제어 시스템 설계를 위한 전문가 시스템

권 욱현, °박 재현, 김 상우
서울 대학교 제어계측 공학과

AN EXPERT SYSTEM FOR CONTROL SYSTEM DESIGNS

W.H. Kwon, °J.H. Park, and S.W. Kim
Dept. of Control and Instrumentation Engineering, Seoul National University

ABSTRACT

This paper describes the development of an expert system for control system designs. Our CACSD expert system consists of 2 parts; one is the knowledge manager and the other is the CACSD package. The knowledge manager contains the inference engine and the control experts' knowledge. The CACSD package consists of the control algorithms used by the knowledge manager. The prototype system of our CACSD expert system is implemented on the IBM PC/AT and the results of tests are given in this paper.

1. INTRODUCTION

We can classify the control system design procedure into several tasks such as modeling a plant, analyzing the modeled controller, selecting a proper controller, executing an appropriate design procedure, validating the job, and implementing the hardware.

During this control system design procedure, the computer aided control system design(CACSD) such as IDPAC[1], MATRIXx[2], SIMNON[1,3], is known to be very useful for academic as well as for industrial purpose because these tasks accommodate with heavy mathematical labors and, in many cases the interactive graphics systems make the tedious traditional graphical design job easier[1].

In general, the expert system can be defined as a special software environment designed to aid solving specific problems that require high level of expertise[6,7,8]. The researches on expert systems for a control system design tool are performed since 1980's and some successful cases have been reported in many articles[4,5,10]. But most of them just combine the domain independent expert system tools and the CACSD packages. As a result, they have some limitations to be used as CACSD expert systems. First, since the domain independent expert system tools are intended to be used as general purpose expert systems, the inference engine including control strategy and rule structure, is not well fitted to the CACSD expert system. Second, as most of the general purpose expert systems primarily deals with symbolic data, they can't manipulate mathematical data adequately. Third, in the CACSD expert system an interactive graphics system is inevitable. These features, however, are not so common properties of the general purpose expert system tools. Because of these weaknesses of the general purpose expert system tools, as mentioned above, an expert system with a special architecture is necessary for the CACSD expert system.

In this paper, our CACSD expert system which is specially designed for control system design is presented. In Section 2, the global architecture of our CACSD system is presented. In Section 3 and 4, the two parts of our CACSD expert system, the knowledge manager and the CACSD package, are described in details. In Section 5, the informations about implementing our system are explained.

2. ARCHITECTURE

The CACSD expert system we developed can be divided into two tightly coupled systems; one is the knowledge manger and the other is the CACSD package. These two systems have very different characteristics, and they are connected via communication line. The main block diagram of the total system is shown in Fig. 1.

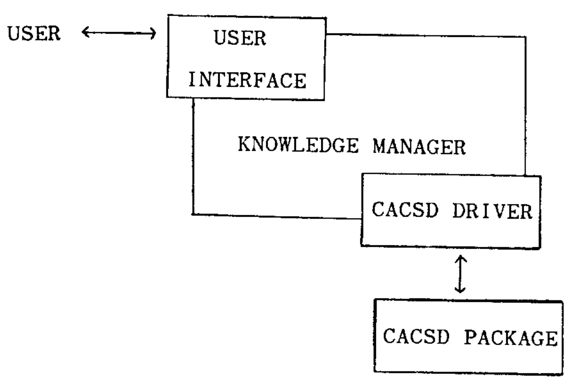

Fig.1 Block diagram of the CACSD expert system.

The knowledge manager is the central part of our expert system. It not only controls all the operations of the expert system but manages all the control experts' knowledge about control system design. Meanwhile, lots of these knowledge demands mathematical calculations, which are offered by the CACSD package. The CACSD package of our system is a stand-alone CACSD system. Functionally, however, it is a part of rule base of the knowledge manager.

3. KNOWLEDGE MANAGER

The knowledge manager is the principal part of our CACSD expert system. Its main task is to interface the user with the expert system and to manage control experts' knowledge. The knowledge manager are divided into 4 parts to its functions; user interface, inference engine, rule base and CACSD package driver (Fig. 2).

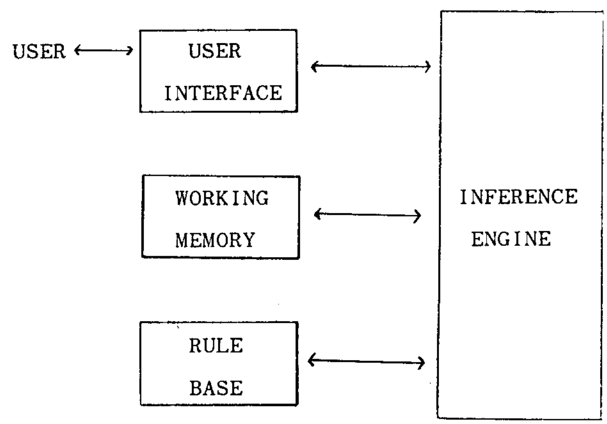

Fig.2 The functional diagram of knowledge manager

In our system, the user interface takes the role to interface the user with the expert system, so its main purpose is to make the user access our expert system easily and to represent the results lively. With the assistances of the user interface, the user can access the resources of our system easily. The implemented user interface has the multi-window systems, the 600 X 350 resolution color graphics systems, the menu drive system, and a mouse as a pointing device.

