Available online at www.sciencedirect.com
SCIENCE DIRECT
Control Engineering Practice 14 (2006) 177–184
CONTROL ENGINEERING PRACTICE
www.elsevier.com/locate/conengprac

Interactive control system design

J.P. Keller*

Institute of Automation, University of Applied Science, Solothurn, Riggenbachstr. 15, 4600 Olten, Switzerland

Received 29 April 2004; accepted 7 December 2004
Available online 2 March 2005

Abstract

This contribution describes a LabVIEW-based interactive computer-aided control system design tool. Although it does not provide the flexibility of common low-level design tools, it offers a set of comfortable, ready-to-use solutions for plant identification, loop-shaping or LQR/LQG-controller design and controller implementation. The tool integrates into industrial plant control and enables engineers to efficiently do the control system design from plant identification to controller implementation on the same platform. The user interface with convenient system editors and analysis tools facilitates the user to interactively attain the design goals. If desired, a configurable wizard leads the user through the design process.

Keywords: Computer-aided control system design; Interactive programs; User interfaces; Identification; LQG control; Implementation

1. Introduction

In control engineering, there is still a wide gap between what a student has to learn during his studies and what he actually uses in the subsequent industrial work. There are two main reasons for this discrepancy. First, adequate equipment is often absent in an industrial environment. Even basic approaches, like PID-tuning with step response are not widely used because tools to measure a step response are not at hand. So there is no reason to expect that more sophisticated methods like, loop shaping-controller design will be used. Only if the control problem is not solvable with a PID-controller tuned by trial and error, a controller design project is started causing considerable development costs. Most likely, the control engineer is faced with a heterogeneous environment. Signal generation and data acquisition equipment have to be installed to get the necessary data for plant modelling. The following steps, i.e. parameter estimation and controller design, are carried out on a PC or a Workstation. Controller implementation and final testing are done on the industrial plant control system. Usually, no workflow is available and experiment design relies solely on the expertise of the engineer. With increasing capabilities of control systems, including the industrial communication networks, it is possible to integrate controller design tools into the capabilities of industrial control systems.

The second reason for the methodological gap between education and daily industrial routine is that many computer-aided controller design systems (CACSD-systems) offer a large amount of basic functionality, well suited for application in research, but do not provide ready-to-use solutions for the most common controller design problems. Additionally a considerable amount of training is necessary to learn mathematical formalisms and syntax of CACSD-systems. There are no easy to use and tested solutions for the complete design process including all necessary steps from problem analysis to controller implementation, including methods for successful plant start-up, plant operation as well as handling emergency situations.

The best premise is an engineer who enjoys doing a systematic controller design. This might be achievable with a controller design tool with the following properties.

*Tel.: +41 62 388 25 58; fax: +41 62 388 25 93.
E-mail address: juerg.keller@fhso.ch.

0967-0661/$ - see front matter © 2005 Elsevier Ltd. All rights reserved.
doi:10.1016/j.conengprac.2004.12.014

The tool hides mathematical formalism as much as possible. It is interactive with respect to parameter variations and design path. If controller or plant parameters are changed, the results are immediately visible on the selected analysis displays. Preferably, system parameters are modified in their native representation. This means that plant parameter variation is done by changing values of physical parameters in an algebraic plant model, whereas a controller transfer function is modified by changing pole or zero locations. Whenever reasonable, parameter values can be changed in graphical editors.

A second aspect of interactivity is interactive wizard support. After completion of a design step, there might be different path to follow and the user has to choose the most suitable. Sensible design paths are modelled in a state machine. Its animated graph is displayed to the user. So, at every decision point, the engineer is aware of the consequences for the following design steps. In order to simplify interaction with the tool, the user interface is object oriented. Consequently, each similar system type, whether it is a plant, a controller or a closed loop, has the same appearance, menu bars and analysis methods.

The tool offers a set of ready-to-use solutions of typical controller design problems. This increases motivation to seek a more sophisticated controller design because there are no tedious control system calculations and there is no CACSD-syntax to learn. Obviously, only problems within the provided set of solutions can be easily solved.

The tool presented in this paper supports the engineer in the complete controller design process. It consists of the following modules:

- System modelling including parameter estimation;
- Loop shaping controller design;
- State-feedback with observer either with LQ or pole placement;
- Support for loop transfer recovery;
- Controller implementation either for PID-type (Aström & Wittenmark, 1997) or state-space controllers.

