Int. J. Engng Ed. Vol. 12, No. 5, p. 373–381, 1996
Printed in Great Britain.
0949-149X/91 $3.00+0.00

Educational Multimedia Application for Block Diagram Manipulation*

TREFOR J MORGAN
MOHAMMAD ALDEEN
Department of Electrical and Electronic Engineering, The University of Melbourne, Parkville, Victoria 3052, Australia

This paper reports on recent development of multimedia-based teaching software facility in the University of Melbourne's Control Systems Laboratory. It combines, text, graphics, audio and video clips to form a comprehensive teaching and learning software. The software runs under X-Windows, and can be easily ported to most Unix workstations. It enables students to perform, on line, complex tasks such as block diagram manipulation and linear system analysis. The software can be interfaced with MATLAB for more advanced and complex system analysis and design.

INTRODUCTION

A FIRST course in classical control, network or system theory involves the teaching of transfer function and block-diagram manipulation. This is then used subsequently for the study of systems in general and feedback control in particular. Feedback control systems may consist of one or a number of loops, each representing a component or subsystem of the overall system. Depending on the kind of system, these loops may consist of a number of blocks and be connected in a variety of ways, such as series, parallel, negative or positive feedback. In order to study the behaviour of control systems, it is desirable to reduce these complex representations into simple standard ones.

This paper describes a multimedia software package that has been developed in the Department of Electrical and Electronic Engineering at the University of Melbourne, Australia. The development of this package is a part of a wider process [1–8] aimed at improving the quality of teaching control theory and therefore enhancing the learning aspect of it by students. The software facilitates the manipulation and reduction of complex transfer representations of dynamic systems. Once the reduction process is complete, the software can then be used to analyse the system's time and frequency responses through using standard graphical methods such as root locus, Bode plots and Nyquist plots.

The software comprises text, graphics, audio and video clips supported by an on-line help facility. Students can, at any stage of their exercise, call upon an illustration video clip, for example, to explain how to use the software with guided tour or to have explained certain aspects of the theory involved.

In addition, the software is interfaceable with MATLAB, and the user may use either mode of operation by switching to and from MATLAB. Therefore after the student has performed the desired block manipulation task, they can easily use any of the more advanced and complex time and frequency domain analysis and design routines provided by MATLAB.

The first part of the paper establishes the need for the proposed software package and provides recommendations as how and when the package should be used. The second part gives a thorough explanation of the options available, their functionality and application. This is followed by an outline of the advantages of the package and its educational value.

Illustrative examples are given and the reader is taken through the package step by step. Through these examples the main features of the package are explained. It will be shown how a given complex feedback control system made up of many subsystems can be simplified into a standard one and used to carry out selected time- and frequency-domain stability analysis and design. Switching back and forth between the package and MATLAB will also be illustrated. Finally, conclusions with regard to the suitability of the package will be drawn.

BACKGROUND

At the University of Melbourne, Australia, control, networks and systems are first studied in the second year of the four-year bachelor of engineering degree course. These subjects deal with systems composed of a number of components, each one of which is represented by one or more transfer functions or blocks. In order for second-year

* Accepted 20 July 1996

students to be able to study such systems, it is necessary that a simple and reduced representation of them is found. The task of finding such a reduced representation by hand manipulation is both laborious and tedious. In the laboratory, such an exercise is futile and waste of valuable time that should instead be used for analysis and design.

The obvious alternative is to use software packages to perform these tasks. However, at this stage students have little or no exposure to commercial mathematical software packages such as MATLAB, MATRIX-X, CC, MATHEMATICS, etc. This lack of exposure is mainly due to students' lack of necessary background material which would enable them to use these packages effectively. Another reason is the complexity of these packages.

Faced with the problem posed above, work was initiated in the Department of Electrical and Electronic Engineering to develop a user-friendly multimedia-based software package that can be used, instead of commercial software packages, to perform such essential tasks as block diagram manipulation and time- and frequency-domain analysis of dynamic systems.

The software has been designed with the view of providing direct interaction with MATLAB, allowing the student to swap from one to the other. Thus students at later stages of their studies have the option of using the same package to access the wide range of analysis tools provided by MATLAB.