The inference engine is the nucleus of the knowledge manager. Its operation is determined by heuristic method. In our expert system the inference engine drives three types of jobs:

* Type 1 : What to do next.

* Type 2 : Which controller is adequate.

* Type 3 : How to design the controller.

To do these jobs, the rules of our system are divided into into 3 groups; the operational rule group, the controller-selection rule group, and the controller-specific rule group. These three types of rules are not classified in the rule base, but the inference engine choose appropriate rule at run time by the control strategy.

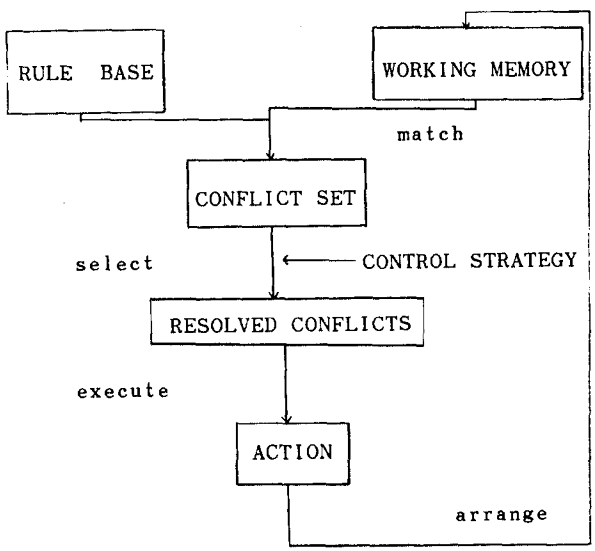

Fig.3 The inference cycle

The inference cycle of the implemented inference engine is shown in Fig.3. This inference cycle is classified into 4 cycles; match-cycle, select-cycle, execute-cycle and arrange-cycle.

In the match-cycle, the inference engine picks up all the applicable rules from the rule base. As the current states are stored in the working memory of the inference engine, the inference engine choose rules whose premises match the working memory's states.

As there are many rules in the rule base, there can be lots of rules that match the current working memory's states. These matched rules makes a conflict set. Since our expert system can do only a function at a time, it should resolve this conflict set. This resolution occurs in the select-cycle. The control strategy used by our system is to select the most specific rule.

The selected rule is executed during the execute-cycle, and the working memory is updated during the arrange-cycle.

The rule base of our system stores all the policies of the operation and control experts' knowledge as rules. The rules in the rule base, are a kind of production system[9]. The basic syntax of the rules in our system is the premises/action pair and optional descriptions. So the rules have three slots(Fig.4). In Fig.4, 'PREMISE' part is the premise slot, 'ACTION' part is the action slot, and 'REMARK' part is the description slot. The description slot is used for explaining the actions of the expert system. The number of premises and actions per rule is not limited.

PREMISE :-
    <u>premise</u> AND <u>premise</u> AND ...

ACTION :-
    <u>action</u> AND <u>action</u> AND ...

REMARK :-
    <u>descriptions</u>.

Fig.4 The implemented rule structure.

All information about the operation of our expert system is stored in the history database. Using the history database, the expert system makes full documentation when it finishes all the design job.

The CACSD package driver acts as a gateway between the knowledge manager and the CACSD package. When the knowledge manager needs mathematical calculations, it requests mathematical operations to the CACSD package via the CACSD package driver. By this way, the knowledge manager can treat the result of calculations of CACSD package as an ordinary fact(rule) in rule base.

4. CACSD PACKAGE

The CACSD package of our system provides the basic mathematical calculations to our CACSD expert system. To design a good control system, a lot of mathematical calculations are inevitable. Hence, a powerful CACSD package is necessary.

In our expert system, for preparing powerful CACSD package on personal computer level, we have designed a CACSD system which can be tightly coupled with the knowledge manager. Our CACSD package is so designed to act as a stand-alone system with interactive graphics system as well as to be used with the knowledge manager. The control algorithms in our CACSD package include not only the programs for SISO control systems, such as PID, Bode plot, Nyquist Plot, etc., but also the programs for MIMO control systems such as LQG/LTR, INA, etc.

5. IMPLEMENTATION

Because the expert system requires a good computing power, most of the expert systems are implemented on mini/mainframe level computers or AI machines. This prevents the user from familiar contact with the expert system.

But we have implemented our CACSD expert system on personal computer level, all the user can access our system easily and comfortably. This is possible because we design all the parts of the expert system in stead of using the general purpose expert system tools which is usually too big to be implemented on PC level.

Our proto-type system is implemented on two IBM PC/AT, one is for the knowledge manager and the other is for the CACSD package. The user interacts with the knowledge manger only, and the CACSD system operates in the background. Some operation screens are shown in Fig.5.

