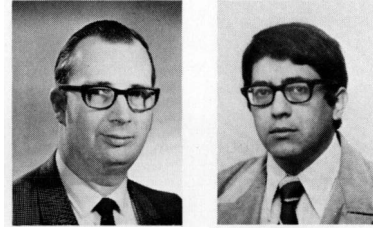


TACL- a timeshared hybrid system for control laboratories



by

Richard A. Volz
Computer Information and Control Engineering
University of Michigan
Ann Arbor, Michigan 48104

and

Jorge Aravena
Departamento de Electricidad
Universidad de Chile
Santiago, Chile

RICHARD A. VOLZ received the BS degree in electrical engineering from Northwestern University in Evanston, Illinois in 1960 and pursued his graduate studies there under a National Science Foundation fellowship, receiving the MS and PhD degrees in 1961 and 1964. In 1964 he joined the staff at the University of Michigan at Ann Arbor, where he has worked on optimal discrete control, radio navigation, multipath problems in high-frequency radar systems, computer-aided design, hybrid computation, and computational methods of optimization. He is currently an associate professor in the Department of Electrical and Computer Engineering and Director of the Computer Graphics Laboratory in the College of Engineering.

JORGE L. ARAVENA received his degree in electrical engineering from the Universidad de Chile in 1966. He joined the staff of the Electrical Engineering Department of the same university in 1967, working in the area of automatic control. From September 1970 until December 1972 he followed doctoral studies at the University of Michigan, and is completing his doctoral dissertation in some aspects of optimal control theory. His research interests include optimal control theory, simulation, and industrial computer control. He is presently at the Electrical Engineering Department of the Universidad de Chile, where he is in charge of the Computer and Automatic Control Laboratory. He is currently developing teaching facilities for undergraduate control courses and graduate courses in digital computer control.

ABSTRACT

A generalized control-system model encompassing a wide variety of systems has been programmed on the analog portion of a hybrid computation system. Five terminals with storage oscilloscopes are connected to the hybrid computer and are serviced on a round-robin basis. The generalized system can be configured in milliseconds to a specific problem by setting the values of the digital coefficient units and switches of the analog system. A library of

specific problems is kept on the system's digital disk. Students use the system by referencing a problem in the library and entering parameter values to define the problem from their terminals. The per-solution time varies with the time scale selected, but averages around a few hundred milliseconds. A user-oriented compiler helps the instructor define new problems to be added.

Digital values are assigned by the student to selected parameters of specific problems, and families of curves representing system performance as a function of control settings may be displayed on the student's terminal. The terminals also display digital data and alphanumeric information.

INTRODUCTION

It would be almost impossible to over emphasize the importance of computers, both analog and digital, to the understanding of engineering concepts and to engineering design. For the past decade or longer the use of simulation in analysis and design has been vital to the development of the engineering profession. Yet, as important as computers are now and have been in the past, they will be even more so in the future. One of the major keys to this continued growth of computer-aided design will be rapid *graphical* 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 on a cathode ray tube will be of utmost importance. The adage, "A picture is worth a thousand words" is often true in engineering. In fact, engineers often spend considerable time translating their data to graphical form to help them obtain a better intuitive feeling for the problem. With the tedium of translating data to graphical form removed, the engineer will be able to examine a much broader range of design possibilities in a shorter time and obtain a much better understanding of the system with which he is working.

With the important role that these tools will play in the future of engineering, it is extremely important that universities make their students aware of the potential role of computers and prepare their students for using computers effectively. This has been attempted in varying degrees in a number of ways at many locations, using both analog and digital approaches. A common approach to this in the control area has been to conduct a number of laboratory experiments on an analog computer. The student then gains both insight into the utility of computer simulation and an introduction to analog programming. In our case the computer is a hybrid machine, the student's attention is focused on the *control problems*, not on how to do the *analog programming*.

The major difficulty with the usual analog approach is that too often the student becomes so embroiled in the mechanics of analog programming that he loses sight of, or at best gives minimal thought to, the control-engineering principles he set out to study. Moreover, he is limited in what he can cover in his course work. Too often the time he devotes to analog programming is subtracted from his study of control engineering.