The tool is based on LabVIEW, National Instruments, which is particularly well suited for this purpose. It can run on different platforms, e.g., on a notebook or an industrial PC with real-time operating system, and is one of the most powerful tools to create user interfaces. A large library for signal processing and mathematical functions can be combined with a complete set of industrial data acquisition and signal generation hardware.

Ideas for modern interactive loop shaping were proposed by Johansson, Gäfvert, and Aström (1998). A useful MATLAB-based tool for control design education is available from the book of Aström and Wittenmark (1997). Typical design problems can be interactively explored. In MATLAB's controller design toolbox an interactive tool for SISO controller design is available (MATLAB, 2002). In Kottmann, Qiu, and Schaufelberger (2000), the work on an object-oriented CACSD-tool at the ETH-Zürich is summarised. The proposed tool provides some interactivity in basic editors. A very effective feature is the action tree, providing the capabilities of workflow-based scenarios for controller design. An application of the tool to a mechanical system is described in Qiu, Schaufelberger, Wang, Keller, and Sun (1999). An ambitious project for web-based control education is the Dynamit project (Löhl et al., 1999). A virtual control lab is provided on an elaborate web interface. Based on SYSQUAKE, interactive user interfaces for several controller design problems where proposed by Dormido (2003) and Tan, Atherton, and Dormido (2003).

In the following, the main ideas of the interactive, computer-aided control system design tool (i-CACSD-tool) are presented. In the first section, general concepts are explained. The next sections describe the main modules, i.e. plant modelling and identification, controller design and finally controller implementation. In the last section, application of the tool is shown for system identification and LQR/LQG controller design.

2. The interactive CACSD-tool

2.1. General properties

The modular structure of the i-CACSD-Tool represents the 3 major steps in control system design: plant modelling and parameter identification, controller design and controller implementation. In order to make the tool easy to use, the GUI-entities are standardised. In Fig. 1, the most frequent objects are shown.

Most elements in a block diagram are of the class 'system'. They have their data editors, the same tools for analysis and standardised methods to load and save the data. System data consists of the model data, e.g. transfer function coefficients, and names of inputs, outputs and states. Naming the model variables allows

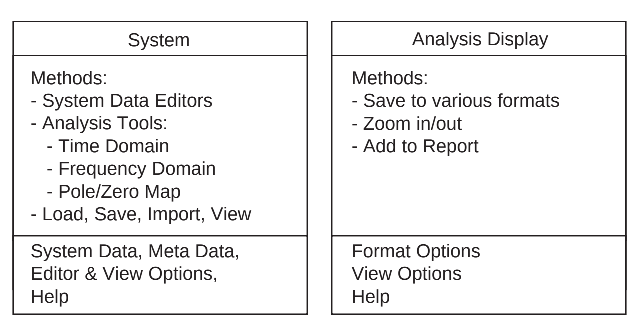


Fig. 1. System and analysis display class.

an easy selection of signals in the analysis tools. The methods for the class Analysis Display are shown in Fig. 1 on the right. Analysis results can be saved to various formats or added to a report. Each design can be easily documented with a report tool into an html-document. Additional plots and comments can be manually appended to a standard report.

Supplementary design information, henceforth called metadata, can also be managed. It contains data-like user name, date and time, name of source data and user specified text. This is very convenient for identifying data in the sequel, even years later. Metadata can be previewed in the tool's file dialog without importing the data.

2.2. Plant modelling and identification

The nonavailability of plant models is one of the main reasons that model-based controller design methods are not broadly used. Since sophisticated physical models are not easy to derive and require a considerable amount of time and money, simpler methods have to be available. Methods for frequency response analysis are well known (Ljung, 1987) and can easily be used on industrial plants if the dominant time constants are reasonably short, preferably shorter than 1 min. In this tool the plant is excited with sine or multi-sine excitation signals, a simple solution to guarantee good signal-to-noise ratio at the investigated frequencies. Since controller design based on a nonparametric frequency response model is limited with respect to time domain analysis and controller design methods, i.e. to loop shaping, it is reasonable to approximate the measured frequency response with a plant model. This might be a black box transfer function or a physical model. There results a two-step approach: first the frequency response is measured and in a second step approximated by a plant model. Obviously, this approach is limited to SIMO-Systems. The system is preferably stable, although it can also be applied to systems with integrators.

