Proceedings of the 1996 IEEE International Symposium
on Computer-Aided Control System Design
Dearborn, MI • September 15-18, 1996

A Knowledge-Based Learning System for H^∞ Control System Design

M. W. Ngai and Y. S. Hung

Department of Electrical and Electronic Engineering
The University of Hong Kong, HONG KONG
mwngai@hkueee.hku.hk
yshung@hkueee.hku.hk

G. K. H. Pang

Department of Electrical and Computer Engineering
University of Waterloo, Waterloo, Ontario, CANADA N2L 3G1
pang@kingcong.uwaterloo.ca

Abstract

A knowledge-based approach to H^∞ computer-aided control system design is considered. The system is made up of a knowledge-based expert system and a learning system. A knowledge structure comprising a design knowledge base, a general knowledge base and a meta-knowledge base is proposed to facilitate learning. Two different learning strategies, learning through critic and learning-from-discovery, are considered for refining the design knowledge base.

1. Introduction

H^∞ theory offers a powerful approach to control system design. There are many computer-aided control system design (CACSD) packages supporting analysis and design tools for H^∞ design. These packages provide sophisticated computational and display facilities for the user to execute complex algorithmic procedures and present the results in graphical forms. The user is however left to make all the other design decisions in the design process based on his/her engineering knowledge. There are two particular difficulties associated with an H^∞ design. First, the complexity of the H^∞ design methodology and the range of possibilities in the choice of the weighting functions (i.e. the design parameters) mean that only an expert user can realize the full capabilities of the H^∞ approach. Secondly, while design specifications are often given in terms of time-domain conditions, the H^∞ approach is a frequency-domain based technique. Therefore, the user has to provide the relationships between the two in the iterative design process. A good knowledge of such relationships holds the key to the success of an H^∞ design because the user has to decide how the frequency response of the weighting functions should be tuned when the time response of the design fails the specifications in some particular way.

In this paper, we will consider how a knowledge-based learning system can be used to alleviate the difficulties of an H^∞ design. It is felt that a knowledge-based expert system can be used to capture the intricacies of the H^∞ approach and guide the user in the design process, while a learning system can be used to help refine case specific design rules which relate time-domain design outcomes to frequency-domain design parameters. The use of expert system techniques in control system design has been considered by various authors (e.g. see [1],[2],[3],[4]). The knowledge-based system to be proposed here differs from the previous work in the use of the H^∞ approach as the design methodology and/or in the learning capability of the system. By learning is meant the ability of the system to modify its own knowledge base in the course of the design process. For our purpose, the learning requires that the expert system has the ability to analyze the cause of success or failure of a design, and a means to incorporate the results of such an analysis into the knowledge base.

The paper is organized in the following way. In section 2, an outline of the typical H^∞ control system design process is given and the role of an expert system and a learning system for H^∞ design is discussed. A general framework of an expert system for CACSD is proposed in section 3. The learning aspects of the system is considered in

0-7803-3032-3/96/$5.00 © 1996 IEEE

section 4. Some concluding remarks are given in section 5.

2. H^∞ control system design

There have been growing interests in the use of H^∞ optimization techniques for control system design (e.g. see [5],[6],[7]). Doyle [5] has suggested a standard H^∞ control problem having a feedback configuration shown in Fig. 1, where $P(s)$ is a generalized plant and $K(s)$ is a controller to be synthesized by minimizing the H^∞ norm of the transfer function matrix from w to z. This standard configuration embraces many H^∞ control problems (e.g. robust stabilization, multi-objective design) and different solutions have been obtained [8], [9].

For the sake of illustration in the remaining of this paper, we will only consider H^∞ design based on the robust stabilization problem, with a configuration shown in Fig. 2 where $G(s)$ is the plant and $K(s)$ is the controller to be synthesized. For robust stabilization, the H^∞ design objective is to ensure that the closed-loop system has a good stability margin while maintaining a certain degree of performance. This can be achieved by minimizing (the norm of) two transfer functions, namely the sensitivity function

$$S(s) = (I + G(s)K(s))^{-1}$$

as a performance measure and the function $K(s)S(s)$ as a robustness indicator for additive perturbations. The H^∞ control problem is:

$$\min_{K(s)} \left\| \begin{matrix} W_1(s)S(s) \\ W_2(s)K(s)S(s) \end{matrix} \right\|_\infty \leq \gamma \qquad (1)$$

This corresponds to minimizing the H^∞ norm of the transfer function from w to $\begin{bmatrix} z_1 \\ z_2 \end{bmatrix}$. The weighting functions $W_1(s)$ and $W_2(s)$ are introduced to impose proper frequency weightings on $S(s)$ and $K(s)S(s)$, respectively. $W_1(s)$ and $W_2(s)$ can be regarded as the design parameters and their choices have a critical effect on the success of the H^∞ design.

