Journal of Intelligent and Robotic Systems **21:** 239–256, 1998.

Control System Design Automation for Mechanical Systems

KIYOSHI MAEKAWA
Industrial Electronics & Systems Laboratory, Mitsubishi Electric Corporation 8-1-1, Tsukaguchi-Honmachi, Amagasaki, Hyogo 661, Japan; e-mail: maekawa@fas.sdl.melco.co.jp

GRANTHAM K. H. PANG
Department of Electronics & Electronic Engineering, The University of Hong Kong, Pokfulam Road, Hong Kong; e-mail: gpang@hkueee.hku.hk

(Received: 10 December 1996; accepted: 5 September 1997)

Abstract. In this paper, a user-friendly and comprehensive control system design package called Control System Design Automation (CSDA) is described. The system consists of five main blocks: a requirement interpretation block, a modeling block, an analysis/design block, a database management and knowledge base block, and a verification block. The requirement interpretation block transforms the specifications in terms of the application to those in terms of control. The analysis/design block selects an optimal control structure and determines the controller parameters. In addition to the conventional design methods, CSDA also contains the more recent design methods such as the LMI design approach and the Kessler/Manabe method. The LMI approach can obtain a controller which satisfies multiple specification items at the same time. The configuration of the system as well as the analysis/design block are described in detail in this paper.

Key words: control system design, mechanical systems, feedback control, design automation, computer-aided control system design.

1. Introduction

Currently, many control system design problems are tackled by skillful control engineers in the industry. One objective of developing the Control System Design Automation (CSDA) is to make control system design methods available to more engineers so that they can carry out control system design and obtain a satisfactory controller in a short time. The users of the CSDA system includes robot designers and machine tool designers, whose training are more in the area of mechanical engineering. The aim is that those engineers would have a user-friendly and comprehensive control system design tool with them when they carry out their design of mechanical systems. The particular targets are industrial robots, X-Y tables, and transfer machines.

Nowadays, the Computer-Aided Control System Design (CACSD) tools, such as MATLAB/SIMULINK and MATRIXx, have high performance in analysis, simulation and rapid prototyping. Those tools are very useful for a skilled engineer to design a controller. Further work have been carried out in constructing

CACSD tools with which optimal control parameters for a given control structure are automatically calculated [1, 2]. These tools, however, cannot transform the requirements in the application to the specifications items for use in CACSD tools. Furthermore, they do not select an optimal control structure automatically. To overcome these problems, a high level and user-friendly control system design package is constructed for controller design of mechanical systems. The control system design environment is called CSDA. The special features of the CSDA system include the following:

(1) The requirements in a particular application may not be expressed in terms of the control design specifications. The CSDA system can transform those application requirements to specifications which can be treated in the control design package.

(2) The CSDA system has a comprehensive set of control design methodologies. Some of the more recent methods such as the LMI design approach and the Kessler/Manabe method are included.

(3) The system can select an appropriate control structure automatically in addition to determining a set of control parameters.

(4) A user-friendly interface has been developed using the Graphical User Interface (GUI) in MATLAB.

(5) Analysis of the designed control system can be carried out with different kinds of command input. The aim is to select a suitable command control signal in the final control system.

(6) Verification of the control system design can be carried out with the DSP-CIT system [24].

(7) A knowledge base module in the CSDA system can provide guidance to the user during the design process. Heuristic knowledge from previous design experience can be codified and make available to the user.

(8) A database management module in the system can help to organize the design results of the user. A database on the past design activities can also help the designer to decide on the appropriate control structure and controller parameters. The objective is that users of the CSDA system can design satisfactory controllers easily even if they do not have much experience in control system design. Fine tuning of the controller can be carried out by a parameter optimizer.

In this paper, the configuration of the CSDA system is described in details. Also, detailed description is given to the analysis/design block. The analysis/design block is the most significant block of CSDA, which provides the desired control system design automation.

2. Configuration of the System

The configuration of the CSDA system is shown in Figure 1. The CSDA system consists of five blocks: a requirement interpretation block, a modeling block, an analysis/design block, a database management and knowledge base block, and a verification block. The modeling block, the analysis/design block, and the simulation module of the verification block are constructed using MATLAB/ SIMULINK.DSP-CIT (developed by dSPACE GmbH) [21] is used for a rapid prototyping module in the verification block.

2.1. REQUIREMENT INTERPRETATION BLOCK

Sometimes, the requirements in a particular application may not be expressed in terms of the control design specifications. In this block, the given requirements from the application are transformed to specifications which can be treated in the control design package. Since the requirements differ from application to application, the requirement interpretation block has specified modules according to each controlled object.

For example, when the controlled object is an industry robot, the requirements in the application are standard cycle-time, positioning accuracy, and tracking accuracy. The control design specifications are crossover frequency, amount of overshoot, and settling time. Acceleration time and deceleration time are calculated, and standard cycle-time is transformed to settling time. The acceptable limit of overshoot is calculated from the positioning accuracy. Crossover frequency is decided according to the required tracking accuracy and maximum speed. The specifications are then passed on to both the analysis/design and the modeling blocks.

2.2. MODELING BLOCK

In the modeling block, the model of a controlled object is obtained by an actual measurement or from the design parameters of the controlled object. In case the order of the model is too high for designing a controller, the order of the model needs to be reduced. For example, a three-mass model is approximated by a two-mass model if the resonant frequency of the second vibration mode is much higher than the crossover frequency. The reduced order model is then sent to the analysis/design block.

