INTERACTIVE CONTROL SYSTEM DESIGN

J.P. Keller

Institute of Automation, University of Applied Science, Solothurn,
Bittertenstr. 15, 4702 Oensingen, Switzerland
juerg.keller@fhso.ch

Abstract: This contribution describes a LabVIEW based interactive computer aided control system design tool. It offers a set of ready to use solutions of typical control system design problems. Instead of a syntax a menu driven user interface with convenient system editors and analysis tools enables the user to interactively attain the design goals. The tool allows the student to efficiently do the control system design from plant identification to controller implementation on the same platform.

Keywords: Computer-aided control system design; Interactive programs; User interfaces; Educational aids; 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 for example loop shaping controller design, will be used. Only if the control problem is not solvable with a PID controller 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 has 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, for example, 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 control education should be focused more on engineering abilities than on research skills. Considerable time is lost on teaching mathematical formalisms and syntax of CACSD-systems. Consequently there is not enough time left to acquire expertise and routine in the complete design process including all necessary steps from problem analysis to controller implementation. Methods for successful plant start up, plant operation as well as handling emergency situations should also be mandatory topics in control engineering education. The student must primarily acquire expertise in selecting and applying the appropriate methods. This can be achieved by providing the student with opportunities to solve several design problems. Evidently, available course time is always too short but the program can be focused on the topics mentioned above by means of suitable tools.

The tool presented in this paper supports the student 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 et al.,1997) or state-space controllers.

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, she 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 is based on LabVIEW, National Instruments, which is particularly well suited for this purpose. It can run on different platforms, for example 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 (1998). A useful MATLAB based tool for Control Design education is available for the book of Astrom et al. (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 (2000), the work on an object-oriented CACSD-tool at the ETH-Zürich is summarised. The proposed tool provide 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 et al. (1999). An ambitious project for web-based control education is the Dynamit project (Löhl, 1999). A virtual control lab is provided on an elaborate web interface.

In this paper, the main ideas of the interactive, computer aided control system design tool (i-CACSD-tool) are presented. In the first chapter general properties are explained. The next chapters describe the main modules, i.e. plant modelling and identification, controller design and finally controller implementation. Practical experience and student feedback are summarised in the last chapter.

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 Figure 1, the most frequent objects are shown.

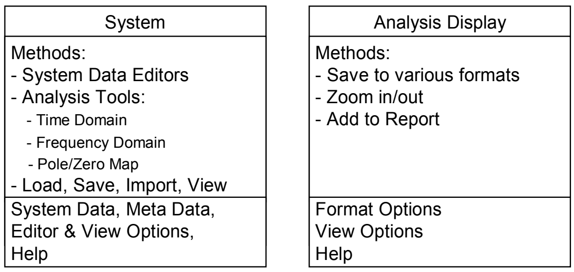

Fig. 1. System and analysis display class

Most elements in a block diagram are of the class 'system'. They have their appropriate editors, the same tools for analysis and standardised methods to load and save the data. System data consists of the model data, for example transfer function coefficients, and names of inputs, outputs and states. This allows an easy selection of signals in the analysis tools. The methods for the class 'Analysis Display' are shown in Figure 1 on the right. The results can be saved to various formats or added to a report. Each design can be easily documented with a report tool into a html-document. Additional plots and comments can be manually appended to a standard report.

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

2.2 Plant modelling and identification

The availability 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 minute. Since controller design based on a non parametric 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. The physical model can be either a transfer function model or a state space model. In the i-CACSD tool's editors, a user can specify the model with algebraic expressions in terms of physically meaningful parameters. Unknown parameters can be manually optimised in order to achieved a good fit of the plant's frequency response. A numeric optimisation of the parameters values could easily be added, but the process of manually adjusting the unknown physical parameter has several appealing properties. When optimising the physical parameter, the engineer gets a tight feeling of the frequency response sensitivity with respect to parameter changes, an inadequate model structure becomes immediately evident and it is in the responsibility of the engineer, that parameter values will remain within a physically sensible range. Furthermore, no knowledge on system identification methods is required. 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 SISO-Systems. The system is preferably stable, although it can also be applied to systems with integrators.

