 5508, Mail Stop L 550  
Livermore, CA 94550

ABSTRACT

The application of knowledge-based programming techniques to real-time control is growing rapidly. One area of application is for systems in which a mathematical model cannot be developed or is too costly to develop. In this case, fuzzy logic controllers and rule-based controllers have been implemented which apply knowledge-based techniques to control process variables. In particular, the chemical control industry, which has many slow, poorly modeled processes, is moving aggressively to apply knowledge-based control techniques. In one successful application, Air Force researchers have developed a qualitative process controller which is improving the quality and decreasing the curing time for composite material production. Another area of application is for systems with an adequate model but large variations in plant parameters. In this case, symbolic processing is being coupled with numerical processing to build knowledge-based adaptive controllers. A knowledge-based adaptive controller is being built for the power and thermal distribution control systems of the Space Station. Several knowledge-based PID controllers, and a knowledge-based adaptive lead-lag controller have been implemented as well. Implementations have been made on microcomputers as well as minicomputers. Multiprocessor configurations have been built for autonomous robots, reconfigurable controllers and an autonomous underwater vehicle. Also, general purpose programming tools have been made available to aid in the implementation of knowledge-based, rule-based controllers, to include one commercial process specifically designed for that purpose. This paper provides an overview of efforts to build knowledge-based systems for real-time control. We begin with a discussion of what knowledge-based systems have to offer to the control systems engineering process. This is followed by examples of applying knowledge-based techniques to system identification and adaptive control. Projects which use qualitative reasoning and rule-based controllers are then outlined. Finally, research issues are discussed.

I. WHY USE KNOWLEDGE-BASED SYSTEMS?

There is presently a surge of interest in building knowledge-based controllers (Åström [1], Bonissone [3], Brand [6], Eldeib [8], Glass [9], Gupta [10], Handelman [12], Hong [13], James [17], Kohn [18], Krijgsman [19], Lane [20], LeClair [23], March-Leuba [27], Russell [29]). A natural question which arises is, “What does this technology have to offer that justifies the resource expenditures required to implement it?” In fact, the extra effort required to build a knowledge-based system is not justified if reliable, inexpensive solutions exist for well-understood problems. Also, as Åström and Wittenmark point out: “Many cases are known where a constant gain feedback can cope well with considerable variations in system dynamics” (Åström [2] page 357). However, the increasing availability of ever more computing power and mass memory storage continue to make what might at one time have been “a sledgehammer solution to a peenball hammer problem” to be now a cost effective way of improving system performance. Serious consideration is being given to increasing system reliability by predicting component failure and recomputing control laws in response to component failure or large plant parameter variations. The advent of the microprocessor has been one of the sources of inspiration for techniques developed for building self-tuning regulators and model-reference adaptive controllers. These approaches are largely heuristic and, since Artificial Intelligence (AI) research has focused on implementing heuristic techniques, knowledge-based controllers can be considered the next logical step. Also, knowledge-based technology offers an approach to deal with the increase in system complexity encountered as efforts are made to stretch the limits of system performance and integrate more capabilities (James [15]). In addition, the inherent ability of knowledge-based systems to support incremental expansion of capabilities and provide justification for recommendations or actions is difficult to achieve with conventional programming techniques. Finally, as Åström and Wittenmark have pointed out, the majority of the code required to build practical controllers is not associated with the analytically-determined controller algorithm but with the logic required for operator communication and associated functions (Åström [2] page 388). Thus, in a practical sense, the application of knowledge-based techniques to control systems engineering can be considered as simply a cost-effective and reliable way of formally representing the ancillary logic required to implement a controller. General areas of application which are being investigated include system initialization logic and exception handling (Åström [1]), control of incompletely understood processes (LeClair [23], Brand [6]) system identification (Glass [9], Takagi [34], knowledge-based adaptive control and rule-based control (Burks [7], Mandani [26], Takagi [34], Handelman [12], Taylor [35,36], Szlapczynski [33]). Thus, knowledge-based control systems can be used to:

1. Implement heuristic control schemes,
2. Diagnose or predict component or sensor failure,
3. Identify changes in plant parameters or structure,
4. Recalculate control laws based upon a knowledge of the current plant parameters,
5. Select appropriate control laws based on the current plant situation,
6. Execute ancillary control logic which has historically been used for practical controllers, and
7. Explain the current situation to the user.