2.3. ANALYSIS/DESIGN BLOCK

The analysis/design block has six modules: a control structure selection module, a control parameters calculation module, a pre-design analysis module, a post-design analysis module, a command signal generation module and a parameter optimizer module. The control structure selection module and the command sig-

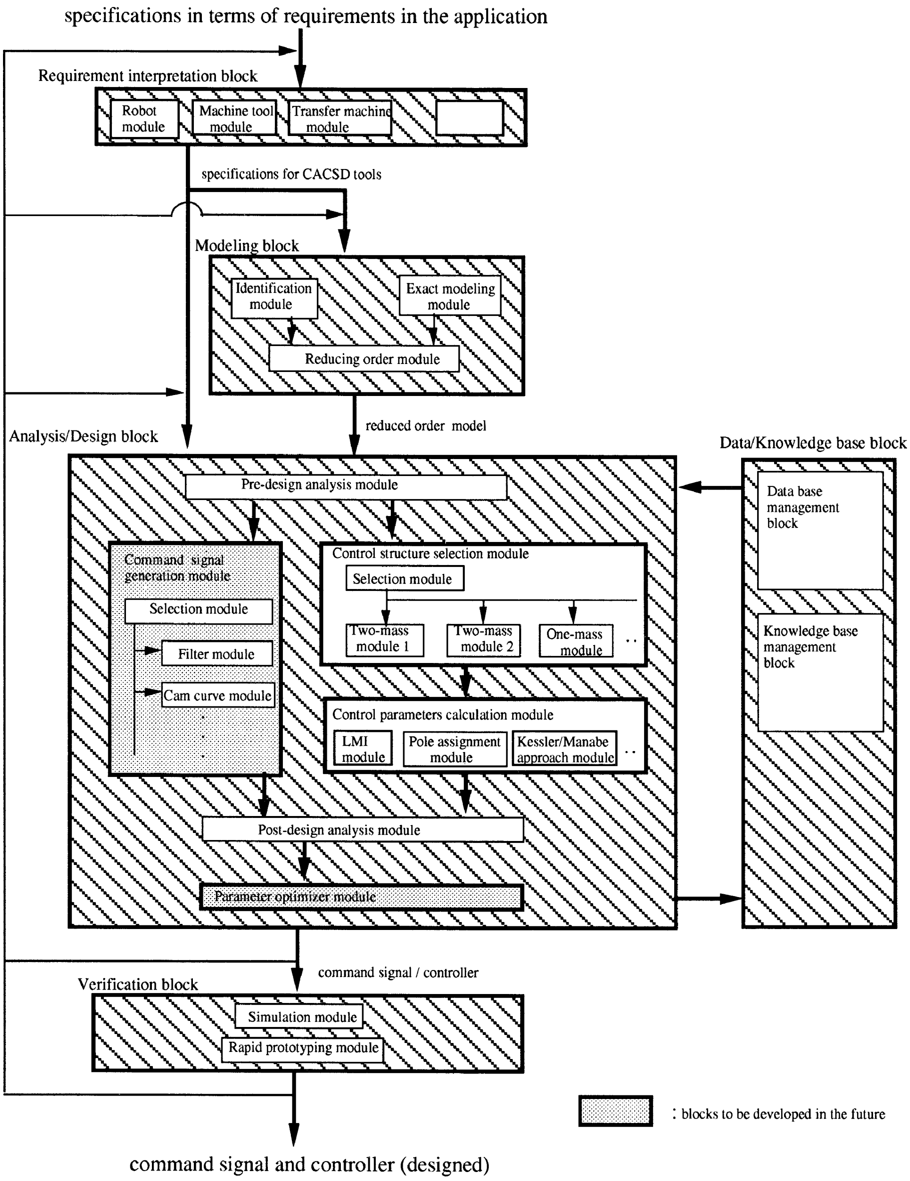


Figure 1. Configuration of the proposed CSDA system.

nal generation module would apply the appropriate module for control system design. All of the modules in the analysis/design block are constructed using MATLAB/SIMULINK.

In this block, the structure and the parameters of a controller are designed according to the interpreted requirements and the system model. In a conventional CACSD system, users need to select the controller structure, such as PID. In addition, the users need to give some of the criteria, such as cost functions, to the CACSD system. The aim of developing the CSDA system is to automate these processes by analyzing the given specifications and the model. Since it is difficult to get the best controller automatically, the system requires some redesigning process. If the results are not satisfactory, the users can modify the design parameters during and/or after this process. If the specifications given by the requirement interpretation block are satisfied, the designed command signal and designed controller are sent to the verification block. If the specifications are not satisfied, redesign demand is sent to the requirement interpretation block. More details would be given in the next section.

2.4. VERIFICATION BLOCK

In this block, simulations and/or experiments are carried out whether or not the given specifications are satisfied. In the simulation of this block, the effects of the sampling time, quantization, backlash, and friction are also taken into account. Furthermore, the original higher order model could be used for the simulation.

The simulation programs are written using MATLAB/SIMULINK M-files and DSP-CIT is used for a rapid prototyping. If the given specifications are not satisfied, redesign command would be sent to the requirement interpretation block, the modeling block, or the analysis/design block. If the specifications are satisfied, the process of designing a command signal and a controller is completed.

2.5. DATABASE MANAGEMENT AND KNOWLEDGE BASE BLOCK

2.5.1. *Database Management Module*

Many control system design packages do not provide any support for database management. Control system design activities are carried out using the package or design environment such as MATLAB. It is then up to the engineer to decide how to store the design results and document the design activity. Very often, the results are put in a data file in text format, and the file is given a meaningful file name. Very few design package has a built-in database management facility as an integral part of the CACSD environment. The database management module in the CSDA system is a built-in facility for the user. The purposes of the facility are:

1. To keep track and to organize the models, design specifications, controllers obtained and analysis results over the control system design life-cycle.
2. To automate the documentation of control system design activities by the engineer. This is to avoid manual organization of the large amount of data and files created during the course of the design process.
3. To form a database on the past design activities which can help future design attempts. It should be easy to retrieve data and information by using a query facility on the database.

