Chapter 1

INTRODUCTION

1.1 Motivation

System configurations as well as control specifications and constraints of modern plants are becoming increasingly complex. To deal with the growing complexities of systems, there has been considerable development in control theory. A large number of formulations and approaches have been suggested for solving design and analysis problems of different types of systems. During the last three decades, with the development of sophisticated computer systems, computer aided design approach has received considerable attention. The area of control engineering is no exception to this natural trend and growing interest has been observed in Computer Aided Control System Design (CACSD).

The gradual development of activities in CACSD can be traced back to 1960s and early 1970s. During this period, single purpose routines (like programs for generating Bode plots, Nyquist plots etc.) were developed as "support tools". In order to accomplish a design or analysis task, a control engineer had to run a large number of such packages and had to be conversant with the abilities and limitations of such packages.

To overcome these limitations, a second generation of CACSD packages have been developed. They integrate a number of single purpose packages in a flexible manner around a single unified database and a command driven environment. However, sufficient expertise and knowledge is required in order to make an efficient use of these packages.

Besides, the environment provided by these packages is of rather low level and a user has to remember several hundred system dependent commands. Also, the tasks of posing the problem properly, selecting the packages to be used, performing design trade offs as well as interpreting the results and validating the design had to be performed by the users. As a result these packages were difficult to be used by "less than expert" users, who lack a comprehensive knowledge of the recent developments in control engineering techniques and tools.

In this perspective, the necessity of a flexible, user friendly and yet powerful technique have been felt, specially for the "less than expert" users. Expert system techniques appear to be a possible solution to the above problem. Expert systems are knowledge based software environments which can utilise the inbuilt domain knowledge in an intelligent and effective manner to achieve a particular goal. The promise of the expert system approach to solve the present day CACSD challenges can be attributed to the reasons enumerated below :

i) The encoded knowledge and experience of an expert enable the system to make proper choices of design and analysis tools and apply them in a systematic manner.

ii) Since an expert system can validate the design, users can be relieved of trial and error method of iterations. In case of an unsatisfactory design, the system can try other alternatives to arrive at an acceptable design.

iii) Incorporation of heuristics permits an expert system to achieve a "reasonable" solution when no straight forward solution yields a design that meets all the constraints perfectly.

iv) Intricacies of the different techniques and packages can be dealt by the system, by using the resident knowledge base.

v) Expert systems can provide suggestions regarding the packages to be executed to achieve a design or analysis goal. Alternatively, it is possible for the system to execute suitable packages, interpret the results and present the ultimate solution to the user.

vi) User friendliness is an inherent feature of any expert system . It allows the user interactions to be performed at a higher level and in an interactive manner. In the process the system can guide a user to present the proper formulation of the problem and to provide any other information required. The user supplied information can also be validated by the system.

vii) Expert system can explain the decisions and actions taken, to the user.

viii) If proper knowledge representation scheme (like rule based representation) is adopted, knowledge can be easily modified and added to the system without affecting other interfacing software.

In view of the above advantages of expert systems, design and developent of expert systems for control engineering have already gained momentum. James [J1], Taylor [T2], Pang [P1], Sanoff and Wellstead [S1] , Astrom [A7] and others have made significant contributions in this field.

However, keeping in view the special features of the control engineering domain, major issues like knowledge representation and inference mechanism call for further attention. These aspects have specially been emphasised in this thesis.

1.2 Scope of the work

The present thesis describes "CONEX" - a rule based expert system structure for control system design and analysis, and demonstrates the applicability of expert system concepts to CACSD activities. CONEX aims at providing the less than expert users, with a knowledge based environment to deal with the various problems and intricacies of control engineering.

Two key issues related to the design of an expert system are the knowledge representation scheme and the inference mechanism. Closely related to these, is the organisation of facts generated. The major issues addressed in the thesis are briefly outlined below.

The present thesis first identifies some requirements of the knowledge representation scheme, so that it suits the specific needs of CACSD activities. CONEX knowledge has been organised accordingly.

In CONEX, the kernel knowledge is formed by various control engineering packages, which accomplish the design and analysis tasks. Depending on the particular design or analysis goal, system description, design specifications and constraints, a particular package or a set of packages needs to be applied to achieve the goal. While selecting appropriate packages, several features such as numerical stability, robustness and computational complexities of the algorithms should be given due consideration. Unfortuantely, these features are often not well specified and experts' experience plays an important role in package selection. Sufficient expertise is also required for selecting controller configurations, design strategies and methods suitable for meeting the constraints and specifications in the best possible manner. These decisions are often heuristic in nature . Such expertise and heuristics are embodied in CONEX in the form of a rulebase. The production rules constituting the CONEX rule base are of the type -

IF <antecedent field> THEN <consequent field>.