2.2.1. Frequency response identification

For frequency response identification several experimental settings must be determined. A wizard supports the user in specifying appropriate signal levels, experimental frequencies and optimal sampling time. Wizards usually offer a sequential navigation with back and next buttons. The wizard is not transparent in the sense that the user cannot see what the next or previews step will do. A transparent representation of a wizard sequence is the sequential function chart (SFC). The representation is similar to the decision tree representation proposed in Qiu et al. (1999). A SFC of a simplified identification wizard is shown in Fig. 2. With a glance at the wizard diagram, the user is aware of what she is currently doing—the highlighted wizard step—and what she will be doing next. From Fig. 2 it can be depicted that identification starts with experiments to get the appropriate signal levels. Next, the dominant time constant is determined to attain knowledge about dominant poles. This allows the wizard to propose a frequency pattern for first experiments. The frequency response is measured using either single frequency scans or a periodic, multi-sine signal. Excitation with sine signals is chosen to guarantee good signal to noise ratio at the investigated frequencies. For multi-sine excitation frequency pattern and phase shift are optimised to get an excitation signal with minimal peak value and maximal amplitude for each sine component within the admissible signal range. As can be seen in the wizard state chart (Fig. 2), estimation can be aborted to specify new signal levels, left to model fitting or improving the frequency response measurements by refining the frequency pattern. Based on the first results, a new frequency pattern is determined so that phase changes between two measured frequencies are small. This is ideal for several reasons. Technically, phase unwrap can be done unambiguously.

From the plant identification point of view, it follows the suggestive hint (Ljung, 1987) that inputs should be chosen in order to sensitise the output with respect to parameter changes. Dominant plant poles and zeros always lead to obvious phase changes. If plant phase is well measured, i.e. if there are no large phase

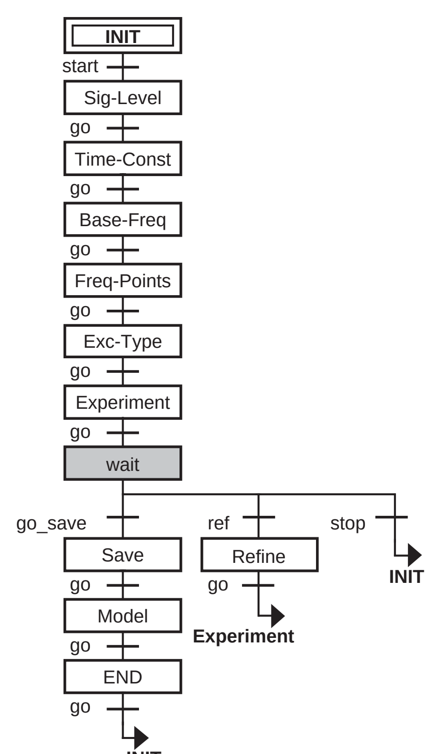


Fig. 2. Wizard SFC.

changes between two measured frequency points, plant magnitude is also well defined at the pole and zero locations. Fig. 3 shows how a new frequency pattern is proposed for the identification of a simulated PT2-type system. The vertical dark lines are at the new frequencies. All frequencies are integer multiples of the base frequency shown on the left to ensure a perfectly periodic signal. This is necessary to avoid spectral leakage.

2.2.2. Plant modelling

An algebraic plant model can be formulated either as a transfer function or a state space model. Each element is an algebraic term consisting of known and unknown physical parameters. The unknown parameters can be obtained by manually fitting the frequency response of the plant to the measured frequency response. Although manual approximation may not be very scientific, it has several appealing advantages over numerical identification methods. First of all, it uses the human skills to weight data and interpret outliers. When optimising physical parameters, the engineer gets a tight feeling of the frequency response sensitivity with respect to parameter changes. This is particularly useful if the plant has to be modified in order to accomplish some requirements. In addition, an inadequate model structure becomes immediately evident and it is the responsibility of the engineer that parameter values remain within a physically sensible range. Furthermore, it does not require any knowledge about identification methods. The resulting model can be used for state space controller design methods and for loop shaping.