The database is formed and updated after each design session. At the end of every session, the system parameters, design specifications, control structure selected, control parameters obtained, design analysis results as well as written comments from the designer is stored as a relational database, together with basic information of the design session such as identity number, date, time and name of the designer. This is a very useful facility for the automated documentation of the design session. As the current version of CSDA is concerned with linear control system design only, the problem of traceability between derivative model [8] does not appear. However, if CSDA is expanded in the future, then information on relationships between different models has to be stored by introducing reference pointers in each database item.

2.5.2. *Knowledge Base Module*

The knowledge base module in the CSDA system contains the heuristic knowledge of the control system design process. It is implemented using the control system design package called MEDAL [9]. MEDAL is a sophisticated computing environment developed at the University of Waterloo for mathematics, simulation and expert system development [10, 11]. The unique feature of MEDAL is that it provides an integrated environment for both numeric computation and knowledge-based system development.

The language syntax of MEDAL is an extension to the popular MATLAB (Matrix Laboratory) language. For examples of the MEDAL language syntax, please refer to [9–11]. MEDAL provides mathematical functions and the M-files features similar to MATLAB. However, MEDAL also has command syntax for (fuzzy) expert system development, as well as data structures like packed matrices, rules and frames. Hence, MEDAL is an integrated expert system shell for the development of knowledge-based systems which can perform sophisticated numeric calculations. Knowledge can be represented using facts, rules, frames and fuzzy logic.

In our CSDA system, relevant information about the system model and the desired specifications can be inputs to the knowledge base module. Findings from the pre-design analysis will also be used for the knowledge base module. The module in turns should provide advice to the user. The user can request for results from the inferencing of the knowledge base module. At the end of a design session, the user is asked to document his design and comments. These design experiences are also useful for the update of the knowledge base module.

As the current implementation of MEDAL is not coupled with MATLAB, the designer has to open a separate window to run the MEDAL package. Data exchange between MATLAB and MEDAL has to be carried out using external data files. Thus, MEDAL is used as an auxiliary program to help organize the knowledge base aspects of CSDA.

3. Analysis/Design Block

This block is the core of the CSDA system. Details of the control structure selection module and the control parameters calculation module are explained here. A pre-design analysis module and a post-design analysis module are mentioned briefly. A command signal generation module and a parameter optimizer module will be included in the future.

The authors have examined several control algorithms to select a satisfactory control structure and its parameters. These algorithms are written as user-defined functions in MATLAB. One special feature of the CSDA system is an "Auto design" button. This button aims to give an automated design solution to the user. This includes a selection of the control structure and a suitable design of the controller parameters. The previous results in designing a controller are also saved. Hence, the system can also choose a satisfactory control structure and decide on its parameters with reference to previous design results.

3.1. PRE-DESIGN ANALYSIS MODULE

System model parameters are displayed or modified in this module. Various diagrams for control system analysis can be displayed. Poles of the system, anti-resonant frequency, and resonant frequency of the model are also shown to help the user to decide on the control structure and controller parameters. The results from this module can also be passed to the knowledge base module, which could provide some heuristic knowledge and advice to the user.

3.2. CONTROL STRUCTURE SELECTION MODULE

A selection module, a two-mass module 1, a two-mass module 2, a one mass-module, and so on are in this group. The selection module selects a suitable module according to the reduced order system model. Then, a satisfactory control structure is used in the selected module, see Figure 3. In case the reduced order model is a two-mass model as shown in Figure 2 and the load position can be measured, the two-mass module 1 will be selected.

In the two-mass module 1, two types of state feedback controllers shown in Figures 4 and 5 are possible. The controller shown in Figure 4 is a usual state feedback controller without disturbance observer. The controller shown in

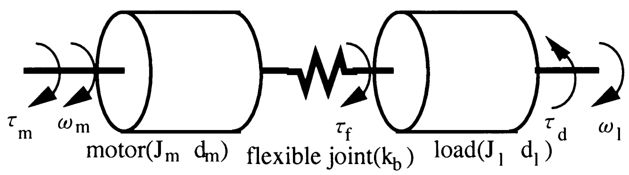


Figure 2. Two-mass model.

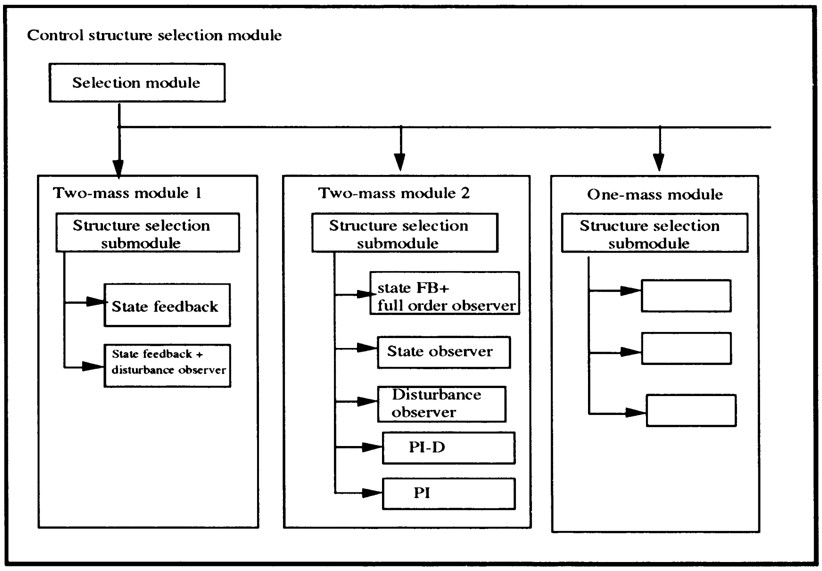


Figure 3. Control structure selection module.

