DOCUMENT RESUME

ED 089 707 IR 000 419

AUTHOR Volz, Richard A.; And Others
TITLE Computer Instructional Aids for Undergraduate Control Education.
INSTITUTION Michigan Univ., Ann Arbor. Dept. of Electrical and Computer Engineering.
SPONS AGENCY National Science Foundation, Washington, D.C.
PUB DATE 74
NOTE 6p.

EDRS PRICE MF-$0.75 HC-$1.50 PLUS POSTAGE
DESCRIPTORS *Computer Assisted Instruction; *Computer Graphics; Computer Programs; *Design; Engineering; *Engineering Education; *Engineering Graphics; Higher Education; Interaction; Man Machine Systems; Program Descriptions; Simulation; Time Sharing; Undergraduate Study
IDENTIFIERS Automatic Control; COINGRAD; *Computer Aided Design; Control Oriented Interactive Graphic Anal Design; TACL; Time Shared Automatic Control Laboratory

ABSTRACT

Engineering is coming to rely mcre and more heavily upon the computer for computations, analyses, and graphic displays which aid the design process. A general purpose simulation system, the Time-shared Automatic Control Laboratory (TACL), and a set of computer-aided design programs, Control Oriented Interactive Graphic Analysis and Design (COINGRAD), are being used to introduce undergraduate engineering students to the power and utility of computer-aided tools for automatic control. TACL and COINGRAD simulate time responses for physical systems, provide standard analysis and design calculators, and introduce students to optimization in design. They are simple and user-oriented, have interactive and graphic display capabilities, and employ a natural mathematical language. Since these programs perform the basic calculations required for classical control system analysis, they free the student from the repeated enumeration of system behavior and allow him to concentrate upon the underlying principles involved. (PB)

# COMPUTER INSTRUCTIONAL AIDS

# FOR

# UNDERGRADUATE CONTROL EDUCATION[*]

BY

Richard A. Volz[+]

Jorge Aravena[++]

Michael Dever[+++]

Theodore Johnson[**]

Darryl Conliffe[***]

## Introduction

During the past two decades engineering has come to rely heavily on the computer for all manner of tasks ranging from scientific computation to on line process control and data management. Emerging from this is a changing view of engineering design. More and more the engineer is coming to rely on the computer to provide computations and analyses which aid in the design process. We are entering an era of computer aided design. One of the major keys to the continued growth of CAD will be the rapid graphic interaction between the engineer and the computer. The idea that an engineer can enter a set of data for the system he is studying and immediately see the results of the computation displayed, not in awkward tabular arrays, but in easy to understand graphical form will be of utmost importance.

With the important role that computer tools will play in the future of engineering, it is extremely important that universities make their students aware of the potential of these tools and prepare them for their use. The objective of the systems developed is to make this possible for the level of material covered in undergraduate courses in automatic control. They provide first of all, simulation of time responses for a wide variety of physical systems. Secondly, they provide the standard analysis and design computations used in classical control theory; frequency response and root locus. Thirdly, a limited capability for system optimization is incorporated to provide the student an introduction to the use of optimization in design.

What requirements should be imposed on the program developed? What should their operating environment be? How should they appear to the user? The answer

[*] This work was supported by the Department of Electrical and Computer Engineering, University of Michigan and the National Science Foundation under grant No. GY-8313

[+] Systems Engineering Laboratory, Department of Electrical and Computer Engineering, University of Michigan, Ann Arbor, Mi.

[++] Departmento de Electricitad, Universitad de Chile, Santiago de Chile

[+++] Education Resources, Department of Dental School, University of Michigan Ann Arbor, Mi.

[**] Power Systems Laboratory, Department of Electrical and Computer Engineering, University of Michigan, Ann Arbor, Mi.

[***] Proctor Gamble Company, Cincinnati, Ohio

to these questions are even more important to the student's view of computer aided design than the range of topics covered.