Almost all the parts of the knowl- edge manger is written in PROLOG and only the CACSD package driver is written in assembly language.

Our system adopts the best-first search for fast reasoning, and the control strategy of the heuristic search is to follow the most specific rule[11].

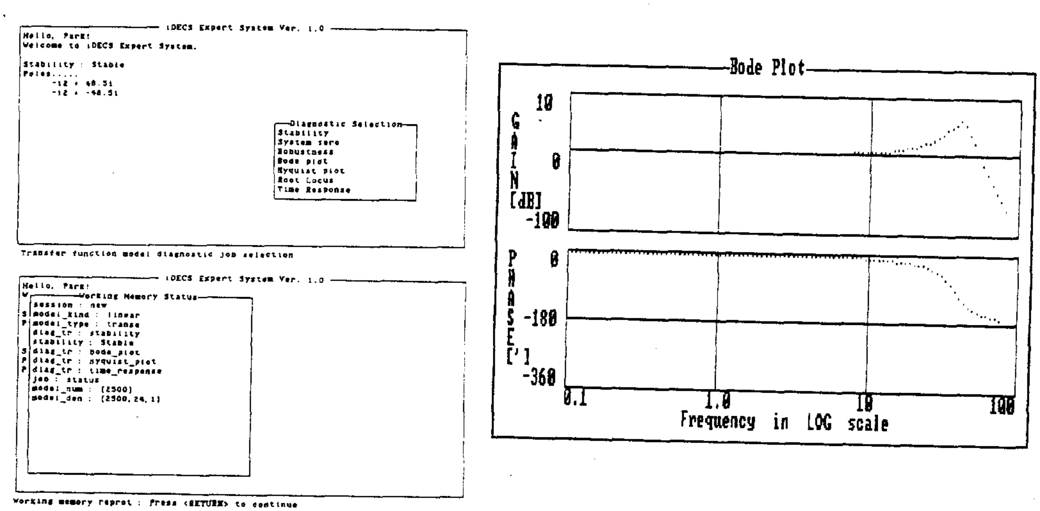

Fig.5 Sample screen

To our experiments, it makes good results in designing controllers. As the matching algorithm, we use the backtracking facility internally-implemented in PROLOG. The size of the knowledge manager is about 5000 lines of PROLOG. The rules currently implemented and tested are mainly for the PID controller; rules to determine whether the PID controller is applicable, and rules to select the PID coefficients.

The CACSD package are written in FORTRAN, C and assembly language. All the control algorithms are written in FORTRAN, and the interactive graphics system is written in C and assembly language. The interactive graphics system is designed for IBM PC EGA system but can easily be changed for other system. Total size of our CACSD package is about 25000 source lines.

6. CONCLUSION

The CACSD expert system in this paper can be divided into two tightly coupled systems; one is the knowledge manger and the other is the CACSD package. With these two systems, the CACSD expert system can manipulate the mathematical knowledge as well as the ordinary symbolic human knowledge. Since it is implemented on PC level, it can be easily accessed and broadly used.

But our proto-type system also proposes several points to be improved. To improve its reasoning speed, its matching algorithm should be replaced with the superior one such as RETE match algorithm.

ACKNOWLEDGEMENT

We would like to thank to all of the students of the Information System Lab. of Seoul National University, specially Dr. S.J. Lee and Y.I. Lee, for useful helps to make the rule base of the CACSD expert system.

REFERENCE

[1] K. J. Astrom, "Computer Aided Modeling, Analysis and Design of Control Systems - A Perspective," IEEE Control Systems Magazine, vol. 3, no. 1, 1983.

[2] R. Walker, C. Gregory, S. Shah, "MATRIXx : a data analysis, system identification, control design and simulation package," IEEE Control System Magazine, vol. 2, no. 4, 1983.

[3] H. Elmquist, SIMNON, an Interactive Simulation Program for Nonlinear Systems. Dept. of Automatic Control, Lund Institute of Technology, Report 7502, Lund, Sweden, 1975.

[4] Thomas L. Trankle, Phillip Sheu, Uri H. Rabin, "Expert system architecture for control system design," Proc. of ACC, WA5-9:30, 1986.

[5] William G. Beazley, "Expert system rules from cad databases," Proc. of ACC, TP5-3:30, 1986.

[6] Donald A. Waterman, A Guide to Expert Systems, Addison Wesley, 1986.

[7] Philip Klahr, Donald A. Waterman, Expert Systems. Addison Wesley, 1986.

[8] Fredrick Hayes-Roth, Donald A. Waterman, Douglas B. Lenat, Building Expert System. Addison Wesley, 1983.

[9] Eugene Charniak, Drew McDermott, Introduction to Artificial Intelligence. Addison Wesley, 1985.

[10] John R. James, Dean K. Frederick, Piero P. Bonissone, James H. Taylor, "A Retrospective View of CACE-III: Considerations in Coordinating Symbolic and Numeric Computation in a Rule-Based Expert System," Artificial Intelligence Applications, Proceedings of 2nd conference, 1985.

[11] Nils J. Nilsson, Principles of Artificial Intelligence. Springer-Verlag, 1982.