II. IDENTIFICATION AND ADAPTATION

A long-standing goal of control systems engineering has been the automatic identification of plant variations and corresponding adaptation of the control law (Mendel [28]). Fortunately, the single most successful area in the development of expert systems has been that of system diagnosis.

Figure 1. Knowledge-Based Adaptive Controller Structure

The combination of expert systems programming techniques with algorithmic system identification or fault detection methods have been investigated by several groups (Glass [9], Handelman [12], Taylor [35]). Theoretical structures for considering such general machines have been proposed by at least two authors (Gupta [10], Kohn [18]). A typical configuration for such a system is shown in Figure 1.

The basic idea (Sanoff [30], Åström [1], James [17]) is that the reconfigurable controller is continuously working to implement the current control law while the logical environment for analysis and design (LEAD) mechanism is functioning in a supervisory role to determine if the controller should be changed. An early commercial venture into knowledge-based control, the Process Intelligent Control (PICON) system, was released in 1984 and could monitor system performance and diagnose faults in order to provide advice on system status but could not directly change the controller actions. A review of the capabilities of PICON, together with a proposal of its application to control of Space Station processes was given by Leinweber [24]. The G2 system recently announced by Genesym corporation is a commercial product for implementing knowledge-based adaptive controllers which provides capabilities similar to PICON but can also directly change controller actions.

The second basic way in which knowledge-based systems have been applied to accommodate uncertain or widely varying plants is through direct application of the rules to control processes. The configuration of such a system is similar to that of Figure 1 except that the reconfigurable controller is replaced by a rule-based controller. The difference between the two approaches is that in the case of direct knowledge-based control (DKBC) the rules themselves determine the control action instead of determining changes to a conventional control law implementation. A LEAD mechanism for supervisory operations and communication with the user may or may not be present. Comments on several projects follow:

1. Sanoff and Wellstead developed the Configuration and Run-Time Expert (CORTEX) at the University of Manchester (Sanoff [30]) for real-time control applications. The system is written in PASCAL and features a run-time expert system which operates in real time and a second expert system which has both foreground functions which operate in real time and background functions which make high-level decisions and produce reports.

Figure 2. Example of a Knowledge-Based Adaptive Controller

2. The Lund Institute of Technology in Sweden has been actively involved in building knowledge-based systems for system identification and adaptation. An early version of a proportional, integral, derivative (PID) autotuning controller (Åström [1]) was implemented on a VAX minicomputer. The VAX environment was also used to build a knowledge-based interface into the IDPAC system identification program (Larsson [21]). Lund continues to lead in knowledge-based, real-time control systems research by being the first academic institution to acquire the G2 system from Genesym corporation.
3. The Systems Autonomy Demonstration Office at NASA, Ames research facility at Moffett Field, California is building a knowledge-based system for model identification and adaptive control of the Space Station prototype thermal testbed. The approach uses a combination of rule-based, object-oriented and data-driven techniques for modeling, identification and adaptive control (Glass [9]).

4. Most of the knowledge-based, real-time control projects discussed in the open literature have been applied to computer-generated simulations of physical systems and have used minicomputers and workstations for the implementations. However, one project has built a knowledge-based adaptive controller for a physical system which uses inexpensive, reliable and readily available equipment (Janes [17] see Figure 2). The inference engine is public domain and is written in TURBO PASCAL. The system identification and control design programs are also written in TURBO PASCAL and the resulting logical environment for analysis and design (LEAD mechanism) runs on an IBM-PC/AT compatible microcomputer. The recongurable controller is implemented on a commercially-available plug-in board which has an analog-to-digital converter, a Texas Instruments TMS32010 digital signal processor, and a digital-to-analog converter (see Figure 3). The system is configured to identify changes in the ES151 educational servo system, recalculate the control law, and implement the new control law.