H^∞ control system design process

Fig. 3 shows an H^∞ control design process. We will first consider the outer loop which is modeled on a typical H^∞ design cycle. The supporting role of the expert and learning system will be discussed in next two subsections.

Suppose a set of design specifications is provided to start off the design process. The specifications can be expressed in terms of the step response of the closed-loop system (e.g. maximum overshoot, rise-time, steady-state error) as well as the system's ability to resist uncertainties (e.g. disturbance rejection, robustness margin). The design problem is to select weighting functions $W_1(s)$ and $W_2(s)$ according to the specifications. This is the point where the designer provides his/her input to the design process, and where an expert system with learning capability can provide support. The purpose of the weighting functions is to shape the frequency responses $S(s)$ and $K(s)S(s)$ into desirable forms. H^∞ optimization is applied to synthesize the controller. Frequency-domain evaluation and time-domain simulation are then performed to obtain the design outcome. At this stage, the designer has to compare the specifications against the design outcome. If the design outcome fails to meet the specifications, the designer has to make judgment on how the choices of the weighting functions cause the specifications to be violated and uses this as a feedback to tune the weighting functions in the next iteration of the design cycle.

The main issue in the design process is what shape the weighting functions should take in order to produce a controller which will meet the specifications. The H^∞ design methodology is based on the assumption that the weighting functions can be used to effectively shape the appropriate frequency response functions (i.e. $S(s)$ and $K(s)S(s)$ in our case) and that the designer knows how these functions should be shaped in order to meet the specifications (including time-domain ones). The difficulties are twofold. First, although H^∞ optimization is often quoted as a frequency-response shaping technique, $W_1(s)$ and $W_2(s)$ cannot be used to exactly define the shapes of $S(s)$ and $K(s)S(s)$ because of the necessary compromise that will have to be made when both functions are optimized in a combined problem. Secondly, although there are heuristic relationships between the shapes of $S(s)$ and the step response, such relationships are not analytical and hard to exploit in a design process. It is proposed that an expert system can play a significant supporting role here. It should however be noted that there are other approaches to the tuning of weighting function based on numerical optimization-type techniques (e.g. see [10],[11]).

Role of expert system in H^∞ design

Referring to the design process depicted in Fig. 3, the role of the expert system is to act as a design assistant to guide the user during the design process and to provide explanations for the reasoning process. It particularly suggests any trade-offs that should be made among the design goals and assists the user in the choice of the weighting functions. In order to do so, it has to take inputs from varies points of the design process and compare design outcomes against the design specifications. The user can choose his own desired weighting functions or seek recommendations from the expert system. The expert system is to include a knowledge-based system (KBS) which captures the expert's knowledge and encodes it in rules form (with IF <condition> THEN <action> representation). The main functions of the expert system in the design process are:

(1) *initialization*

Given the design specifications, the expert system will analyze the plant (e.g. stability, locations of zeros etc) and the conditions of the specifications (i.e. overshoot, rise-time, steady-state error). It will make use of the expert knowledge in its knowledge base to transform the specifications into some frequency-response envelopes (which will be referred to as the transformed specifications) for $S(s)$ and $K(s)S(s)$. The weighting functions are then generated within these bounding envelopes.

(2) *design result assessment*

After an H^∞ control optimization, the expert system will analyze the design result through frequency-response evaluations and time simulations. For example, a time-domain analysis will yield the percent overshoot, rise-time, and the steady-state error (sse) of the step response. A frequency-domain analysis will give the dc gain, the closed-loop bandwidth and the location of the resonant peak (if any). These design outcomes are verified against both the original design specifications and transformed (frequency-domain) specifications, and the results are fed to the learning system for critical evaluation.

(3) *design iteration*:

In each design iteration, our approach will be to set a single design goal (e.g. to reduce the rise-time) aimed at steering the design outcome closer to the original specifications. The determination of the design goal is based on the assessment of the current and past design results obtained in (2) above. After the design goal is set, the expert system has to translate it into 'actions', namely the tuning of weighting function parameters for H^∞ optimization.