The i-CACSD tool also provides the possibility to fit a black box transfer function. An initial transfer function has to be specified by the number of integrators, the relative degree and a guess of the order. Poles and zeros are dragged to their optimal locations. On a scenario basis, poles and zeros can be provisionally added. Both the system with and without the singularity are displayed, enabling the user to accept or discard the changes. Furthermore, DO and UNDO are available. The model has to be saved for the next step, the controller design.

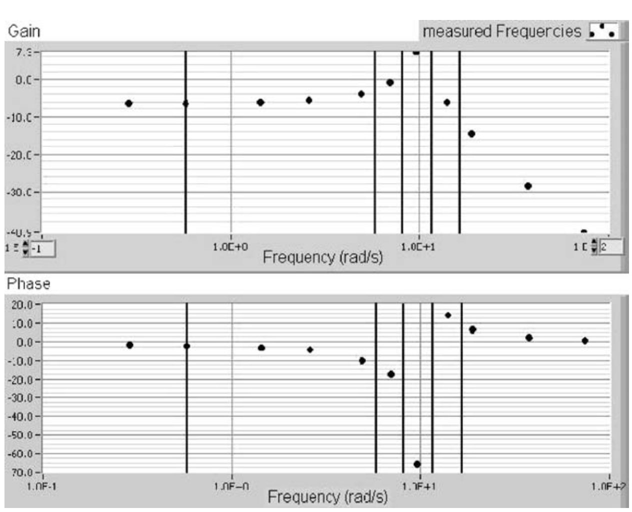


Fig. 3. Improving the identification.

The wizard SFC can be modified using a convenient editor. This editor allows the user either to specify his own workflow for some design task or to use the SFCs for plant control, see Keller (2001).

2.3. Controller design

At present the controller design tool consists of two design modules. The first is simple loop shaping design and the second is state feedback with observer, with either LQR/LQG or pole placement. The loop-shaping tool is similar to the MATLAB tool and only major differences are outlined in Section 2.3.1. The state-space controller design is described in Section 2.4.

2.3.1. Simple loop-shaping controller design

Loop shaping controller design can be done for the system shown in Fig. 4. Many design problems can be formalised into this simple structure. The loop-shaping design tool is similar to MATLAB's 'SISO controller design tool'. In both tools, the idea is to vary controller parameters to get the desired open-loop frequency response. This can be done either by editing the controller parameters directly within the open-loop frequency response plot or by editing the frequency response of the controller transfer function while the open-loop frequency response is immediately adapted to the controller changes. In the MATLAB toolbox the first approach was chosen. In the proposed controller design tool, the second approach is favoured. Experience shows that it does not make sense to freely shape the open loop without monitoring the properties of the resulting controller. It easily results in controllers with nonoptimal and unrealistic lead-elements leading to nonacceptable stress on the actuators due to high controller gain at high frequencies.

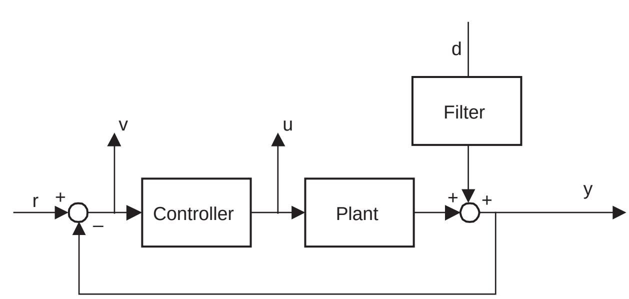


Fig. 4. SISO control system.

2.3.2. State feedback with observer

As pointed out in Johansson et al. (1998), there is also a need for an interactive controller design tool for state space methods. Controller design consists of the following steps (Geering, 2001): Determine a state feedback gain so that closed-loop requirements are satisfied, then design an observer and modify the observer until loop transfer recovery is satisfactory. State feedback and observer design can be done either with linear quadratic (LQ) methods or with pole placement. In many situations integral action is also required for tracking control. The resulting control system structure is as proposed in Pierre (1994) and is shown in Fig. 5.