First and foremost, whatever else is done, the programs must be *friendly to the user!* That is the input/output and interaction between the user and the computer must be as simple and self-explanatory as possible. Whenever possible data should be in forms as close to the natural expressions used as possible. Failure to observe this can result in even very useful programs lying around virtually unused. This consideration is particularly important in an educational environment where a student's objective is not to learn how to code data for CAD programs, but to be able to utilize them almost immediately to enhance his understanding of the material at hand. The typical student has enough to do trying to learn basic control principles without having to learn complicated data coding schemes. Secondly, to achieve maximum effectiveness, the programs should have a graphic capability and operate in an interactive mode with the user.

There have been a number of computer-aided design programs developed for control systems. However, none of them fully meet these requirements. The work described here is two fold. First, a general purpose simulation system, TACL (Time-shared Automatic Control Laboratory), was developed on a hybrid computation system. Secondly, a set of digital computer aided design programs, COINGRAD (Control Oriented Interactive Graphic Analysis and Design), were developed for use with computer graphics terminals. These are intended both to provide useful CAD function and be user oriented.

## TACL Configuration

The TACL simulation system is comprised of a general configuration control system programmed on the analog portion of AD/4 analog hybrid computer and a digital executive program, programmed on a PDP/9 digital computer operating concurrently. A variety of specific systems is obtained by selectively setting to zero or identity unneeded portions of the system. The system uses the high speed analog capabilities of the hybrid computer to solve the dynamic equations being studied and the logical capabilities of the digital computer to configure the analog for the particular system.

Communication of student users with the system is through the terminal units which are interfaced to the computers. All information flowing from the terminal units to the hybrid system flows first through the AD/4 and then through sense lines and A/D conversion channels to the PDP/9. The most important hardware features of this configuration to TACL (other than the terminals and computers themselves) are the half million word disk of the PDP/9 which stores the arrays of problem data, the 32 analog digital conversion channels and the 120 digital coefficient units, which effectively act as 120 high speed digital to analog converters. It is the use of these latter units as switches as well as coefficients which allows problems to be rapidly reconfigured to satisfy the needs of individual terminal users.

Each problem studied within the TACL system must be capable of being generated within the context of the general system. Student users of TACL, however, need not be concerned with how the selection of a specific problem is made. The general configuration is completely transparent to the user. The system will respond as if it consisted only of the specific problem which he is studying. Problems are determined by arrays of coefficient values, switch positions and pointers to parameters to be controlled from the terminals. The terminal system operates on a set of predefined problems, whose data arrays have been stored on the system disk. Essentially, operation of the system proceeds as follows:

1. Read the terminal logic word and decode the problem number and function request.
2. Read the terminal parameter values.
3. If necessary read the problem description from the disk (requires only an average of 50 ms).
4. Set all 120 coefficients, thus configuring the problem as desired (Due to the high speed Digital Coefficient Units of the AD/4 this only takes a few milliseconds).
5. Run the desired solution.
6. Release the requesting terminal.
7. Interrogate the next terminal.

In addition the system has been given an alphanumeric text capability. With each set of problem data are several pages of text explaining the problem briefly and identifying the problem parameters and how they are to be entered. To provide a permanent record of results obtained, a hard copy unit is available which can be used by any of the terminals.

Finally, to aid the instructor in problem preparation a problem generation program has been written, which operates interactively with the instructor through a teletype terminal. The instructor is able to enter problem descriptions in the mathematical terms he would normally use, e.g., coefficients of transfer fuctions, values for slope, saturation level, or deadzone in nonlinear devices. The problem generator will assist in determining scale factors and take care of all the necessary book work on coordinating scale factors. It will also correlate problem oriented information with the specific device addresses in the computer, relieving the instructor of some of the major burdens of problem preparation.

## General COINGRAD Features

COINGRAD is a set of three computer programs developed for the analysis and design of control systems. They will efficiently perform root locus, frequency response, time response and certain optimization computations for linear systems. In addition, certain nonlinearities may be included in the time response calculations. Though intended primarily for use in an interactive environment on a graphics terminal such as a Computek, they may be used on a standard teletype terminal or in batch mode.