In the design iteration, the expert system makes use of the knowledge base containing rules relating weighting functions to the conditions of the specifications. These rules are based on heuristic reasoning and it is felt that there is scope for refining the rules specifically for the system under consideration. This leads us to introduce a learning system to modify the knowledge-base of the expert system.

Role of learning in H^∞ design

In any iterative design process, the designer gains task-specific knowledge about the system under consideration as the design progresses. However, in a complex design, the designer can rarely make good use of negative design examples to guide him in the iterative process. The role of the learning system is to perform a critical evaluation of all the steps taken in the current design, to identify the cause of success or failure of the design, and to assign credit or blame to the rules which are judged to be associated with the success or failure. Credit or blame can be given to a rule by strengthening or weakening its responsible elements. Rules which do not 'work' will also be considered for more drastic action such as modification or deletion from the knowledge base. In short, a learning system should be able to use the past design experience to improve the expert system's inference performance.

3. General framework of expert system for CACSD

An arrangement of the proposed knowledge-based learning system for CACSD is shown in Fig. 4 which comprises an expert system and a learning system. The expert system has an inference engine, a data base and a design knowledge base. The design knowledge base contains rules which will be used directly to support design process. In addition to the design knowledge base, a general knowledge base and a meta-knowledge base are defined for the purpose of learning, with the following characteristics.

General-knowledge base

The general-knowledge base is to store the domain-independent knowledge which includes the concept definitions such as set theory, logic and the mathematical relationships. For example, the terms 'proportional to', 'inversely proportional to' etc. will be needed in specifying the relationships between variable names. These are general terms which have to be defined explicitly for the system to make sense of the design rules. The general knowledge can also be expressed in the IF-THEN rule format but it is not executable. e.g. Two rather coarse definitions of the concept "proportional to" can be stated as:

IF x is *inversely proportional* to y and x is *decreased* THEN y is *increased.*
IF x is *inversely proportional* to y and x is *increased* THEN y is *decreased.*

Meta-knowledge base

The meta-knowledge base is to store the domain-specific meta-knowledge. By definition, meta-knowledge means knowledge about knowledge [12]. The meta-knowledge is represented by means of meta-rules. Each meta-rule is associated with only one specific rule in the design knowledge base, and not all the rules have meta-rules. Each meta-rule comprises two parts: the 'Expectation' and the 'Justification'. The 'Expectation' is to specify the expected result or to define the aim of the associated rule when the rule is executed. The 'Justification' can be regarded as a true statement which provides an explanation of the reasoning underlying the rule. The logical relationship between the meta-rule and the associated design rule can be stated as: if 'Justification' and the condition of the associated rule is true and the action part of that rule is executed, then the 'Expectation' is implied. This is illustrated in the following example.

Design Rule:

IF 'sse of step response' is large,
THEN increase the dc gain of W_1.

Meta-Rule:

Justification:

'sse of step response' is proportional to 'dc gain of $S(s)$', and

'dc gain of $S(s)$' is inversely proportional to 'dc gain of W_1'.

Expectation:

'sse of step response' is decreased.

The aim of the above rule, as stated in the 'Expectation', is to reduce the 'sse of step response'. The sse cannot be manipulated directly but is related to the 'dc gain of W_1' through the relationships stated in the 'Justification', which explains why the action given in the design rule leads to the 'Expectation'. The design rule is fired only if its condition is satisfied. The outcome of the action will be checked against the 'Expectation' by the learning system.

4. Learning system

The learning system shown in Fig. 4 employs two different learning strategies, that of a 'critic' and a 'discoverer', with distinct learning objectives and learning methods. The aim of the critic is to perform a critical evaluation of the current design to prevent negative experiences from recurring, whereas the aim of the discoverer is to discover new facts or new relations for incorporating into design rules. Both strategies are aimed at improving the performance of the knowledge-based expert system.

A learning mechanism for the critic is suggested by Hayes-Roth F. et al [13]. The critic uses two sets of heuristic rules, referred to as diagnostic rules and learning rules, to perform its job. The diagnostic rules are to identify problematic knowledge by comparing the prior beliefs (i.e. expectation) with the actual (design) outcome. This identifies the cause of success or failure of the current design and detects faulty rules. A faulty design rule is one that has a false premise or entails a false conclusion. Once identified, the learning rules suggest fixes to any erroneous design rule. The purpose of the critic is therefore to prevent the same faulty result from happening again in a similar situation.