It is important to distinguish here between acquainting a student with the utility of computer tools in some branch of engineering and teaching him how to program the computer to achieve useful results. The former by no means requires the latter. In fact, when a student is primarily engaged in learning an engineering discipline, it is better to teach engineering and programming separately. The engineering student has enough to grasp without becoming entangled in learning more than minimal programming techniques. If computer tools can be made friendly to the user, i.e., easy to learn and easy to use, many benefits accrue. The reduction or elimination of time-consuming computation enables the student to concentrate on the basic concepts involved. Further, if he can readily use computer tools to develop and reinforce his understanding of the discipline he is studying, he will in the process acquire an appreciation of the use of computers in analysis and design. The details of the programming involved can and should be left to specialized courses. The important point is that the initial familiarization of a student with the use of computer aids should be through *user-oriented* programs.

Several efforts along these lines have been made in the area of control systems utilizing hybrid computers. Martin¹ and Hoberock and Thompson² have both developed time-shared simulation systems for automatic-control studies. Basically these systems consist of several permanently patched problems on a single set of hybrid computer boards. A student user at a terminal may then switch between these to select the problem he wishes to study. One of the major limitations of these systems is that for any one set of boards only a small number of problems can be programmed and that to change a problem would require rewiring the boards and rechecking out the wiring configuration, a time-consuming task, whether done by the professor or by a student.

The approach described here³ is based, instead, on the development and patching of a single general-configuration system on a hybrid computer. The system, dubbed TACL for Time-shared Automatic Control Laboratory, is a time-shared simulation system which allows the user to investigate the properties of a wide variety of dynamic systems. 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 section for the particular system to be studied by setting the gains of unused portions of the general system to zero (effectively an open switch) or unity (effectively a closed switch) as appropriate.

Student users of TACL, however, need not be concerned with how the "wiring" of a specific problem is done. The user portion of the system operates by reference to a library of predefined problems which the instructor has prepared. To the user it appears as if the system consists only of a model of the problem he has chosen from the library. However, because of the high-speed devices of the hybrid computer all users may be working on the same or different problems. The system can be reconfigured in at most 50 milliseconds.

In addition the system has 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 from the student's terminal. 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 preparing problems, a problem-generation program has been written. It 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 functions, values for slope, saturation level, or deadzone in nonlinear devices. The problem generator will assist in determining scale factors and will take care of all the necessary book work on coordinating scale factors. It will also correlate problem-oriented information with specific device addresses in the computer, relieving the instructor of some of the major burdens of problem preparation.

TACL CONFIGURATION

The TACL simulation system consists of a general configuration control system programmed on the analog portion of the AD/4 in the Simulation Center of the University of Michigan, and a digital executive program, programmed on the Center's PDP/9, operating concurrently. Student users communicate with the system through the terminal units which are interfaced with the hybrid computer as illustrated in Figure 1. 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 of TACL (other than the terminals and computers themselves) are the half-million-word disk of the PDP/9, which stores the

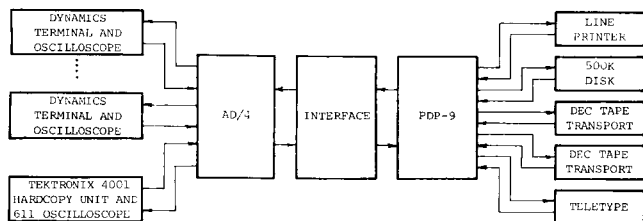


Figure 1 - Hybrid system used in TACL

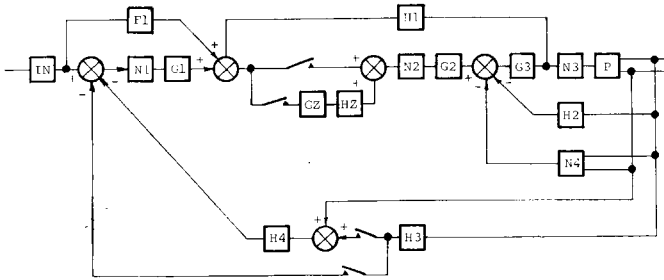


Figure 2 - General configuration of the TACL system

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 or as analog switches. It is the use of these converters as switches as well as coefficient potentiometers which allows problems to be rapidly reconfigured to satisfy the needs of individual terminal users.