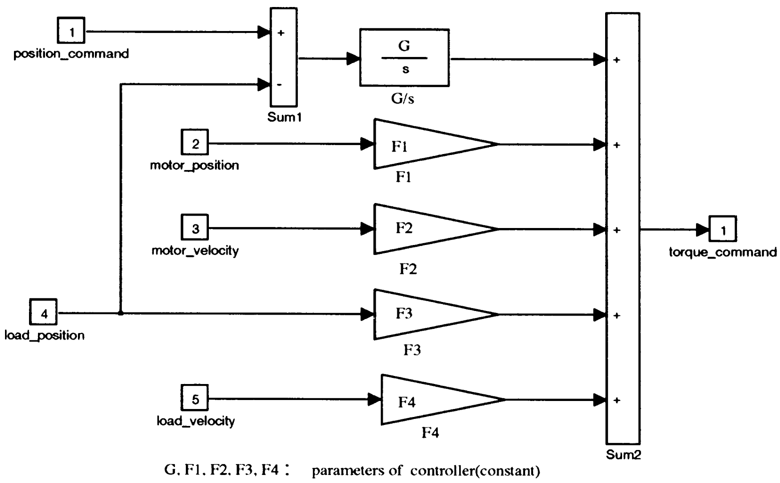


Figure 4. State feedback controller.

Figure 5 is a state feedback controller with an additional disturbance observer which contains a nominal inverse model of the load and the flexible joint. This observer is introduced to suppress the influence of parameter variation in load inertia and stiffness coefficient of the two-mass model. So in case the minimal value of J1 is smaller than a half of its nominal value, controller 2 shown in Figure 5 is selected in the auto design mode.

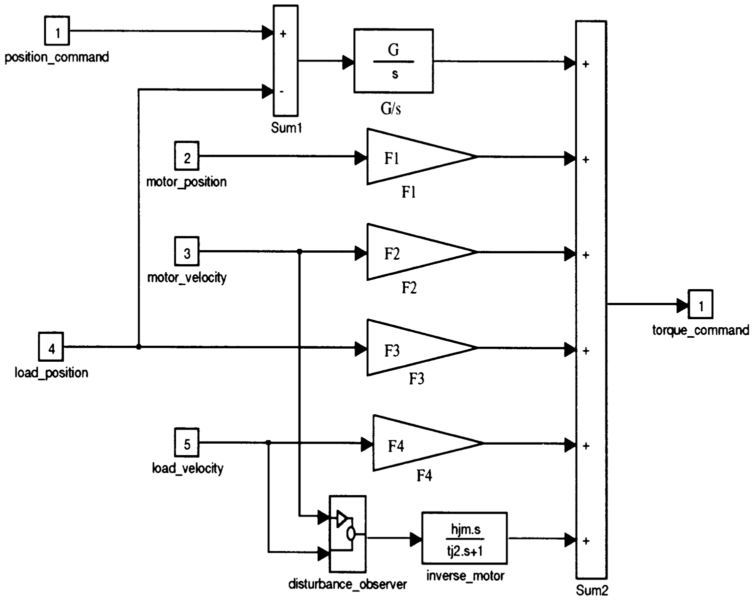
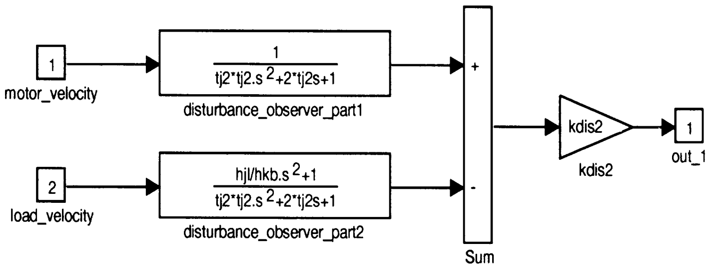


Figure 5. State feedback + disturbance observer controller.



hjl: nominal load inertia, hkb: nominal stiffness, tj2,kdis2: constant parameters

Figure 6. Disturbance observer in Figure 5.

When the load position cannot be measured, the two-mass module 2 is selected instead of the two-mass module 1. In the two-mass module 2, a state feedback controller with full order observer and four controllers with inner speed loop are prepared. The four controllers are disturbance observer controller, state observer controller, PI-D controller, and PI controller. In the two-mass module 2 a satisfactory control structure is selected according to the anti-resonant frequency of the controlled object and the required cutoff frequency. If the anti-resonant frequency is higher than the cutoff frequency, a state observer controller is selected. In another case, a disturbance observer controller is selected. Of course, the user can also select a controller by himself.

3.3. CONTROL PARAMETERS CALCULATION MODULE

Parameters of several control structures can be calculated using the same control system design technique such as pole assignment or Kessler/Manabe approach. The ways to calculate satisfactory parameters have something in common even if the structure of the controllers are different. The main modules are LMI module, pole assignment module, and the Kessler/Manabe [15, 16] approach module. The parameters of the control structure selected are calculated in theses modules.