![Figure 3. Block Diagram of the ASP 320/FC-10](image)

5. General Electric Corporation has been investigating the application of knowledge-based, real-time control (Taylor [35, 36]). This project uses DELPHI, a proprietary forward and backward-chaining inference engine, to implement a failure detection and isolation methodology for a simulation of the GE-21 turbine engine. The inference engine is written in VAXLISP and features coupling of the inference mechanism with FORTRAN analysis and design programs.

6. Another General Electric effort is associated with the Defense Advanced Research Projects Agency (DARPA) Pilot's Associate project. A group of software tools, the Uncertainty Tool Set, has been developed to aid in the transition of the knowledge-based software from development to deployment (Bonissone [4]). The tool set includes a program which applies the theory of plausible reasoning, developed at General Electric, to build rule bases and a separate program to reason opportunistically with compiled versions of these rules in order to achieve real-time performance.

7. Several projects at Delft University of Technology have investigated the application of knowledge-based systems to in-line control applications. (Krijgsman [19]). One project involves the construction of a PID autotuning mechanism in which an expert system, written in FORTH and running on a PDP 11/60, is used to adjust the structure (P, PI, or maximum control output) as well as the values of the control parameters. The system has been used to control a simulation of the temperature control of a large building. A separate project has used ENVISAGE (a large expert system shell), running on one µ-VAX-II computer, to supervise an adaptive control algorithm, running on a separate µ-VAX-II which is implemented using MUSIC, a real-time simulation and control package, and

8. An effort at the University of Maryland (Lebow [22]) has focused on the development of a practical implementation of an expert industrial controller using microprocessor-based systems. An adaptive PID controller has been constructed around a Texas Instruments programmable logic controller as the initial step in the project.

**III. QUALITATIVE REASONING**

Qualitative process control involves the construction of a qualitative model of the physical process and subsequent reasoning about the qualitative model to determine control actions. A comparison of various qualitative representation schemes is given in Bonissone [5]. Efforts in this area include:

1. The development of qualitative process automation at the Air Force Wright Aeronautical Laboratories, Wright-Patterson Air Force Base, Ohio. This project has resulted in applying qualitative process control to a composite material curing process. The technique has been applied to a variety of materials and geometries and has resulted in a higher quality product in less curing time. The system is implemented using multiple microprocessors (one 80286 and two MC68000s) and a blackboard architecture (LeClair [23]). In this case qualitative reasoning remains in a direct knowledge-based controller.

2. The previously-mentioned NASA project (Glass [9]), uses a qualitative model of the Space Station prototype thermal testbed in order to detect qualitative changes in the plant and provide input into the portion of the system which redesigns the control law. In this case qualitative reasoning is used in a knowledge-based adaptive controller, and

3. Professor Hong’s effort at Grumman Corporation has the goal of providing dynamic control adaptation to respond in real time to unanticipated situations (Hong [13]).
Qualitative reasoning is being investigated to minimize processing time.

IV. RULE-BASED CONTROLLERS

Rule-based, real-time controllers have been built using fuzzy logic as well as forward and backward-chaining inference mechanisms. Dr. H. Watanabe has designed a VLSI chip which could perform approximately 50,000 logical inferences per second using fuzzy logic. He is currently involved in improving the design and building a fuzzy logic controller.

Projects in this area include:

1. The seminal effort by Dr. Mandani to build a fuzzy logic controller (Mamdani [26]),

2. The effort by Takagi and Sugeno to implement an approach for fuzzy identification and subsequently apply the technique to derive fuzzy controllers for a water cleaning process and a steel-making process (Takagi [34]),

3. The HERMES series of autonomous robots built by the Oak Ridge National Laboratory's Center for Engineering Systems Advanced Research (CESAR). A concise review of the HERMES series of experiments is provided by Burks et al [7]. The HERMES-IIB robot provides a testbed for research in perception, planning, dealing with unexpected occurrences and goal recognition. The project uses an industrial version of the IBM-PC/AT (the IBM 7532) as the host machine for a hypercube of two NCUBE 32-bit machines which provide on-board computation and also communicates with a VM/38 subsystem over an 8-megabaud parallel link for off-board computation. The HERMES-IIB robot has a rule-base written in a LISP type format which provide for high-level decisions and diagnoses unexpected occurrences. The article by Burks et al references several other small mobile robot projects which have similar research goals.