SOFTWARE DESCRIPTION

The package, named xTJ-Control, is designed to run on a Unix machine that supports X-Windows applications. It is written in the C programming language, drawing upon the resources of X-Windows (namely the Athena Widget set, Xlib and Xt) for the user interface and graphical displays. It also draws from the resources of the control toolbox of MATLAB for complex root finding, Bode plots, root locus etc.

The software allows the user to enter the transfer function (TF) for each component of the overall system in terms of its numerator and denominator polynomials. As each TF is entered it is displayed on the monitor as a block with a designated name. These blocks may then be arranged in any configuration (interconnection) desired. At each stage a block diagram representation of the system entered is displayed along with its transfer function.

In order to build complex systems, simpler subsystems are first built and then interconnected to yield the required configuration. Frequency- and time-domain analysis can be carried out on any subsystem or the entire system at any stage.

INTERFACE DESIGN

The main window consists of four regions, as shown in Fig. 1. The first region is the main

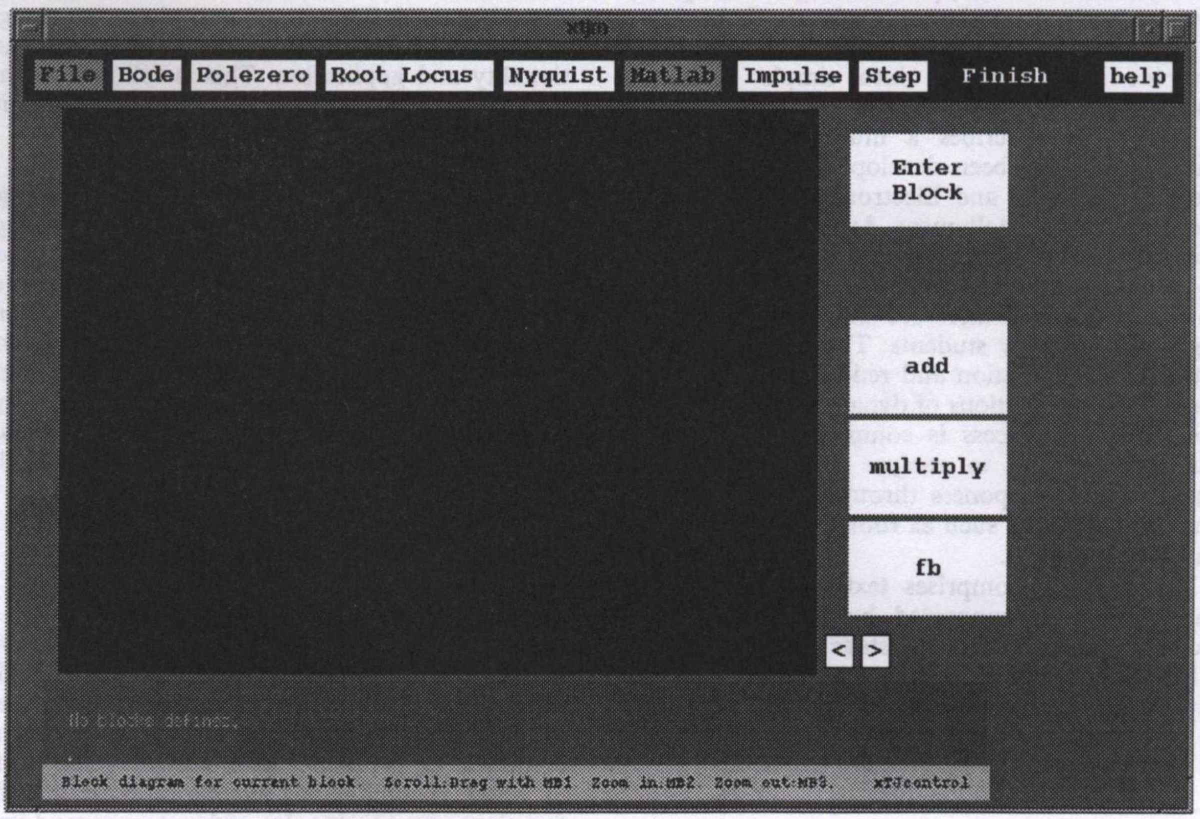
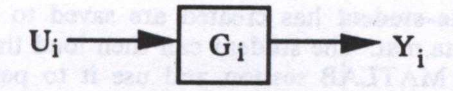