The complexity of the controller design requires the user interface to be well structured. The control system in Fig. 5 is divided into subsystems. Each system is realised with a similar object consisting of analysis and data handling methods and data. This is shown in Fig. 6. In a top down view: at the Control System-level the user can analyse closed-loop properties like time or frequency response, or open-loop frequency response including different measures for loop transfer recovery. The 'true' plant can be handled and analysed in the true plant object. Properties of the controller can be investigated in the controller object. To be able to design the controller, the object has three subsystems, i.e. the design plant, the state feedback controller and the observer. In the state feedback object, a state feedback controller can be designed and the system is analysed with state feedback only. When using pole placement, the poles can be graphically moved to the desired location on the complex plane. Real poles remain on the real axis until two real poles are moved to the same location. A change of colour indicates that the poles can be moved onto the complex plain. The reverse action is also possible. The state feedback gain is calculated with a very robust method proposed by Roppenecker (1990).

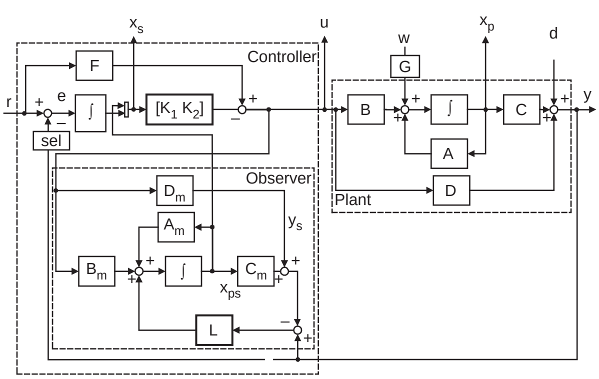


Fig. 5. State feedback with integral action and observer.

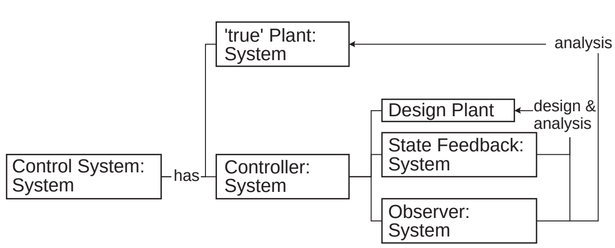


Fig. 6. Object structure.

For LQR design weighting matrices for states, inputs and optionally cross-terms have to be specified. To set the values of the state weighting, the following options are available: diagonal, $C^{T}VC$ with C being the measurement matrix for y or no special structure. Furthermore, an additional matrix C_{opt} can be defined resulting in a state weighting $C_{opt}^{T}VC_{opt}$. With the matrix C_{opt} one can think of additional outputs, $y_{opt} = C_{opt}x$ which is subjected to optimisation. The observer can be designed in a similar way.

State feedback and observer design are based on a design plant. The resulting design can be analysed with both the design plant and the 'true' plant. With the 'true' plant uncertainties or changes in plant parameters or even in plant model can be simulated. This enables a user to examine design robustness.