The general system programmed on the analog portion of the system is shown in Figure 2. (The functional capabilities of each of the blocks in this system are shown in Table 1). Each problem studied within the TACL system must be capable of being generated within the context of this general system. The calculations for the sampled-data blocks shown are performed on the digital unit.

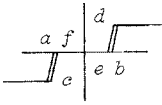
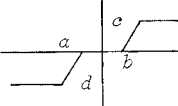
This general configuration is completely invisible to the user. The system will respond as if it consisted only of the specific problem which he is studying. Problems are defined by arrays of coefficient values, switch position and pointers relating certain of the coefficients to terminal thumbwheels (these are the coefficients controlled by the student). 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: The digital program sequentially interrogates terminals until it sees a request for a solution. It then locks on to the terminal, determines the problem being worked at that terminal, loads the fixed data arrays, etc. into core, reads the settings on the student's terminal, sets the switches and coefficients on the AD/4, and generates a solution to the problem in accordance with the settings on the student's terminal. To the user, then, it appears that the system consists only of his problem.

DYNAMICS TERMINAL OPERATION

The control portion of the student's dynamics terminal is shown in Figure 3. The configuration buttons are used to send an eight-bit instruction word to the hybrid system; it contains such information as problem number and special function requests. The parameter thumbwheels are used to enter user data (such as time constants, gains, etc.) to the system being studied. Assignment of these thumbwheels varies from problem to problem and is specified in a manner to be described below.

There are 16 analog trunk lines bringing analog data from the computer system to the terminal. These carry voltages proportional to the various solutions being generated. Thumbwheel selectors may be used to display any pair of these voltages on the oscilloscope. For each problem a description of the allocation of these is provided (discussed later). The

Table 1

Block name	Characteristics
IN	$A_1 \sin(\omega t) + A_2 \cos(\omega t) + A_3 + A_4$
F1, G3	$\frac{A_2 s^2 + A_1 s + A_0}{s^2 + B_1 s + B_0}$
N1, N2	 slope K
G1, G2, H1, H2, H3, H4	$\frac{A_1 s + A_0}{s + B_0}$ In the case of H4, the input may come from H3, the plant derivative, or both.
P	$\frac{A_3 s^3 + A_2 s^2 + A_1 s + A_0}{s^4 + B_3 s^3 + B_2 s^2 + B_1 s + B_0}$
N3	 slope K
	Similar to N1 and N2 except that hysteresis is omitted.
	$YN4 = B(XN4) - CC \cdot \cos(C3 \cdot YP) - CS \cdot \sin(C3 \cdot YP)$
	$XN4 = D1 \cdot X_1 X_2 + D2 \cdot \text{sgn}(X_1 \cdot X_2)$
	$X_1 = YP^i \cdot DYP^j \cdot DYP ^K + e$
	$X_2 = YP^l \cdot DYP^m \cdot (A1 \cdot \cos \omega t + A2 \cdot \sin \omega t)^n$
	YP = output of block plant
	$DYP = \frac{d(YP)}{dt}$
	This nonlinearity allows some rather complicated functions of the output of the plant P and its derivative to be simulated. All coefficients and exponents may be set by the user. YN4 is the output of N4 and XN4, X_1 and X_2 are internal variables to the block.
GZ	$\frac{A_4 Z^4 + A_3 Z^3 + A_2 Z^2 + A_1 Z + A_0}{Z^4 + B_3 Z^3 + B_2 Z^2 + B_1 Z + B_0}$ This represents a sampled-data path in the system. The sample period may be selected by the student.
HZ	Same form as GZ.

NOTE: All coefficients in this table are digital values which may be set by the instructor when the problem is generated or (optionally) controlled by the user when a solution is run.