Fig. 1. Layout of the user interface.



$$G_i(s) = K_i \frac{N_i(s)}{D_i(s)}$$

Fig. 2. A simple block representation of a transfer function.

viewing screen which displays the graphical representation of the current block diagram. This region also shows the pattern in which individual blocks are interconnected. The second region is to the right of the main viewing window. In this region the most commonly used commands, such as entering a block or add two blocks can be accessed. The third region is at the top of the screen which displays the menu bar. The menu bar contains a variety of commands including options to save and retrieve information about given sets of blocks. It also contains various in-built frequency domain analysis tools and an option to save the current set of transfer functions in a MATLAB format. The bottom part of the screen, which is the fourth region, displays the transfer function for the current block diagram as well as a one-line description of the function which the student is currently pointing the mouse at.

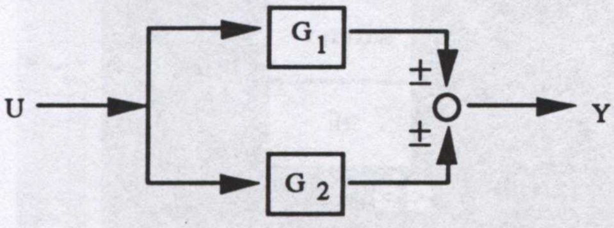
COMMANDS

The following is a brief outline of functionality of each command in the software. Commands can be accessed via keyboard or mouse.

Basic commands

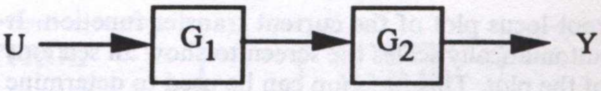
Enter Block: This command allows the student to enter the gain, K_i, the coefficients of the numerator polynomial, $N_i(s)$, and the coefficients of the denominator polynomial, $D_i(s)$, of a transfer function, $G_i(S)$, representing the ith component of a system, as shown in Fig. 2.

In Fig. 2, K_i is the gain and U_i and Y_i are the input and output variable associated with $G_i(s)$. When the student is satisfied that the TF is correctly entered, the student can easily save it by simply pressing the 'DONE' button. If, however,



$$\frac{Y(s)}{U(s)} = G_1(s) \pm G_2(s) = \frac{N_1(s)}{D_1(s)} \pm \frac{N_2(s)}{D_2(s)}$$

Fig. 3. Addition of two blocks.



$$\frac{Y(s)}{U(s)} = G_1(s)G_2(s) = \frac{N_1(s)\,N_2(s)}{D_1(s)\,D_2(s)}$$

Fig. 4. Multiplication of two blocks.

changes need to be made, the cancel button is pressed and the program returns the user to the main menu without saving the entered TF.

Add Block: Adds two blocks together, as shown in Fig. 3. Each block in turn may contain multiple blocks.

Multiply Blocks: Multiplies two blocks (subsystems), i.e. connects them in series, as shown in Fig. 4.

Feedback: This command is used to find the equivalent of either positive or negative feedback loops. The loop contains one block in the forward path and one in the feedback path, as shown in Fig. 5. Here again each of these blocks may contain a number of other blocks of any configuration.

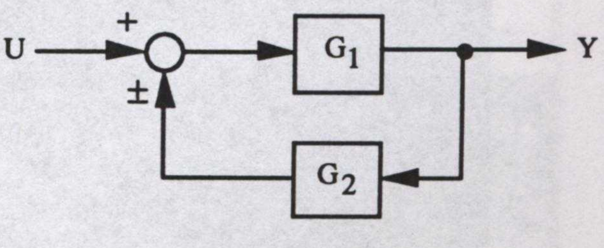
Utilities

There are several functions that can be used for analysis and design of simple control schemes. These are now briefly outlined below.

Bode plot: This command is used to plot the Bode plots (magnitude and phase angle) of the current transfer function. It automatically scales upper and lower frequency in rad/s, and gain in dB. The plots can be used to determine gain and phase margins of the system concerned. These may then be used for stability analysis or for subsequent use in the design of appropriate compensation schemes.