The objective of the discoverer is to learn new facts and relations through observation of past design examples, and to examine the implications of the new facts on design knowledge. The new domain of knowledge can be developed mechanically by using heuristics and is discussed in [14]. For H^∞ control system design, one of the applications of this technique is to find the mathematical relationship (i.e. fact) between variables, e.g. whether two variables x and y are directly or inversely proportional. This kind of relationship is easy to verify, but it is less straightforward to incorporate newly discovered facts in the knowledge base. A proposed mechanism to do this is as follows. As a new fact N is discovered, it is checked against the 'Justification' of each meta-rule. Suppose inconsistency is found in a meta-rule M, i.e. any incorrect statement which contradicts *N* in the 'Jusification'. Then the associated rule *D* has to be modified so that the 'Expectation' *E* of the meta-rule *M* remains valid. To do this, the implications of the new fact *N* are examined by applying the general knowledge base to the new fact. If a rule in the general knowledge base with premise *P* is selected whose consequence *Q* matches the 'Expectation' of the meta-rule *M*, then the 'Justification' of *M* is replaced by *N* and the consequent of *D* will be replaced by *P*.

The learning system will inform the user any newly discovered facts and recommendations of possible modifications of the knowledge base. The user can decide whether to allow the learning system to make the modifications.

5. Conclusion

We have considered the application of knowledge-based systems and learning techniques to H^∞ control system design. A knowledge structure comprising a design knowledge base, a general knowledge base and a meta-knowledge base is proposed to facilitate learning. Two different learning strategies, learning through critic and learning-from-discovery, are considered for refining the design knowledge base. The choice of the learning techniques relies on the domain-specific application. But the general framework of the knowledge-based expert system and the learning system described in this paper can be used for control system design methodologies other than the H^∞ approach.

6. Reference

[1] J. R. Taylor, "Expert-Aided Environments for CAE of Control Systems," IFAC Computer-Aided Design in Control System, Beijing, P. R. C., 1988.

[2] J. R. James, D. K. Frederick, and J. H. Taylor, "Use of Expert-System Programming Techniques for the Design of Lead-Lag Compensations," IEEE Proceedings Part-D, vol. 134, no. 3, pp. 139-144, 1987.

[3] G. K. H. Pang, J. M. Boyle, and A. G. J. MacFarlane, "The Development and Implementation of MAID: a Knowledge Based Support System for Use in Control System Design," Trans. Inst. MC., vol. 11, no. 1, Jan-March, 1989.

[4] I. Postlethwaite, D. W. Gu, S. J. Goh, and J. F. Whidborne, "An expert system for robust controller design," Proc. Control 94, pp. 1170-1175, Conventry, England. March, 1994.

[5] J. C. Doyle, "Advances in Multivariable Control," ONR/Honeywell Workshop, Minneapolis, MN, 1984.

[6] B. A. Francis, "A Course in H^∞ Control Theory," Lecture Notes in Control and Information Sciences, vol. 88, NY, Springer-Verlag, 1987.

[7] K. Glover, and D. C. McFarlane, "Robust Stabilization of Normalized Coprime Factor Plant Description with H^∞-bounded Uncertainty," IEEE Trans. Automatic Control, vol. 34, pp. 821-830, 1989.

[8] Y. S. Hung, "H^∞ Optimal Control - Part I: Model Matching and Part II: Solution for Controllers," International Journal of Control, vol. 49, no. 4, pp. 1291-1330 & 1330-1359, 1989.

[9] K. Glover, and J. Doyle, "State-space formulae for all stabilizing controllers that satisfy and H^∞-norm bound and relations to risk sensitivity," System and Control Letters, 11, pp. 167-172, 1988.

[10] J. F. Whidborne, I. Postlethwaite, and D. W. Gu, "Robust controller design using H^∞ loop-shaping and the method of inequalities," IEEE Transactions on Control Systems Technology, 2(4), pp. 455-461, 1994.

[11] J. F. Whidborne, G. Murad, D. W. Gu, and I. Postlethwaite, "Robust control of an unknown plant - the IFAC 93 benchmark," International Journal of Control, 61(3), pp. 589-640, 1995.

[12] F. Hayes-Roth, and D. A. Waterman, "Building Expert Systems," Addison-Wesley, 1983.