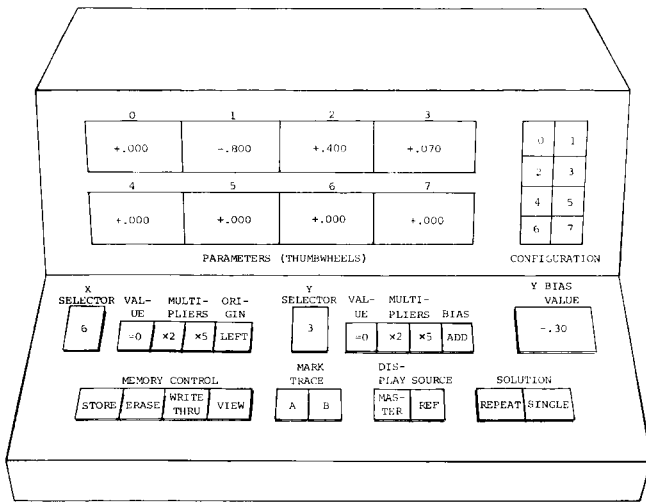


Figure 3 - Control portion of student's terminal. Each terminal also has a storage oscilloscope which holds displays until they are erased.

multipliers, 10, 2, and 5, may be used to amplify the signal to provide a better view of the solution. It is also possible to add a bias displacement to the vertical coordinates by depressing the ADD button and setting the displacement value on the Y BIAS thumbwheel.

There are four memory control buttons on the lower left portion of the terminal. The left two perform the obvious functions indicated, and the third (WRITE THRU) is disconnected. Completing the controls are the solution request SINGLE and REPEAT. Normally, the oscilloscope is placed in the STORE mode and a SINGLE solution requested. If REPEAT is depressed the solution will be generated each time the computer interrogates the terminal. This may be useful in studying the effect of parameter variation or the phase plane response for a stability analysis. Use of the MASTER button allows solutions generated on one terminal to be viewed on all of the others. In this way the instructor may demonstrate a solution to the entire class.

TACL OPERATION

The general system of Figure 2 is programmed on the analog part of the computer. The exact nature of the system for any given problem is determined by the settings of 120 coefficient devices. At run time (i.e., at student operating time) the TACL executive program sequentially interrogates the state of the solution request buttons on the terminals. Whenever a request is detected, the requesting terminal is given a grant signal which connects it to the computer. The computer then performs the following sequence:

1. Read the configuration buttons and decode the problem number and function request.
2. Read the parameter thumbwheels.
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-

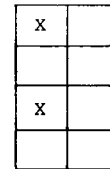
speed Digital Coefficient Units of the AD/4 this only takes a few milliseconds).

5. Run the desired solution.
6. Disconnect the requesting terminal from the computer.
7. Interrogate the next terminal.

The configuration buttons are used as follows:

Configuration button	Usage
0 - 3	Problem number
4 - 6	Function number
7	Request hard-copy service

In all cases, the smallest numbered bits are the least significant. Thus, to request problem number 1 and function number 1, one would have the situation illustrated below, where an X indicates that a button is depressed.



It is evident that this assignment allows 16 problems and 8 functions to be specified. To date only five of the functions have been assigned.

Function number	Operation
0	Write alphanumeric text or graphic data
1	Run a continuous dynamic solution
2	Run a dynamic solution with a time delay
3	Make hard copy
4	Run a sampled-data solution
5 - 7	Unassigned

In addition to the operating conditions imposed by the programming of the TACL system, a number of operating conventions have been adopted for the convenience of users. First, to increase the flexibility of the system, an alphanumeric character generator developed at the Simulation Center has been adapted to TACL. This allows coordinate systems, error messages, and instructions to be printed on the terminals. With every problem stored in the system there are several pages of descriptive text identifying the problem, what the thumbwheels control, and how the data channels are allocated. In addition, several different graph formats are provided for each problem. Function zero may be set on the configuration buttons to request that a page of this graphic text be displayed on the terminal screen. Thumbwheel number seven is used to identify the particular page to be displayed.