For frequency response identification several experimental settings must 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 user can not see, what the next or previews step will do. A suitable representation of a wizard sequence is the sequential function chart (SFC). The represenation is similar to the decision tree representation proposed in Qiu et al (1999). The SFC of an identification wizard is shown in Figure 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 Figure 2 it can be depicted

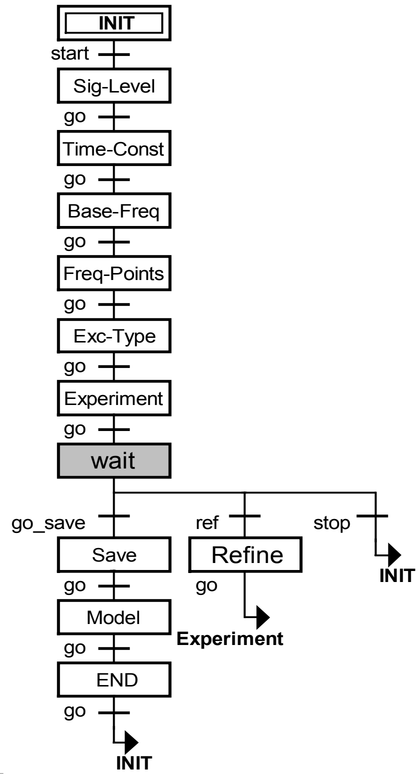

Fig. 2. Wizard state chart

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 are 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 (Figure 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 changes between two measured frequency points, plant magnitude is also well defined at the pole and zero locations. Figure 3 shows how a new frequency pattern is proposed for the identification of a 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.

The plant model is obtained by manually fitting the frequency response of the plant to the measured frequency response. The plant model can be either a transfer function or an algebraic state space model. Although manual approximation may not be very scientific, it has several advantages over numerical identification methods. First of all it does not require any knowledge about identification methods. Furthermore, it uses the human skills to weight data and interpret outliers. In order to fit a 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. Matching a physical model can be quite cumbersome, but as a result, physical parameters lie in a sensible range and the user has gained a lot of insight about theirs effects on the frequency response. This is particularly useful if the plant has to be modified in order to achieve some requirements or if the physical model is wrong or too simple to fit the frequency response. The model has to saved for the next step, the controller design.

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

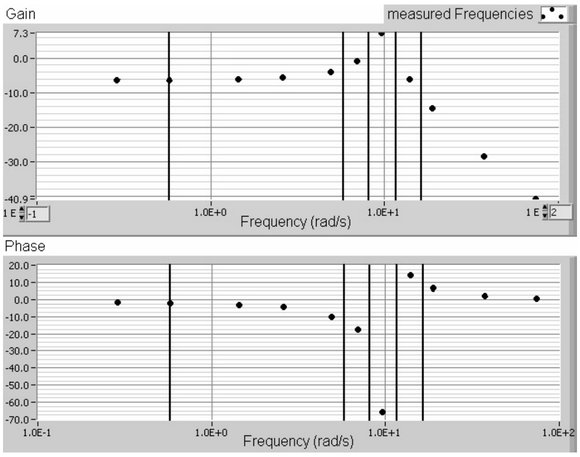

Fig. 3. Improving the identification

2.3 Controller Design
Presently 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. For the loop-shaping tool the only major differences to the MATLAB tool are outlined in the next paragraph. With more details, the state-feedback tool is described in the following paragraph.

2.4 Simple Loop Shaping Controller Design
Loop shaping controller design can be done for the system shown in Figure 4. Many design problem 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. This easily results in controllers with non optimal and unrealistic lead-elements leading to non acceptable stress on the actuators due to high controller gain at high frequencies.