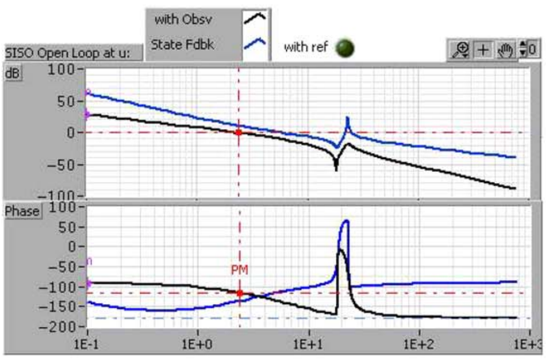
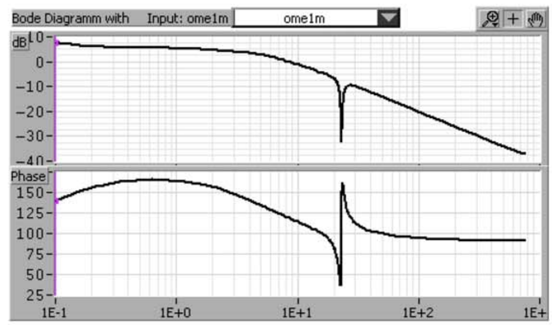
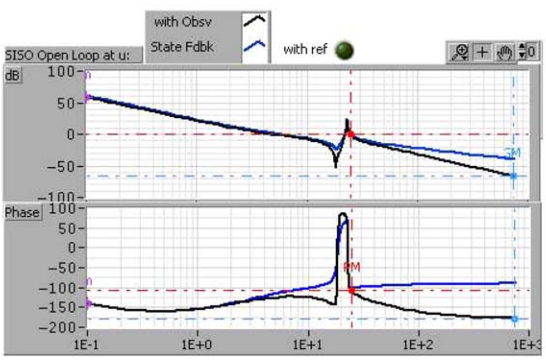
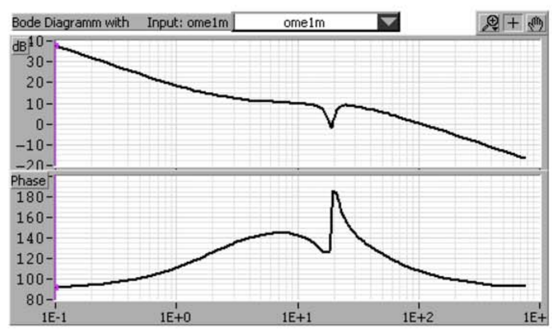
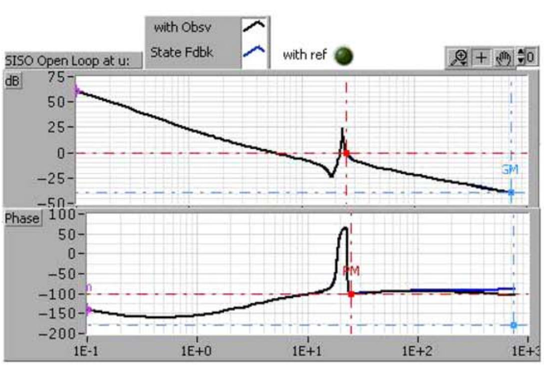
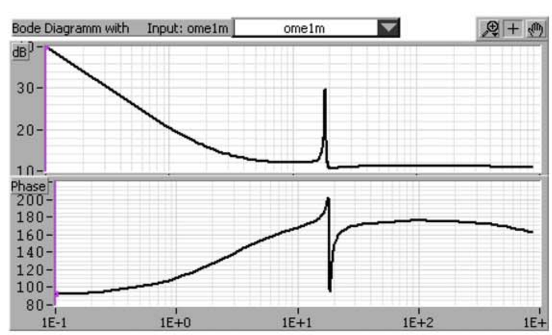
Since loop transfer recovery is a property of the open-loop control system, analysis tools are available in the control system object. Observer parameters can be changed according to Doyle and Stein (1981) to recover the open-loop frequency response. Frequency responses of the open-loop with and without observer can be immediately compared on a plot. Examples are shown in Table 1. Without additional programming effort, the controller frequency response and the control signal's response to measurement noise can be monitored during loop transfer recovery. This may uncover the consequences of a state feedback design with unrealistically large bandwidth because in most practical applications, it is not realistic to let a LQ-controller increase plant phase by more then 90° unless disturbances are minimal. Properties of the resulting controller can be analysed in the controller object.

Again, the tool is interactive and parameter changes lead immediately to a recalculation of the observer properties resulting in a true interactive design.

2.4. Controller implementation

After successful controller design, the controller has to be discretised and implemented. Simple lead–lag controllers are discretised and implemented as PID-controllers as proposed in Aström and Wittenmark (1997). The resulting controller can be tested on the same system. State feedback controllers are discretised and implemented in modal form. Integrators are equipped with anti-windup strategies and actuator

Table 1
Interactive LTR

State Feedback Open loop without and with observer	controller frequency response
no loop transfer recovery:	
	
optimal loop transfer recovery:	
	
exaggerated loop transfer recovery:	
	

saturation is taken into consideration (Aström & Wittenmark, 1997). Effects of controller discretisation, signal quantisation and saturation can be investigated by means of simulation. If simulation results are satisfactory, the controller can be tested with the real plant. Real-time charts allow comparison measured and predicted data, providing useful information about observer design and quality of the plant model. Since system identification, controller test and controller operation are done with the industrial plant, suitable safety measures have to be integrated. A very convenient solution is shown in Fig. 7. Plant measurements and control signal generations are done with a configurable IO-device. An FPGA-based device allows data acquisition for simple analog inputs with filtering, encoder readings with interpolation or even resolver signals. Furthermore, security logic can be implemented on the FPGA-device for protecting the plant. This can be, for