The user would normally begin an operating session by first viewing the description for this problem. By convention the text pages are numbered sequentially beginning with one, and the first page will always contain a table of contents for the rest of the text.

The following sequence is used to write a page of graphics.

1. Place the oscilloscope in the STORE mode (ERASE screen if necessary).
2. Set function to zero.
3. Enter problem number in buttons 0 - 3.
4. Set graphics page number/100 on thumbwheel 7, e.g., to write file 1, set thumbwheel 7 to + 0.010.
5. Depress SINGLE solution button.

When a user is ready to generate a dynamic solution to the system he is studying, he should first display the text page containing the coordinate system he wishes to use. TACL will remember which coordinate system is being used and locate the origin of his solution display accordingly. A typical sequence would be:

1. Place the oscilloscope in the STORE (ERASE screen if necessary).
2. Use the procedure above to display the desired coordinate system.
3. Enter the desired parameter values on the thumbwheels.
4. Enter function one, two, or four according to the type of solution requested.
5. Depress SINGLE button to obtain a single solution.

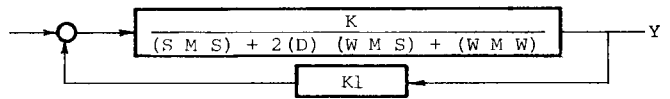
At the student's option a Tektronix hard-copy unit produces a permanent record of any solution generated. Only one of these Tektronix units is available and must be shared by all users. As it is one of the early units produced, it must be individually matched to the oscilloscope with which it is to work, making it difficult to switch the unit from one scope to another. To solve this problem, one oscilloscope is permanently dedicated to the unit. When a terminal is given use of the hard-copy unit, what actually happens is that the dedicated hard-copy oscilloscope is placed in parallel with the terminal oscilloscope. Thus, any signals displayed on the terminal are also written on the hard-copy oscilloscope. Therefore, after being issued a grant for use of the hard-copy unit, the user must erase his screen and build the display he wants to copy. When he is ready to make his copies, he sets the number of copies/100 on thumbwheel 7 (a maximum of three are allowed) and issues a solution request with function set to 3.

Example

To illustrate the use of the terminals with the TACL system, we shall consider a simple example which is identified as problem number 0. With no additional information, the first step a user takes is to display the first page of text material for this problem. He does this by selecting problem 0 (first four configuration buttons in the UP position) and function 0 (function buttons 4 - 7 in the UP position). He then sets the fraction .01 on thumbwheel 7, places the display oscilloscope in STORE mode and then erases the screen. Depressing the SINGLE solution request button results in the display shown in Figure 4.

SECOND ORDER SYSTEM

THIS PROBLEM ALLOWS THE STUDY OF THE FOLLOWING SECOND ORDER SYSTEM



STEP AND RAMP INPUTS ARE AVAILABLE TO THIS SYSTEM. INITIAL CONDITIONS ARE ASSUMED ZERO.

TABLE OF CONTENTS	
GRAPH FILE	CONTENTS
2	THUMBWHEEL CONTROLS
3	GRAPH OVERLAY - ORIGIN LEFT CENTER
4	GRAPH OVERLAY - ORIGIN LOWER LEFT
5	GRAPH OVERLAY - ORIGIN CENTER

Figure 4 - Display page 1 of text on oscilloscope in student's terminal.

This display shows that the system under consideration is a simple closed-loop second-order system. The thumbwheels give the student control of the gain, damping ratio, natural frequency, and the feedback gain. This is indicated by the appearance of the variable names in these positions in the transfer function displayed on the screen of the oscilloscope. In addition, subsequent pages of text displayed on the oscilloscope tell the student how to control the time scale and select the input functions to be displayed. The Table of Contents states that page two contains the thumbwheel assignments and data-channel allocations. Subsequent pages of text lay down graph formats.

To display the second page of text one merely changes thumbwheel 7 to .02, ERASES the screen and again depresses SINGLE. The resulting display is shown in Figure 5.