Pole-zero plot: This option produces a pole-zero diagram of the current transfer function. It automatically scales the screen to show all poles and zeroes. As well as showing the distribution of the TF zeros and poles, this option can be used to determine stability of the system.

Root-locus plot: This commands produces the



$$\frac{Y(s)}{U(s)} = \frac{G_1(s)}{1+G_1(s)G_2(s)}$$

$$= \frac{N_1(s)D_2(s)}{D_1(s)D_2(s) \pm N_1(s)N_2(s)}$$

Fig. 5. A block-diagram representation of feedback loop.

root-locus plot of the current transfer function. It automatically scales the screen to show all sections of the plot. This function can be used to determine the stability of the system, and for the design of appropriate compensation scheme, if required.

Nyquist diagram: This command produces a Nyquist plot of the current transfer function. This may be used for either compensation design or performance and stability analysis.

Step response: This command produces a time-domain response of the current system to a step change in its input.

Impulse response: This command produces the time-domain response of the current system to an impulse function.

File Commands

A number of commands are available to handle files and are accessible through the 'file' menu. These are outlined below.

Save blocks: Saves the blocks currently entered into a file.

Loads block: Loads any previously saved block.

Exit: Exits from the program, first prompting if the user wishes to fill out a feedback questionnaire form.

Quit: Immediately exits program.

MATLAB

The MATLAB menu allows the student to write to or read from a common file that is accessible to MATLAB. When the student chooses this option all the transfer functions that the student has entered as well as the composite ones that the student has created are saved to a file xtjmdata.mat. The student can then load this file into a MATLAB session and use it to perform further MATLAB supported operations.

Miscellaneous commands

Unix workstations are generally provided with a three-button mouse. Various buttons have different responses when pressed on different parts of the screen to give the student more control over the information they are looking at. The following is a summary of the various functions available through the mouse.

Button 1: Pressing mouse button 1 (MB1) and dragging on the main viewing area allows the student to scroll around the current block diagram. Pressing MB1 and dragging on the area with the transfer function, allows the student to view any part of the transfer function that does not fit on the screen. This button is also the button used to select the various menu options.

Button 2: Pressing mouse button 2 (MB2) on the main viewing screen allows the student to zoom in on the current block diagram. Pressing MB2 on the area with the transfer function, swaps between showing the function in polynomial form and pole-zero form.

Button 3: Pressing mouse button 3 (MB3) on the main viewing area allows students to zoom out of the current diagram. On other functions, MB3 is used to get extra help either in the form of a short audio track, short video clip or static image (see Fig. 6).

In addition there is a one-line help display at the bottom of the screen whenever the student passes the cursor over any of the functions.