[13] F. Hayes-Roth, P. Klahr, and D. J. Mostow, "Advice Taking and Knowledge Refinement: An Iterative View of Skill Acquisition," Cognitive Skills and Their Acquisition, Earlbaum, 1981.

[14] D. B. Lenat, "The Role of Heuristics in Learning by Discovery: Three Case Studies," Machine Learning: An Artificial Intelligence Approach, vol. 1, Springer-Verlag, 1983.

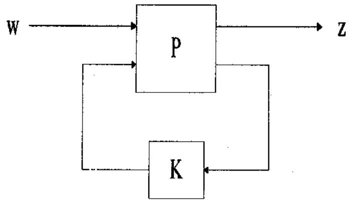

Fig. 1

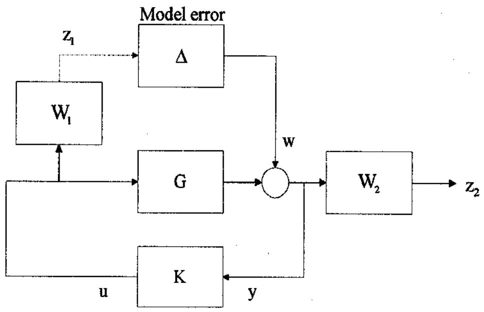

Fig. 2

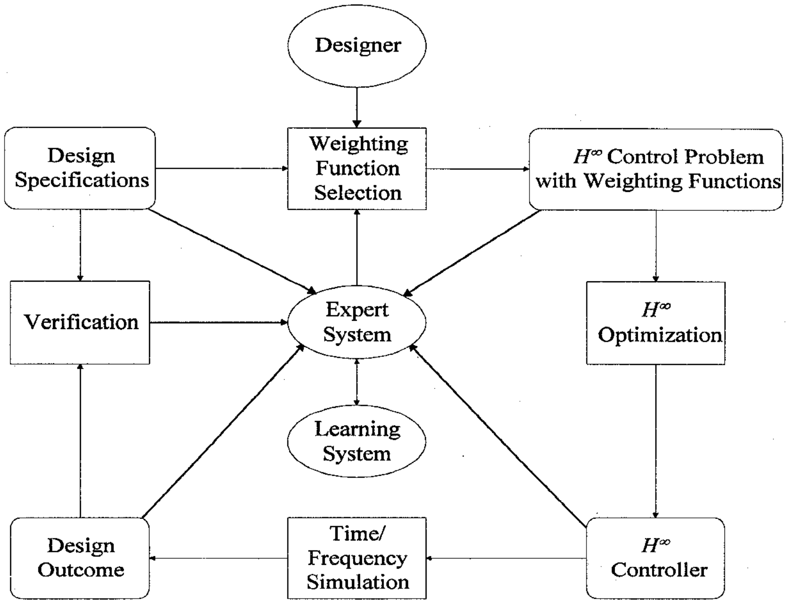

Fig. 3

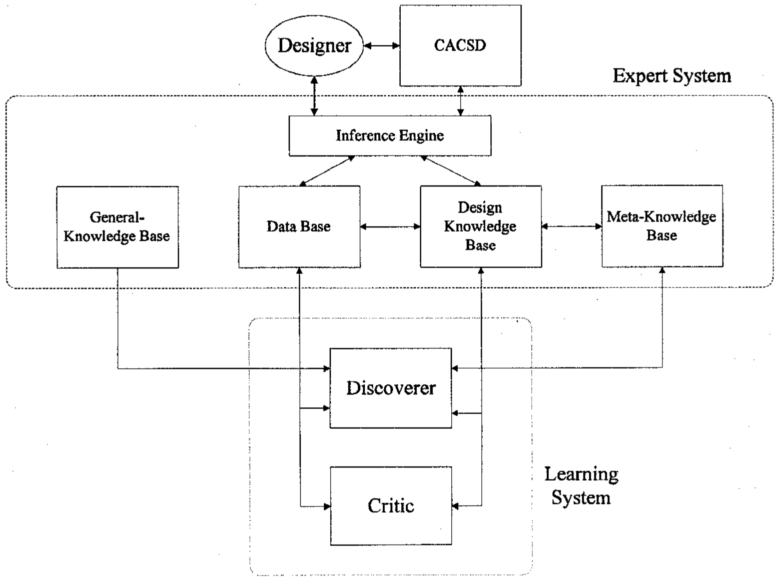

Fig. 4