Suppose it were desired to run a family of step responses corresponding to the open-loop system with a natural frequency of 1 and a damping ratio varying between 0.2 and 1. From Figure 5 it is evident that the thumbwheels should be set to the following values to run the first of these curves (damping ratio=0.2). The open-loop gain is chosen to be 1, and the solution interval is chosen to be 5 seconds.

Thumbwheel number	Value
0	.1
1	.999
2	.04
3	.0
4	.5
5	.0
6	.1

The choice of step amplitude was made in order to make scaling easy.

Before asking for the solution, the student should lay down a coordinate system. Let us choose the one

THUMBWHEEL ALLOCATIONS	
THUMBWHEEL	PARAMETER
0	K/10
1	W
2	D/5
3	K1
4	STEP AMPLITUDE
5	2(RAMP)
6	2/TMAY
7	GRAPH NO.

DATA CHANNEL ALLOCATION	
CHANNEL	INFORMATION
0	TIME BASE
1	SYSTEM OUTPUT - - Y
2	YDDT/10
4	ERROR SIGNAL
10	INPUT SIGNAL

Figure 5 - Page 2 of text is displayed on student's terminal.

on page 4 of the text. Setting .04 on thumbwheel seven, ERASEing the screen and depressing SINGLE results in displaying this trace on the screen. Next, enter function one by depressing configuration button 4 and again depress SINGLE. This results in the solution's being displayed on the screen. Then by varying thumbwheel two up to .2 and depressing SINGLE each time we build up the family of curves, one curve at a time, for a set of damping ratios. The composite curves and coordinate system are displayed in Figure 6. Notice the number in the lower right hand corner which identifies the terminal that generated these solutions.

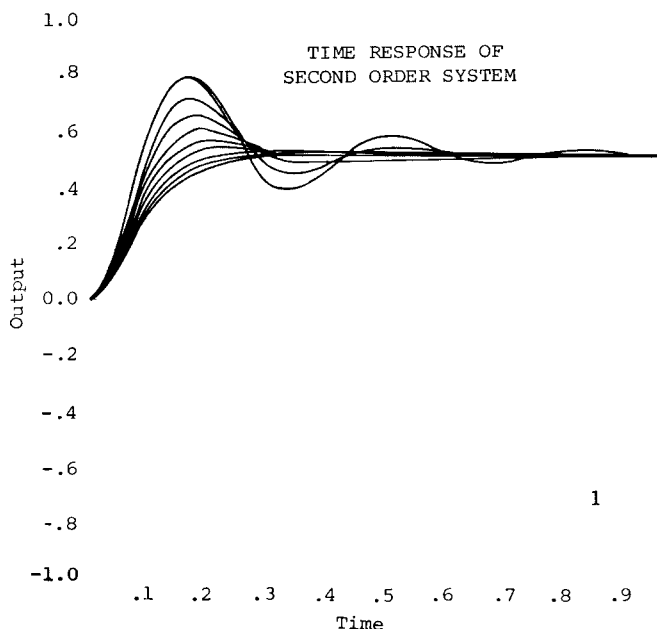


Figure 6 - Family of curves showing the effect of varying the damping ratio of a simple second-order system.

As this example shows, terminal operation is easy to learn and simple to use. The predefined files contain a wide variety of dynamic problems, any of which the student users may select for their studies.

TACL COMPILER

The TACL operating system described above operates on a set of predefined problems. Before students can use a problem, its defining arrays must be specified. Unwanted blocks must be identified and their coefficients set to zero, which is equivalent to opening a switch. System parameters must be specified, thumbwheel control of desired parameters must be set up, problem variables must be properly scaled, and appropriate alphanumeric information for the "text" and other oscilloscope displays must be generated. This work is normally done by the instructor and does not involve the students. To aid in this task a problem-generation program called the TACL compiler was written.

It is the compiler's objective to make the instructor's task of generating problems as easy as possible. The instructor interacts with the compiler program via a teletype terminal through which he issues a series of commands and receives the computer's responses. The basic commands are simple and are limited in number. In the response to these commands the compiler will interrogate the user regarding the details of his request (command) and at several points it allows the user to request explanatory information about the compiler's command structure. With this kind of help even an instructor inexperienced in TACL can readily define problems.