The programs are all interactive in nature and basically operate by executing a sequence of commands issued to them by the user. The commands are all kept simple in nature such as RUN (a solution) or PLOT (the result) to make them easy for the user to remember (or look up in a short list). Once the user has indicated the basic function he wishes to perform, the program will interact with him and ask for whatever additional data may be needed, e.g. what is to be plotted. For example, to obtain a root locus plot, only three basic commands are needed:

```
G(s) = .....
RUN
PLOT
```

The program will ask for any remaining data that is needed.

An important feature of COINGRAD is that the manner in which the system is described is simple and kept as close to the natural mathematical language as possible. The programs assume a block-oriented system structure. The user will specify the characteristics of individual blocks of his system and an interconnection scheme between these, from which the program will perform the desired calculations. Each linear block is described by a transfer fuction of the form

$$G(s) = \frac{\sum_{i=1}^{m} \prod_{j=1}^{r_j} Q_{ij}(s)}{\sum_{i=1}^{m} \prod_{j=1}^{k_i} P_{ij}(s)} \qquad (1)$$

where $P_{ij}(s)$ and $Q_{ij}(s)$ are polynomials in the Laplace operator s. That is, the blocks are expressed as the ratio of sums of products of polynomials. Each polynomial must be presented in order of decreasing powers of s. To enter a transfer fuction one merely types it in the form one would normally write it, except that as raised superscripts are not allowed in the input character set, powers of s are indicated by integers following the operator s. Coefficients of s may be integers, real numbers, or single character variables to be manipulated later. The function is entered in free format, i.e., blanks are ignored. For example, to enter $G1(s) = \frac{K(s+a)}{(s+b)}$ and $G2(s) = \frac{(s+3)}{(s^2+6.25s+25)\ (s^2+100s+75)}$, one would merely type

```
G1(s) = (k)(s+A)/(s+B)
G2(s) = (s+3)/(s2+6.25s+25)(s2+100s+75).
```

The programs accept lines of this form as commands to enter the indicated transfer functions into system description arrays. As the line is processed the structure of the transfer function is identified and stored, coefficients are entered into the appropriate arrays, and variable references are set up. If a transfer function is too long to be entered on a single line, it may be continued on succeeding lines by ending unfinished lines with a colon (:) and continuing succeeding lines after the first column.

Following the definition of the linear blocks, on would normally specify the block interconnection. For the time response program there is a simple command (ROUTE) to do this. The user will be asked to enter a sequence triples (I,J,K) indicating signal flow from Block I to block J with gain K. For frequency response a single loop structure is assumed with transfer functions G(s) and P(s) assumed to be in series in the forward loop. One merely specifies whether the system is to be open loop or closed loop. The root locus program simply works with a single transfer function.

The use of variable names allows a part of the system to be modified without having to reenter the entire transfer function. To set (or change) the variable values, one again merely issues a simple command.

Both the time response and frequency response programs allow parameter optimization. When the optimization command is given, the program will ask the user for a desired transfer function. The coefficients given variable names during the system description will automatically be assumed to be the parameters with respect to which the optimization is to be performed, and the user will be asked for initial values for these. For time response, the function to be minimized is based on the integral square error between the desired and actual responses. For frequency response, it is based on a summation of the errors in the phase and logarithm plots of actual and desired transfer function at the frequency points for which the response is calculated. The optimization program used is the PRAXIS routine developed by Brent [1]. This is a conjugate direction procedure without derivatives based on Powell's method [2].

In addition, the COINGRAD programs have the following features:

(1) The programs determine upon being entered whether the initiating device is a graphic terminal, teletype or batch stream and set the input output formats accordingly.

(2) Useful reference lines such as constant damping ratio lines or the unit circle for root locus, or constant M contours for frequency response may be overlaid on the graphs produced.

(3) When appropriate, useful subsidiary information such as rise time, steady state error, gain margin and phase margin are calculated.

(4) The plot boundaries may be altered to "blow-up" interesting portions of the plots.

(5) Plots generated may be saved in MTS (Michigan Terminal System) files and redisplayed later without rerunning the program.

(6) CALCOMP plots may be simply produced (essentially just by asking for them).

(7) For graphic devices having a cursor which can be read, page position is checked before each output and writing off the bottom of the screen is inhibited. The user can erase the screen and continue when he is ready.