4. The Autonomous Land Vehicle program sponsored by DARPA has funded the construction of several knowledge-based, real-time systems. The FMC mobile robot is one of the large mobile robots built under this program (McTamaney [25]). The objective has been to develop an autonomous vehicle capable of real-time operation both on- and off-road. The FMC system has demonstrated mission planning, route planning, obstacle detection and avoidance, and road following. The system can be operated in a teleoperated mode and the majority of the computation is done in a separate command center. The mission planning subsystem uses a LISP-like plan generation language to both perform global path planning and execute the planned path. The various modules use a blackboard architecture to access a common symbolic data base.

5. The BOFFIN program has resulted in the construction of an unmanned autonomous submersible (Russell [29], Lane [20]). The autonomous robot is deployed from a tethered vehicle and is intended to perform simple missions such as diver support. Interminent communication with the tethered vehicle is over two acoustic communication channels. The system uses a blackboard symbolic computation architecture implemented in the C programming language. The system is built with multiple microprocessors and uses the UNIX operating system.

6. A series of experiments in reconfigurable control have been funded by the US Army Research Office at Princeton University (Huang [14], Handelman [12]). The project has resulted in the construction of a multiple-microprocessor implementation of a blackboard architecture for recognition of plant faults and recalculation of control laws. This application shares the use of the blackboard approach to segment the symbolic computation effort and achieve real-time performance with the mobile robot projects and the BOFFIN effort mentioned above.

7. A FORTH-based expert system built by the Delft research group which optimizes the controller settings of a conventional three-term controller, based on the measured values of the system error and its first difference, and

8. An ongoing project at George Mason University to build a rule-based controller for a servomechanism (Eldeib [8]). This project has involved using a single microprocessor for the inference mechanism as well as the control mechanism and reports on the slowness of the resulting system. Other approaches are being investigated.

V. RESEARCH ISSUES

While there are a growing number of practical applications using expert controllers, there are also a number of unresolved issues. Some of these include:

1. Stability. The use of supervisory control to alter controller parameter values introduces the question of how this affects the stability of the system. Taylor et al have argued that this is actually no substantial change from what has been done for years and considered to be "optimal."

2. Data Base Management. The present ability to monitor plant parameters and performance and adapt the controller to those conditions requires that careful thought be given to how the plant and controller data are managed.

3. Nonmonotonic Reasoning. Nonmonotonic reasoning occurs when previously believed facts (such as proper sensor performance) are no longer held to be true. In this case conclusions which have been drawn from false information must be revised accordingly. This is a difficult problem which has yet to be adequately resolved.

4. Rule-Base Consistency. A key advantage of rule-based systems is the ability to incrementally add rules to improve the performance of the knowledge base. Associated with this feature is the need to ensure that a rule which is added does not conflict with rules which are already in the set of rules. While object-oriented systems offer isolation of rule sets which are internal to the logical objects, the same problem occurs when reasoning about the status of objects within an object-oriented programming environment.

5. Design Optimization. An active area of control system research has been the investigation of methods for optimization of system performance. The introduction of rule-based control systems adds another consideration to the set of conditions under which system performance can be considered to be "optimal."

583
6. **Computer Architecture** As the projects referenced above indicate, many different computer configurations are being investigated. Parallel computing and hypercube architectures are being applied in attempts to increase the processing speed of symbolic processing. It is not at all clear how to build control processors which integrate fuzzy logic, rule-based and conventional control approaches.

7. **Operating Systems** A variety of operating systems have also been used in the knowledge-based systems built thus far. The ability to create, communicate with, interrupt and eliminate multiple computing processes is needed for the most ambitious of these projects. A promising system appears to be real-time UNIX, and

8. **Reliability, Availability, and Maintainability** As we mentioned in Section I of the paper, one reason for pursuing the application of knowledge-based programming techniques to real-time control is the ability to provide structure and incremental expansion capability to the ancillary logic associated with practical controllers. It would be most interesting to see research results concerning the reliability, availability, and maintainability of knowledge-based controllers.

The views expressed herein are those of the authors and do not purport to reflect the position of the United States Military Academy, Department of the Army, or the Department of Defense.

**REFERENCES**