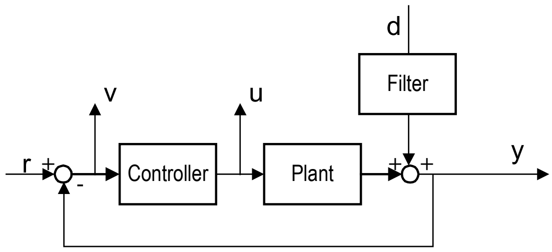

Fig. 4. SISO control system

2.5 State Feedback With Observer
As pointed out in Johansson (1998), there is also a need for an interactive controller design tool for state space methods. Controller design consists of the following steps (Geering): 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 Figure 5.

The complexity of the controller design requires the user interface to be well structured. The control system in Figure 5 is divided into subsystems. Each system is realised with a similar object consisting of methods and data. This is shown in Figure 6. This allows to user to first analyse the plant. This can be

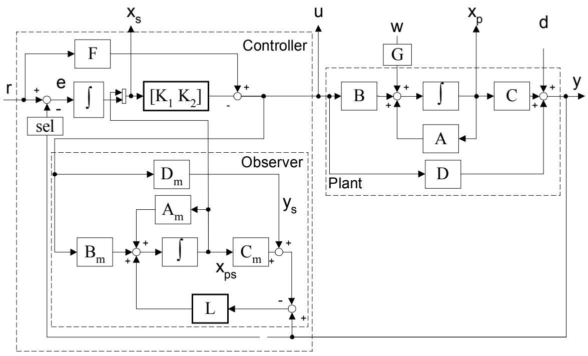

Fig. 5. State feedback with integral action and observer

done with the available system analysis tools. In the following a state feedback controller can be designed. Properties of the system under state feedback can be interactively explored. When using pole placement, the poles can be graphically moved to the desired location on the complex plane. In order to reposition complex pole pairs, the position of the pole with positive imaginary part has to be changed. 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. For pole-placement, a very robust method proposed by Roppenecker is used. For systems with more than one plant input, the additional degrees of freedom appear in parameter vectors (Roppenecker 1990). At present, no parameter optimisation is realised but they can be manually modified.

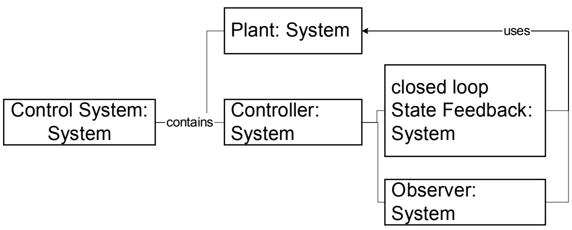

Fig. 6. Object structure

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^TVC 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 $C_{opt}{}^TVC_{opt}$. With the matrix C_{opt} one can think of an additional outputs, $y_{opt} = C_{opt}\ x$ which is subject to optimisation. This could also be represented with a 2-port model of the plant with outputs y and y_{opt}. Experience shows that y_{opt} often has to be modified during state feedback design. It is therfore reasonable to specify C_{opt} in the LQR-design.

The observer can be designed in a similar way. Dual to C_{opt} a matrix G (see Figure 5) can be defined for LQE design. Again, parameter changes lead immediately to a recalculation of the observer properties resulting in a true interactive design. At present, the state space model for the observer is the same as the plant model. For sensitivity analysis in plant parameters, it might be interesting to have two different models. This will be implemented in future versions.

Since loop transfer recovery is a property of the control system, analysis tools are available in the control system object. At present, the open-loop transfer function opened at the control value u can be analysed. Observer parameters can be changed according to Doyle 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. An example is shown in Figure 7. 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 degrees unless disturbances are minimal. Properties of the resulting controller can be analysed in the controller object.