(8) In interactive terminal mode (graphic terminal or teletype) both program and attention interrupts are trapped and control is returned to the command structure, not to the computer operating system, Thus allowing the user to take some corrective action and make another try.

(9) Both tabular and graphic data may be obtained.

As an illustration, the following transfer function was entered into the

$$G(s) = \frac{s^2(.0016s^2+.032s+1)(.3s^2+s-1)}{s(.0016s^2+.032s+1)(.3s^2+s-1)+10(.23s+1)(s+1)}$$

root locus program. The resulting locus is shown in Figure 1, together with a reference line corresponding to a damping ratio of .5.

## Conclusions

The programs described here provide a means of introducing the power and utility of computer aided analysis and design tools for automatic control to undergraduate students. They perform the basic mundane calculations required for classical control system analysis, making it possible for students to concentrate on the principles involved, not repeated calculation of system behavior. Too often students without computer aids become more embroiled in eliminating arithmetic errors than in understanding the subject. With computer aids it is possible to study more realistic and meaningful problems.

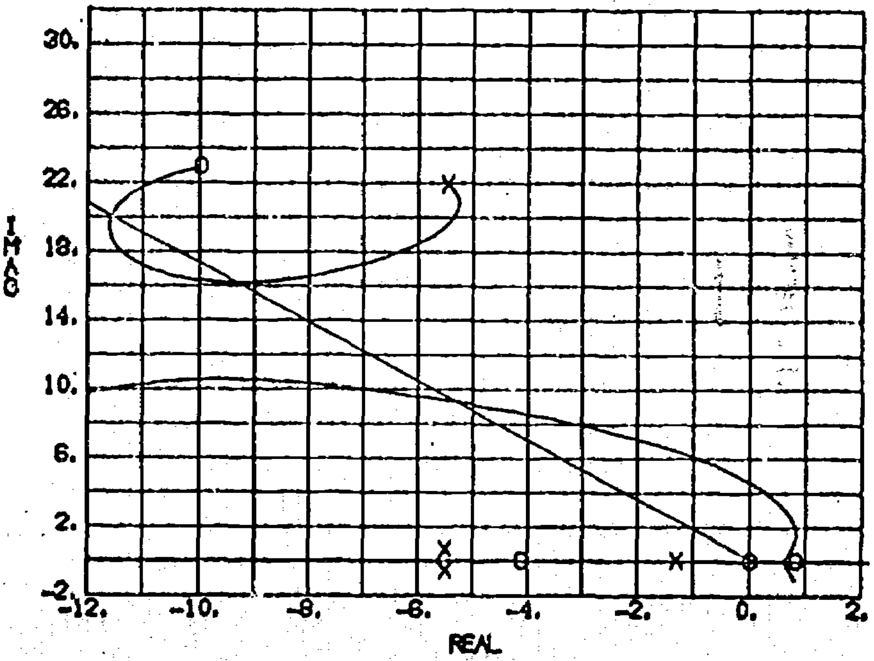
Most important in this implementation of the basic control calculations are the graphical interaction which takes place between the user and computer, and the simplicity of use. The student can enter his problem description simply and see the pertinent calculations displayed almost immediately in easy to interpret graphical form. Errors can be immediately corrected and a number of trials investigated rapidly, making it possible to deal with problems efficiently. The user oriented data formats have proven to be very easy to learn, and students have been able to begin using COINGRAD and TACL with minimal instruction. This ease of use is important, for without it much of the advantage of computer aids for instruction would be lost by time needed to learn how to use them.

## References

1. Brent, Richard P., "Algorithms for Finding Zeros and Extreme of Functions Without Calculating Derivatives," Ph.D. Dissertation Computer Science Dept., Stanford University, February 1971.

2. Powell, M. J. D. "An Efficient Method for Finding the Minimum of Several Variables Without Calculating Derivatives," The Computer Journal, Vol. 7, pp. 155-163, July 1964.

3. R. A. Volz, D. Conliffe, M. Dever, T. Johnson, J. Aravena, The Development of Computer Instructional Aids for Undergraduate Control Education, Final Report, NSY Grant 8313, University of Michigan, August 1973.



RT LOCUS

FIGURE 1