3.3.1. *LMI Module*

Linear Matrix Inequalities (LMIs) have started to be used in control system design [5–7]. The advantage of the LMIs is that many analysis and synthesis problems can be formulated as LMIs, e.g. H-∞, H2, pole-assignment and more. In addition, the LMIs can often be solved efficiently using modern numerical algorithms. If a solution is found, it is the global optimal solution subject to the LMI-constraints. The LMI concept is appealing because it makes it possible to combine different control specifications or methods in the same framework. The designer can then find an *optimal* controller constrained by the control specifications. The optimal solution of the LMI problem is found through iterations. Computing power required is very high and is enough to solve LMIs. In the future, it will be even higher thus making it possible to solve more complex problems faster.

In the LMI module, three different specifications are combined. These specifications are, H-∞, pole-assignment, and parameter variations [7].

(A) *H-∞ LMI.* H-∞ is used to achieve a system with low sensitivity to disturbances. In other words, an H-∞ optimal controller is designed to minimize the worst error which can result from any disturbance.

(B) *Pole-Placement LMI.* Pole-placement LMI is a means to specify system performance. By assigning a pole-placement region, the time-response and damping of the system can be controlled. This enables us to take actuator limitations into account. For example the motor has only limited driving torque, and hence the size of the feedback gains has to be limited. Also, the degree of stability and time-response can be specified by moving the poles from the origin.

(C) *Plant Parameter Variations LMI.* To enhance robustness, two specifications are being used. First, the above described H-∞ objective ensures low sensitivity in the low frequency region. Secondly, parameter variations are being accounted for. The controlled object is expressed by a descriptor form in the LMI module, and LMIs are formed for every maximum/minimum value of the parameters of the reduced order model. For example, the number of LMIs becomes four when two parameters are varied.

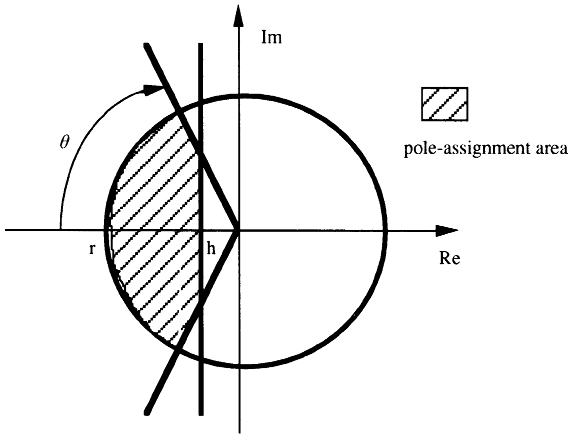


Figure 7. Pole assignment area in LMI module.

In this LMI module all poles are assigned in the area shown in Figure 7. Users set the radius r, angle θ, and distance h. The values of r and θ are fixed. If there is no solution, h is decreased and LMI is solved again.

3.3.2. *Pole Assignment Module*

The pole assignment module has four algorithms:
1. Algorithm 1: all poles are on the real axis in the left half plane.

The poles are $[-\alpha_0, -\alpha_0, -\alpha_0, -\alpha_0, \ldots]$

2. Algorithm 2: all poles are on the real axis in the left half plane.

The poles are $[-\alpha, -\alpha, \ldots, -2\alpha, -2\alpha, \ldots]$

3. Algorithm 3: all poles are placed according to the Butterworth pattern. In case the Butterworth pattern is used, the characteristic polynomials are as follows.

$$
\begin{aligned}
s &+ \alpha \\
s^2 &+ 1.4\alpha s + \alpha^2 \\
s^3 &+ 2.0\alpha s^2 + 2.0\alpha^2 s + \alpha^3 \\
s^4 &+ 2.6\alpha s^3 + 3.4\alpha^2 s^2 + 2.6\alpha^3 s + \alpha^4.
\end{aligned}
$$

4. Algorithm 4: all poles are placed according to the ITAE (Integral of Time multiplied by Absolute Error) minimum pattern. In case the ITAE minimum pattern is used, the characteristic polynomials are as follows.

$$
\begin{aligned}
s &+ \alpha \\
s^2 &+ 1.4\alpha s + \alpha^2 \\
s^3 &+ 1.75\alpha s^2 + 2.15\alpha^2 s + \alpha^3 \\
s^4 &+ 2.1\alpha s^3 + 3.4\alpha^2 s^2 + 2.7\alpha^3 s + \alpha^4.
\end{aligned}
$$

A state feedback controller and a full order observer can assign its poles at will. Algorithm 1, algorithm 3, or algorithm 4 are used to assign the poles of such controllers. For the nominal plant, the state feedback controller designed by algorithm 1 has no overshoot, but the rise time is long. The state feedback controller designed by algorithm 3 has large overshoot, and the response is oscillative. The controller designed by algorithm 4 has a short rise time, but it has a small overshoot. Algorithm 2 has a similar performance as algorithm 1. The robustness to the parameter variation is the highest in the case of algorithm 2. According to the features mentioned above, algorithm 2 is used in the Auto Design mode.

3.3.3. *Kessler/Manabe Approach Module*

The Kessler/Manabe approach is an algebraic method, whose design philosophy is based on the relative size of the closed-loop characteristic polynomial coefficients. It was first proposed by Dr. Claus Kessler at Siemens in 1960 [15] and then refined by Blaschke in 1974 [17]. A standard form of the design parameters was suggested and the method has been used with great success for DC and AC machine control. Important studies on the stability and robustness issues was carried out by Lipatov [12] in 1978. Sufficient condition for stability and instability of linear continuous systems were derived and these are expressed in the form of inequalities based on the adjacent coefficients of the characteristic polynomial.

The method was applied successfully for the attitude control of satellites at Mitsubishi Electric in Japan. Dr. Manabe has further developed the method by introducing a graph and called it the Coefficient Diagram Method (CDM) [16]. He has also carried out extensive study on the design parameters and modified the standard form suggested by Kessler. It has been used for steel mill control [18], air heater control and regulation of an urban heating network differential pressure [19]. Hence, the Kessler/Manabe [20] design approach has been known and applied with great success in many industry applications, although the method is not generally well-known in the academic circle.