Different strategies have been adopted to reduce search overheads, incurred for identifying the enabled rules.

Since the rules work in close coordination with the collection of facts, sufficient emphasis has been laid on their organisation. Some of these facts are provided at the time of initiation, to serve as the initial premise of CONEX activity. Facts are also generated or modified, at the time of execution, by the application of the rules.

In CONEX every feature of a control system problem is viewed as an information object having several attributes. Corresponding to these features, the facts have been organised as a collection of frames, each having a number of slots to store the attribute information. The organisation of facts as a collection of frames results in a modular structure and permits easy and flexible referencing. Other requirements posed by the nature of CACSD activities have also been taken into account while designing the scheme. Moreover, the organisation lends active support to the inference machine, by eliminating irrelevant rules and also by maintaining a history of the decision steps taken.

The entire process of decision making by CONEX is centered around an inference machine (INFER). Following forward chaining process of inferencing [N4], it selects the applicable rules, uses suitable strategies for conflict resolution and fires the rules to infer new facts. A distinctive feature of the inference machine is its ability to adopt different conflict resolution strategies for different types of rules. Often the conflicts are resolved based on heuristic measures associated with the rules. For specific cases provisions are also there to evaluate heuristic functions at run time and thereby resolve the conflicts.

The inference machine supports "intelligent" as well as chronological backtracking. This allows CONEX to search for alternative solutions. The architecture of INFER has been tailored to make efficient inferencing. This has been accomplished by ensuring that search for a solution is restricted to a limited and relevant set of rules only.

INFER is coupled with a flexible user interface module and an explanation generator. The former permits menu driven user interactions while the latter endows CONEX with the ability to explain the decisions taken, in a user friendly manner.

CONEX has been implemented on HP9000 system at IIT Kharagpur and has been applied to a number of control engineering problems. Some of the design problems selected have been previously solved using "non expert system" approaches. The solutions obtained by CONEX demonstrate that expert systems can attain similar results in a more user friendly manner.

The present repertoire of control knowledge, in CONEX, mainly pertains to the state space approach of design and analysis. However, the structure of the system is general enough to support frequency domain and other approaches in control engineering by enhancing the rule base.

With the advent of VLSI technology and parallel processing techniques, architectures for parallel inference have received considerable attention. A major overhead of inferencing in rule based systems is incurred in the pattern matching phase for finding the rules that can be fired at any instant. In the present thesis an alternative parallel inference machine architecture (INFLOW) has been proposed. The proposed architecture attempts to alleviate the pattern matching overhead. Alongwith it, parallelism at the rule execution level has also been exploited wherever possible. The architecture has been simulated and it has been shown that considerable speed up in inferencing can be achieved compared to the sequential inference machine architecture of CONEX.

1.3 Layout of the thesis

A brief overview of the chapters following and their contents is presented below.

Chapter 2 : A review of related works has been presented in this chapter. Different techniques for knowledge representation and existing expert system shells have been discussed. Expert systems for CACSD have been emphasised.

Chapter 3 : This chapter discusses some design issues control engineering with examples. Different phases of control engineering activity have been identified. The requirements of a knowledge representation scheme aimed at control engineering applications have been discussed. The last section of this chapter presents an overview of CONEX.

Chapter 4 : The details of the knowledge representation scheme adopted in CONEX has been presented. A frame based organisation of facts has been described. The organisation of the CONEX rules and the interaction between frames and rules have been explained.

Chapter 5 : This chapter deals with the architecture of the inference machine of CONEX . Important data structures and functional modules of the inference machine have been described. The backtracking mechanism has been explained in detail. Different conflict resolution strategies and the heuristic approaches adopted for run time resolution of conflict resolution have been considered. The user interface, explanation generation module and the features of the knowledge editor have been discussed.

Chapter 6 : CONEX has been applied to solve a few real life control problems . This chapter describes the decision process of CONEX. This has been illustrated through typical CONEX sessions. The results obtained by CONEX have been compared with solutions obtained through other techniques. The conformity of the results demonstrates the ability of CONEX to solve control engineering problems.

Chapter 7 : This penultimate chapter develops the concept of "inference flow graph" and consequently develops the parallel inference machine architecture, INFLOW. The efficiency and advantages of this architecture have also been presented through simulation results.

Chapter 8 : In conclusion, the major aspects of the thesis have been summarised and future scope of development has also been discussed.