Typically, problems are defined by specifying the contents of each block in the system, one block at a time. For convenience a "default" specification for all blocks is made by initializing the compiler, and the user need only specify those blocks which he wishes to change from the "default" condition. The "default" is unity for all blocks in the forward path except plant (which cannot be set to unity) and the feedforward block F, which is zero, and is zero for all blocks in the feedback paths. Block specifications are in terms familiar and convenient to the user. The compiler takes care of coordinating scale factors and translating the specification to values for specific devices.

To define a block, one types the name of the block, followed on separate lines by various block definition commands. Thereafter, the compiler will interrogate the user for the needed parameters. When all blocks have been defined, the compiler program will check the entire system to ensure that all coefficients fall within an allowable range. To check the validity of estimated scale factors, the user may enter trial values for the thumbwheel parameters and obtain solutions for these. During the solutions the peak values of pertinent variables are obtained and displayed for the user. In this way he may judge the adequacy of the scales factors used and then make revisions and retrials as necessary.

CHARACTER GENERATION

A significant part of the system is the character generator. All alphanumeric characters and graphic figures are generated by software. Each character or figure is represented by a set of straight lines which are produced on the analog using a scheme developed at the Simulation Center of the University

of Michigan. Each line is drawn by tracing out the convex combination of its end points.

A portion of the problem-generation system facilitates definition of the proper sequence of line segments. Each page of text is broken into several sections. For each section the user enters the starting location, character size, and whether the data is to be written horizontally or vertically, and then types the text as he wishes it to appear. The program produces the file of data in the form required by the character generator. Line drawings are slightly more involved because the user has a menu of line segments, curves, and elementary figures to use as building blocks. A figure must be coded to put these together with the proper size and location to obtain the desired display.

CONCLUDING REMARKS

The TACL system, then, is comprised of an easy-to-use time-shared terminal system which operates on (1) a set of predefined problems stored on a disk and (2) a problem-generation system which allows an instructor to readily define the problems and descriptive text and graphics. It has been used over the past four years in different capacities in three different undergraduate courses. It has been used occasionally for demonstrations in conjunction with a lecture course on basic automatic control theory, as an occasional laboratory tool in a course on dynamic systems and modeling, and as a central means of study in an undergraduate control laboratory.

The difficulties which have been encountered with the system have primarily been occasional failures in the system hardware and noise on some of the transmission lines. However, as the Simulation Center performs diagnostic tests on the computation system each morning, these troubles have been kept to a minimum.

The laboratory, which is based around the use of TACL, is structured so as to provide the students

with an opportunity to study the hardware aspects of control systems as well as simulation. Approximately the first month of the course is spent in the laboratory working with actual servo units. The simulation work is begun by simulating these laboratory units and comparing the simulation with results obtained by experimental measurements. The remainder of the time is then spent using the TACL system to explore a variety of control concepts. Included among these are (1) an introduction to nonlinear systems, (2) proportional, derivative, and integral control, (3) state-variable control, (4) design of an aircraft control system, (5) Van der Pol's equation, and (6) data sampling and reconstruction. Operation with this system has allowed much more material to be covered in the laboratory with less effort on the part of the student than in the past. Student acceptance of the system as indicated by their course evaluations has been quite good. Overall, we have found that the system has been a very useful tool for laboratory instruction and that its use has significantly increased students' insight into and understanding of dynamics problems and control systems.

REFERENCES

- 1 MARTIN, D.C.
The Analog Computer Is Dead
SIMULATION vol. 15 no. 3 September 1970
pp. 125-130
- 2 HOBEROCK, L.L. THOMPSON, G.
Demonstrations in Feedback Control and Dynamic Systems for the AD-Dynamics Terminals
University of Texas report to Reliance Electric Company February 1971
- 3 VOLZ, R.A. CONLIFFE, D. DEVER, M.
JOHNSON, T. ARAVENA, J.
Undergraduate Control Education: Final Report
NSF grant 8313 University of Michigan
August 1973