The design parameters in this approach are a set of stability indices γ_i and an equivalent time constant τ. $P(s)$ is a characteristic polynomial, and the stability indices γ_i and the equivalent time constant τ are defined as follows:

$$P(s) = a_n s^n + \cdots + a_1 s + a_0, \tag{1}$$

$$\gamma_i = \frac{a_i^2}{a_{i+1} a_{i-1}} \quad (i = 1, 2, \ldots, n-1), \tag{2}$$

$$\tau = \frac{a_1}{a_0}. \tag{3}$$

The default values for γ_i are $[2.5,\ 2,\ 2, \ldots]$ and those values are used in the Auto Design mode. Of course, an expert user can set the values by himself.

3.4. POST-DESIGN ANALYSIS MODULE

The post-design analysis module serves to give a preliminary assessment of the control design. This will give an idea on the suitability of the control structure selected and the control parameters obtained. More detailed analysis has to be carried out in the Verification Block with a more detailed system model. Also, fine tuning of the controller parameters could be done to further enhance the control design in the parameter optimizer module.

In this module, nine items related to the performance and robustness of the system are examined. They are rise time, overshoot, maximum magnitude of control signal, bandwidth, maximum drop when the step disturbance is added, maximum gain from measurement noise to load angle, gain margin, phase margin, and maximum tracking error when the velocity command is trapezoid. Five figures can also be shown in this module. Those figures are step response of the load position, torque command when a step input and a step disturbance are added, tracking error when the velocity command is a trapezoid, open loop bode diagram, and closed loop bode diagram.

Whether the specifications are met or not can be examined by these nine items and five figures. If the specifications are met, the result of the design and the comments on the design are saved into the database, which will help future design attempt. If the design result is not satisfactory, the user will return to the previous module and design a controller by using another controller structure or other controller parameters.

4. Graphical User-Interface and Examples

4.1. GUI

In order to make CSDA more user-friendly, an Graphical User-Interface (GUI) is developed. Figures 8 and 9 give an example of the GUI in the analysis/design block. Figure 8 shows the interface at the beginning of the block. Figure 9 shows the control structure selection module group. A user can go to the next or previous step by clicking the relevant button. Static help files can be seen by clicking the Info/Help button. The information is very useful for the user to select a control structure and a control parameter calculation algorithm. When the Auto Design button is clicked, control structure and control parameters are automatically decided according to the experience of the authors. The display button is used to show a block diagram of the control structure.

4.2. DESIGN EXAMPLES

In this section, a design example is given based on the use of the CSDA system. Figures 10 and 11 are simulation results of a controller designed by CSDA. The controlled object is a two-mass model shown in Figure 2, and a step input and

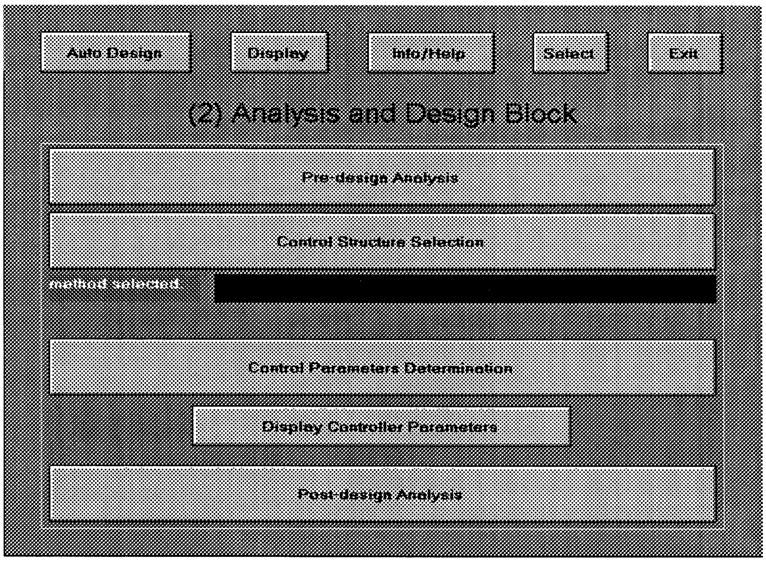


Figure 8. Example of GUI (1).

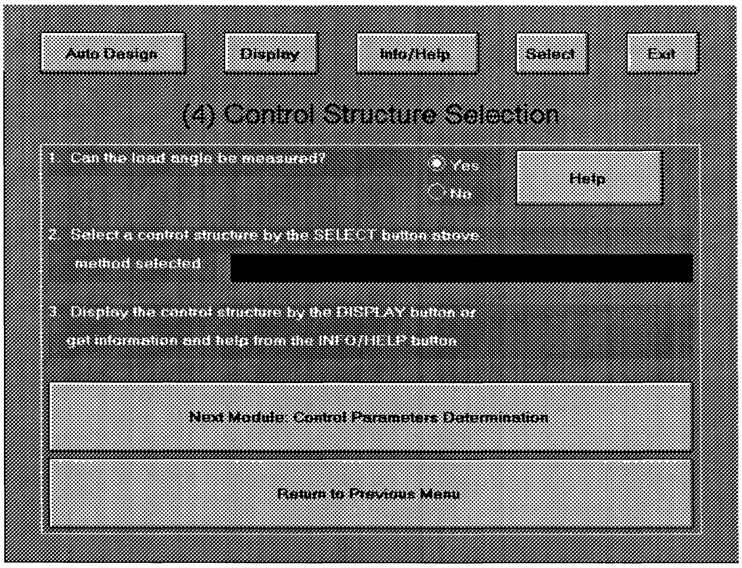


Figure 9. Example of GUI (2).

a step disturbance is added to the controlled system. Simulations based on both a nominal model and a real model are carried out.

