5.1. Introduction.

This chapter discusses the implementation of the expert system using the facilities provided by the CXS expert system shell. It begins by listing the important features to be considered when selecting an expert system shell, and then describes two of those features - the communication with external programs, and the database facilities - in detail. A discussion of the overall structure of the expert system follows, which includes examples of typical rules form the knowledge base. Here particular emphasis is placed on those sections of the expert system which assist the designer in formulating the design specification, and in dealing with any conflicts in the performance constraints.

5.2. Selection of the Expert System Shell.

The following properties were considering important in selecting an expert system shell for implementing an intelligent design system:

- The shell should provide an efficient and effective method of communication with external, preferably memory-resident, programs. Numerical analysis algorithms are best implemented in a conventional programming language, and the expert system should be able to interact with these external programs easily.

- The shell must be suitable for implementing non-monotonic logic systems. Being a feedback process [29], control system design forms an inherently non-monotonic logic system [19].

- The shell should provide flexible knowledge representation capabilities. The nature of the multivariable problem makes database facilities attractive.

- The shell should have flexible user interface facilities. A significant part of the expert system is devoted to interaction with the designer, and this portion should be both easy to use and program.
The shell must support a large knowledge base, and should execute efficiently. Control system design is a complex procedure, and thus requires a considerable knowledge base.

The memory requirements (RAM) of the expert system should be modest. Ideally it should be co-resident with the CACSD package in RAM.

The shell should support at least simple arithmetic computations.

The shell must run under the MS-DOS operating system on an IBM-PC type computer, and must be very modestly priced. These two constraints were demanded by limited research funding.

None of the existing expert system shells examined for the IBM-PC (VP-Expert, Synapse, K-Shell, dmX, and Personal Consultant) satisfied all of these requirements. A solution was eventually obtained by extending the CXS expert system shell written previously by the author [58]. Some of the special features of this shell, and how they are used in building the expert system, are detailed below.

CXS is a backward-chaining rule-based expert system shell for the MS-DOS operating system. At present the knowledge-base contains approximately 400 rules.

5.3. Communication with External Programs.

An efficient interface to external software is essential for a high-performance design system. CXS allows linking to external memory resident programs, in this case the CACSD package described in chapter 3, through a remote procedure call scheme based on MS-DOS interrupts. The external programs have access to an array of expert system variables, with elements named T0, T1, ..., T99, each of which may contain symbolic or numeric values. This interface is used by the expert system to execute functions such as loading and saving transfer functions, plotting step or frequency responses, translating the specification into a quadratic programming problem, or examining the status of a performance constraint. Details of this interface are given in appendix B. A typical set of rules using this interface, in this case to plot the singular value (SVD) frequency response, is shown below.

```
IF LINE cmdline ["svd"]
  THEN
    FIND `set_response` :If the SVD command was entered
    T0 = "freq" :Use the rule below to assign the currently
    INTR 96,35,0 ;selected response to the array variables.
    INTR 96,1021,0 ;Always use the frequency response
    cmd_action is "done"

```

If the SVD command was entered, the rule below assigns the currently selected response to the array variables. Always use the frequency response.