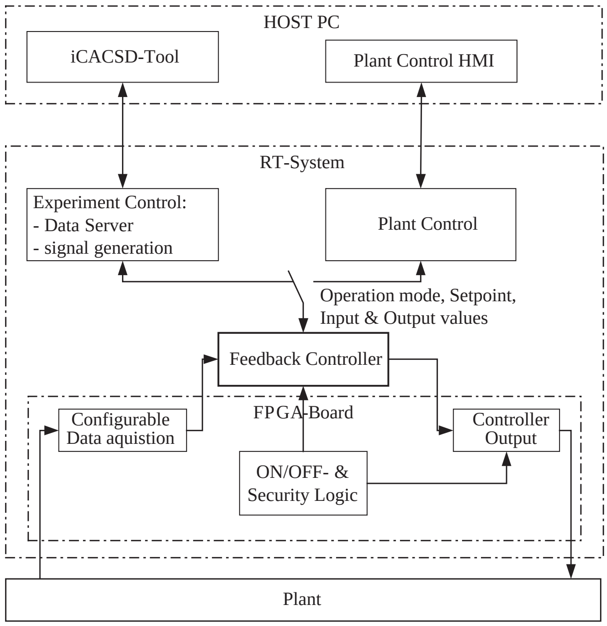


Fig. 7. Integration of iCACSD into industrial plant control.

example, simple limit switches combined with model-based overload protection for motors. The control law is implemented in Feedback-Controller module. For plant identification and controller design, this module is under control of the iCACSD Tool. For plant operation the module obeys the plant control software. The structure in Fig. 7 shows how the iCACSD-Tool is integrated into industrial plant control.

3. Hardware environment

The i-CACSD-Tool can run on different industrial plattforms, i.e. on industrial PC, Fieldpoint modules or PXI-Systems. National Instruments offers a large range of products for industrial automation. The Fieldpoint modules with a real-time operating system are a PLC-like systems, which is suitable for distributed process control. The PXI-Systems allow the implementation of real-time feedback control at high sampling rates. All systems have the capabilities to run the real-time part of the i-CACSD-Tool in addition to the plant control tasks. There are no barriers like those in common PLC-systems that prevent an engineer to integrate more sophisticated controllers into plant control software.

4. Example

Plant identification and controller design using the iCACSD-tool are shown for a small rotating system with 2 rotation masses as shown in Fig. 8.

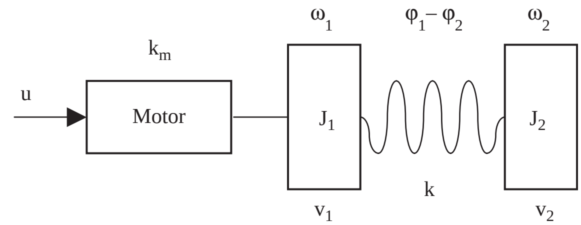


Fig. 8. Plant model.

The plant frequency response was identified with single sine excitation. Plant input was motor amplifier input; plant output was angular velocity of the first mass J1. The plant was modelled with algebraic state space model. Inertias were calculated, whereas the other model parameters, i.e. motor gain k_m, spring constant k, viscous frictions v_1 and v_2 were varied to get a good fit of the plant frequency response. The result is shown in Fig. 9.

Based on this model and LQR/LQG-controller was designed. In the sequel, the benefits of interactive controller design in a high-level design tool will be shown for the very important loop-transfer recovery aspect. Important analysis results do not have to be created by programming a script. They are available in the menu bars, tested and ready to use. In Table 1 different solutions are compared. In the first row an observer with minimal LTR was designed. Obviously, loop gain and crossing over frequency are drastically reduced. The controller frequency response shows a controller with integral action with strong low pass filtering and a narrow notch-filter at the systems resonance frequency. If the recovery gain (Doyle & Stein, 1981) is increased, there results a good LTR as shown in row 2 of Table 1. The controller has a sensible frequency response, with a robust notch filter. If LTR is exaggerated, as in row 3, perfect LTR can be achieved, but the controller frequency response shows that the controller cancels the plant zeros in a nonrobust way. Minor changes of the plant zeros lead to sharp resonance peaks in the closed-loop frequency response becoming apparent as undesirable ripple in the time domain. With the interactive design tool, the recovery gain can be altered, while the reaction of important system properties can be monitored.