Parameters of the nominal model:

$$\text{jm} = 0.0045\ [\text{kgm}^2], \quad \text{jl} = 0.0045\ [\text{kgm}^2], \quad \text{kb} = 2.05\ [\text{Nm/rad}],$$
$$\text{dm} = 0.0105\ [\text{Nms/rad}], \quad \text{dl} = 0.0057\ [\text{Nms/rad}].$$

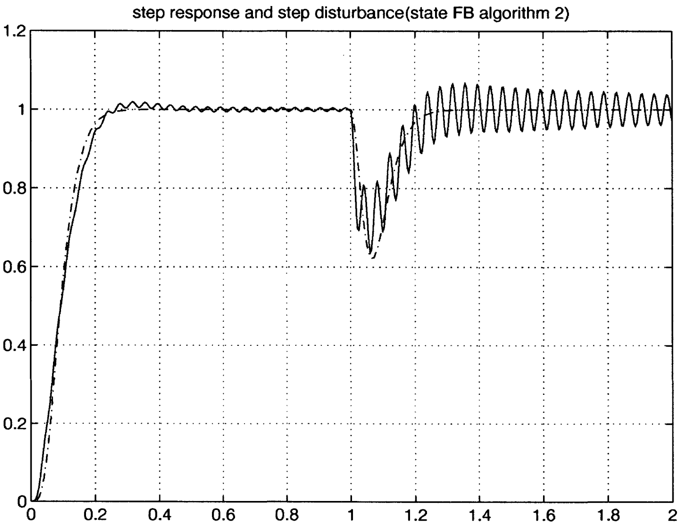


Figure 10. Simulation results (state feedback controller).

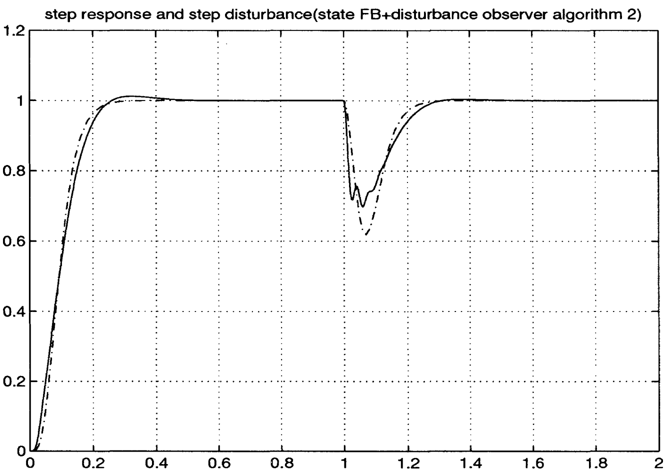


Figure 11. Simulation results (state feedback + disturbance observer).

Parameters of the real model:

$$\text{jm} = 0.0045\ [\text{kgm}^2],\quad \text{jl} = 0.001\ [\text{kgm}^2],\quad \text{kb} = 2.05\ [\text{Nm/rad}],$$
$$\text{dm} = 0.0105\ [\text{Nms/rad}],\quad \text{dl} = 0.0057\ [\text{Nms/rad}].$$

The dotted lines are the results of the nominal system, and the solid lines are the results of the real system. Figure 10 is the results when CSDA is used without Auto Design. A state feedback controller is used in this case. Figure 11 is the result when the Auto Design mode is used. A state feedback plus disturbance observer controller is selected in this case.

When the controlled object is a nominal model, the state feedback controller has a similar performance as the state feedback controller with a disturbance observer. The state feedback controller is, however, inferior to the state feedback controller with a disturbance observer in the case that there is a large parameter variation. The CSDA system has many control structures and many algorithms to decide on the controller parameters. The CSDA system also has a knowledge base/data base management block. With the help of the CSDA, a user can easily obtain a controller whose performance is similar to the controller designed by an expert control engineer.

5. Future Developments

In the future development of CSDA, a parameter optimizer module and a command signal generation group will be included.

5.1. PARAMETER OPTIMIZER MODULE

In the Kessler/Manabe approach, the stability condition by Lipatov [12] gives the mathematical basis for this approach. Several theorems on sufficient conditions for stability have been derived. A standard form on the choice of the stability index has been proposed by Manabe. Although the standard form has been found very useful and usually results in a good design, it is believed that fine tuning is always needed to get an optimal design. Also, because of different design requirements and applications, the standard form should only be used as an initial design. Hence, the controller design problem can be transformed into a search for an optimal set of stability index and equivalent time constant. Hence, a search technique can be used to find a set of stability indices and equivalent time constant which can meet the specifications and minimize an objective function. This turns the design problem into a multi-objective optimization problem [13, 14]. The simplex algorithm in the MATLAB Optimization toolbox has been used for this kind of search.

5.2. COMMAND SIGNAL GENERATION MODULE

This is a module to design a command signal. It is important to design an optimal command signal which can satisfy the requirements in the application. The knowledge on designing a command signal is stored in the database/knowledge base block, and the users can select a satisfactory command signal and its parameters according to the authors experiences or previous design results.