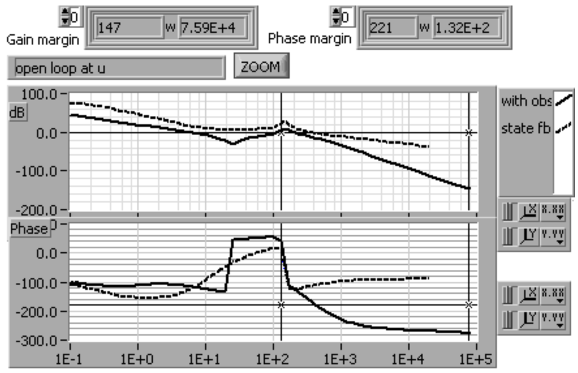

Fig. 7. Loop transfer recovery plot

Robustness can be explored if the plant editor is opened again. Physical plant parameter or pole/zero locations can be varied within a reasonable range. The resulting closed-loop properties can be monitored while changing the parameters.

2.6 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 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 saturation is taken into consideration (Aström 1997). When testing the controller with the true plant, predicted observer states and outputs are drawn in a chart in order to verify the observer design.

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 is a PLC-like system, which is suitable for distributed process control. The PXI-Systems allow the implementation 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 in common PLC-systems, that prevent an engineer to integrate more sophisticated controllers into plant control software.

4. Experience and Student Feedback

The CACSD tool has been used in the advanced control system course since 3 years. It is used as demonstration tool in control lectures, in exercises and in laboratory courses. As a demonstration tool, it is very suitable to demonstrate system properties and design rules. The effect of parameter changes on the plant's frequency response, the consequences of a small gain margin or the effect of changing LQ-weightings can be demonstrated in an impressive manner. When used in exercises, the student's work is focussed on controller design problems and can easily explore the phenomenon of controller design. In the laboratory course, a complete controller design from plant modelling to controller implementation and test is performed. The i-CACSD-tool had a large impact on the course concept. Its reason can be imagined from a colleague's comment: 'with this tool every idiot can design a controller'. The exercises had to be redesigned in such away that not only a documented controller design had to be done, but additional problems had to be solved. A challenging problem is to let the student find general design principles. A typical example is controller design for plants with a resonance peak close to crossing over frequency. Faced with this problem, it requires special skills to derive general design principles. With the CACSD-tool, the student can easily verify his ideas. Also a comparison of loop shaping with state feedback controller design can be achieved in a reasonable amount of time.

The students' acceptance was reflected in the tool selection. For loop shaping controller design about half of the class used MATLAB. The other half used the LabVIEW tool and added a valuable contribution by testing the tool. For state feedback controller design, most of the students used the LabVIEW tool. In the laboratory the students ran the tool on their notebooks and used the lab-monitor as second display. This indicates a shortcoming of all the design tools: either you have several plots as small as a stamp or your screen is too small.

5. Conclusions

This contribution presented an interactive controller design tool suitable for control engineering education. It offers the opportunity to focus education on learning important aspects of 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 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 students to an increased commitment to control education. The tool is based on LabVIEW and makes use of the wide variety of available industrial process interfaces provided by National Instruments.

6. References

Aström K.J. and Wittenmark B. (1997). Computer Controlled Systems Theory and Design. 3rd ed, Prentice Hall, Upper Saddle River, New Jersey.

Doyle J. and Stein. G. (1981). Multivariable Feedback Design: Concepts for a Classical/Modern Synthesis. IEEE Trans. on Automat. Contr., AC-**26**, 4-16.

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

Johansson M., Gäfvert, M. and 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. Proc. NIWeek 2001, Austin.

Kottmann, M., Qiu X. and Schaufelberger W. (2000). Simulation and Computer Aided Control Systems Design using Objec-Orientation. vdf Hochschulverlag, Zürich.

Ljung L. (1987). System Identification, Theory for the user. Prentice Hall. ISBN 0-13-881640-9

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

Pierre, D.A. (1994). Discrete-Time Enhanced Linear Quadratic Control Systems In*: Control and Dynamic Systems*, 66, Academic Press.

Qiu X., Schaufelberger W., Wang J., Keller J.P. and 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.

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