5. Conclusions

This contribution presented an interactive controller design tool suitable for control engineering practice. It offers the opportunity to focus on control system design and minimises the effort required to master short-time valued syntax of CACSD-systems. The underlying object-oriented interface structure combined with

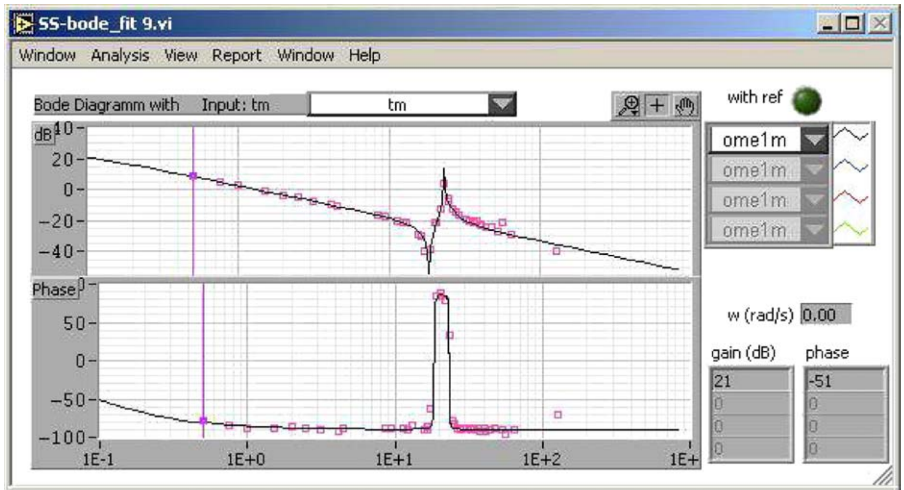


Fig. 9. Frequency response identification.

wizard support simplifies user interaction. Graphical editors, immediate update of analysis panels to parameter changes, automatic report generation and no tedious control system calculations motivate engineers to an increased commitment to systematic controller design. The tool is based on LabVIEW and makes use of the wide variety of available industrial process interfaces provided by National Instruments.

References

Aström, K. J., & Wittenmark, B. (1997). *Computer controlled systems theory and design* (3rd ed). Upper Saddle River, NJ: Prentice-Hall.

Dormido, S. (2003). The role of interactivity in control learning. *Proceedings of the sixth IFAC symposium on advances in control education* (pp.11–22).

Doyle, J., & Stein, G. (1981). Multivariable feedback design: concepts for a classical/modern synthesis. *IEEE Transactions on Automatic Control*, *AC-26*, 4–16.

Geering, H. P. (2001). *Robuste Regelung*. ETH-Zürich: IMRT-Press.

Johansson, M., Gäfvert, M., & Aström, K. J. (1998). Interactive tools for education in automatic control. *IEEE Control Systems*, *18*(3), 33–40.

Keller, J. P. (2001). Programming sequential function charts in LabVIEW. *Proceedings of the NIWeek 2001*, Austin.

Kottmann, M., Qiu, X., & Schaufelberger, W. (2000). *Simulation and computer aided control systems design using objec-orientation*. Zürich: vdf Hochschulverlag.

Ljung, L. (1987). *System identification, theory for the user*. Upper Saddle River, NJ: Prentice-Hall.

Löhl, T., Pegel, S., Klatt, K.-U., Engell, S., Schmid, C., & Ali, A. (1999). DYNAMIT—Internet based education using CACSD. *Proceedings of the of the 1999 IEEE International Symposium on CACSD*, 273–278.

MATLAB (2002). The Mathworks. *Inc. MATLAB—Control system toolbox*. Reference Manual.

Pierre, D. A. (1994). Discrete-time enhanced linear quadratic control systems. In, *Control and dynamic systems*, vol. 66. Academic Press.

Qiu, X., Schaufelberger, W., Wang, J., Keller, J.P., & Sun, Y. (1999). Object-oriented analysis and design of 03CACSD using OMT. *Proceedings of IFAC World Congress, Bejing*.

Roppenecker, G. (1990). *Zeitbereichsentwurf linearer Regelungen: Grundlegende Strukturen und eine Allgemeine Methodik ihrer Parametrierung*. Oldenbourg, München, ISBN:3-486-21640-6.

Tan, N., Atherton, D. P., & Dormido, S. (2003). Systems with viable parameters; classical control extensions for undergraduates. *Proceedings of the sixth IFAC symposium on advances in control education* (pp. 287, 292).