6. Conclusions

A control system design environment called CSDA (Control System Design Automation) system has been developed. The CSDA system is very comprehensive and is consisted of a requirement interpretation block, a modeling block, an analysis/design block, a data base/knowledge base block, and a verification block. Regarding the current status of the package, the major portion of it has been completed. The main structure of the package has been developed as described in the paper. There are many different modules in the various blocks of the system, and they are being completed. The system is being developed in the MATLAB/SIMULINK environment. In this paper, the analysis/design block has been described in detail. The main contributions of the system are that:

(a) Powerful control system tools and methodologies are made available to more engineers. CSDA can help to obtain satisfactory control system design even by an inexperienced engineer.
(b) The LMI design module was introduced for control parameter calculations. This module helps to satisfy multiple design specifications.
(c) The Kessler/Manabe design approach is also available in CSDA. This method has been found very successful in many industrial applications.

The CSDA system also has a user-friendly interface developed using GUI in MATLAB. The effectiveness of the CSDA system for controller design has been demonstrated with a two-mass model.

Acknowledgment

The authors would like to express their thanks to Mr. Tobias Ericsson in the Royal Institute of Technology for his contribution in making up the LMI module.

References

1. Reilly, J., Levine, W. S, Eadan, E., and Huang, C.: A computer-aided optimization-based controller design tool, in: *Proc. Am. Control Conf.*, 1991, pp. 990–995.
2. Murakami, K., Knnno, T., Kurotani, K., and Takano, M.: Optimization system for control system design, *IEEE Tokyo Section Denshi Tokyo* **33** (1994), 124–127.
3. Gahinet, P., Nemirowski, A., Laub, A. J., and Chilali, M.: *LMI Control Toolbox*, The Math Works Inc., 1994.
4. Hori, Y.: Novel methods and recent trends of 2-inertia resonant system control, in: *ROBO-MEC '94*, 1994 (in Japanese), pp. 1278–1283.
5. Boyd, S. et al.: *Linear Matrix Inequalities in System and Control Theory*, SIAM, 1994.
6. Chilali, M. and Gahinet, P.: H-∞ design with pole placement constraints: an LMI approach, *Trans. Automatic Control* **41**(3) (1996), 358–367.
7. Sugie, T. and Hamamoto, K.: Controller design of two mass-spring system via LMI, *Systems Control and Information* **9**(5) (1996), 219–226 (in Japanese).
8. Taylor, J. H.: Database management for computer-aided control engr., in: M. Jamshidi and C. J. Herget (eds), *Recent Advances in Computer-Aided Control Systems Engineering*, Elsevier Science, 1992, pp. 127–150.
9. Pang, G. K. H.: MEDAL: Matrix & expert system development aid language, in: *Proc. IEEE Symp. on Computer-Aided Control System Design*, Napa, California, USA, 1992, pp. 148–155.
10. Pang, G. K. H.: Knowledge-based control system design, in: M. Jamshidi and C. J. Herget (eds), *Recent Advances in Computer-Aided Control Systems Engineering*, Elsevier Science, 1992, pp. 127–150.

11. Pang, G. K. H.: A knowledge environment for an interactive control system design package, *Automatica* **28**(3) (1992), 473–491.
12. Lipatov, A. V. and Sokolov, N. I.: Some sufficient conditions for stability and instability of continuous linear stationary systems, *Automat. Remote Control* **39** (1979), 1285–1291 (translated from *Automatika i Telemekhanika* **9** (1978), 30–37).
13. Ng, W. Y.: Perspectives on search-based computer-aided control system design, *Control System Magazine* **13**(2) (1993), 65–72.
14. Boyd, S. P. and Barratt, C. H.: *Linear Controller Design: Limits of Performance*, Prentice-Hall, 1991.
15. Kessler, C.: Article on theory of multiple loop control (translated from Ein Beitrag zur Theorie mehrschleifiger Regelungen), *Regelungstechnik* **8**(8) (1960), 261–266.
16. Manabe, S.: Coefficient diagram method as applied to the attitude control of controlled-biasmomentum satellite, in: *Proc. 13th IFAC Symposium on Automatic Control in Aerospace*, Palo Alto, California, Sept. 12–16, 1994, pp. 322–327.
17. Blaschke, F.: Das Verfahren der Feldorientierung zur Regelung der Drehfeldmaschine, Dr.-Ing. Dissertation, Technische Universität Braunschweig, 1974.
18. Hori, Y.: Two-mass system control based on resonance ratio control and Manabe polynomials, in: *Proc. First Asian Control Conf.*, Vol. 3, 1994, pp. 741–744.
19. Voda, A. A. and Landau, I. D.: A method for the auto-calibration of PID controllers, *Automatica* **31**(1) (1995), 41–53.
20. Manabe, S.: Coefficient diagram method: a polynomial design approach, to be submitted to *Journal of Guidance, Control, and Dynamics*.
21. Herbert, H.: Automotive control: from concept to experiment to product, in: *Proc 9th IEEE Int. Symp. on Computer-Aided Control System Design*, 1996, pp. 129–134.
22. Taylor, J. H. and Seres, P.: An intelligent front end for control system implementation, in: *Proc 9th IEEE Int. Symp. on Computer-Aided Control System Design*, 1996, pp. 7–13.
23. Whidborne, J. F., Postlethwaite I., and Gu, D.-W.: A mixed optimization approach to multiobjective computer-aided control system design, in: *Proc 9th IEEE Int. Symp. on Computer-Aided Control System Design*, 1996, pp. 309–314.
24. Sienel, W., Bunte, T., and Ackermann, J.: PARADISE – Parametric robust analysis and design interactive software environment: A Matlab-based robust control toolbox, in: *Proc 9th IEEE Int. Symp. on Computer-Aided Control System Design*, 1996, pp. 380–385.
25. Chipperfield, A. J. and Fleming, P. J.: PARSIM: A paralel optimization tool, *IEEE Control Systems* **15**(2) (1995), 48–53.