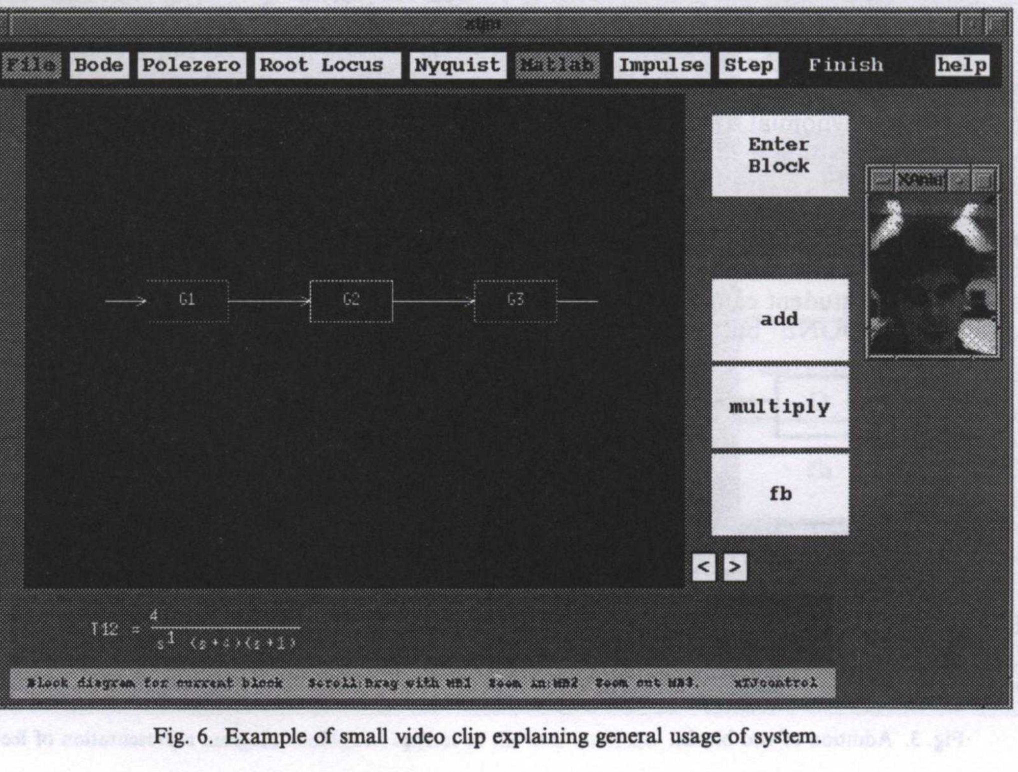


Fig. 6. Example of small video clip explaining general usage of system.

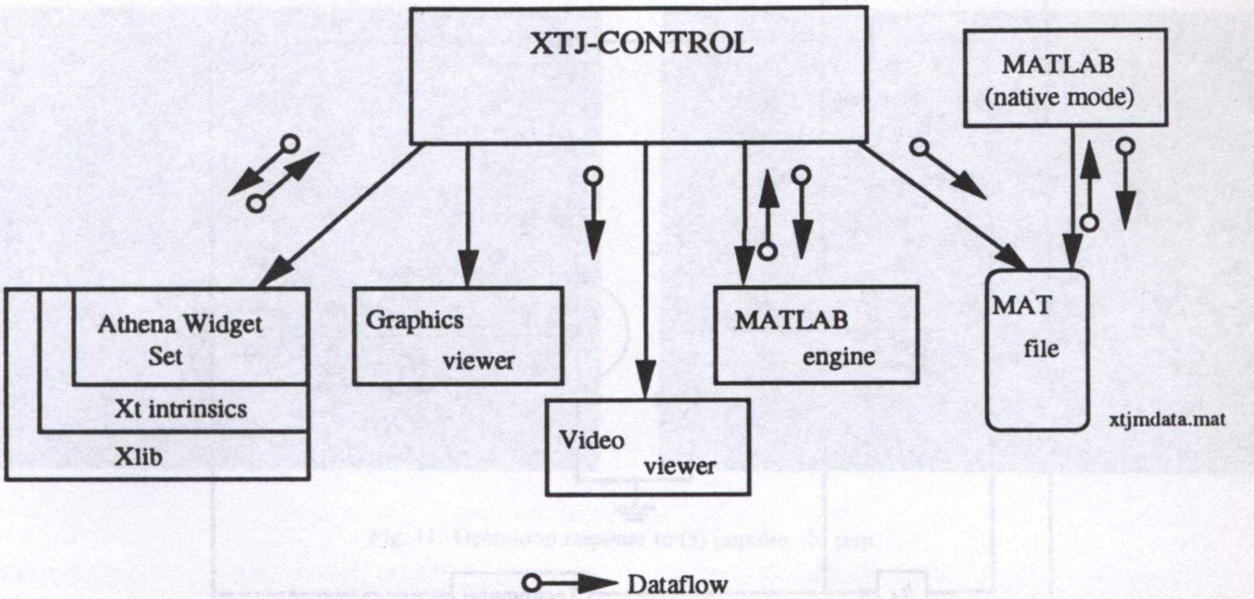


Fig. 7. A schematic of the main interactions between software.

PROGRAM STRUCTURE

The following is a brief description of the program structure and the interrelationship between the program and the UNIX environment. A schematic of this is shown in Fig. 7. This figure shows how xTJ-Control draws upon the resources of X-Windows (in particular the Athena Widget set) for its interface. The program also uses any standard graphical viewer (such as XV) and any audio/video player (such as Xanim or mpeg_play). Figure 7 also indicates how the main program uses MATLAB in two different ways. The first way is by direct communication with MATLAB

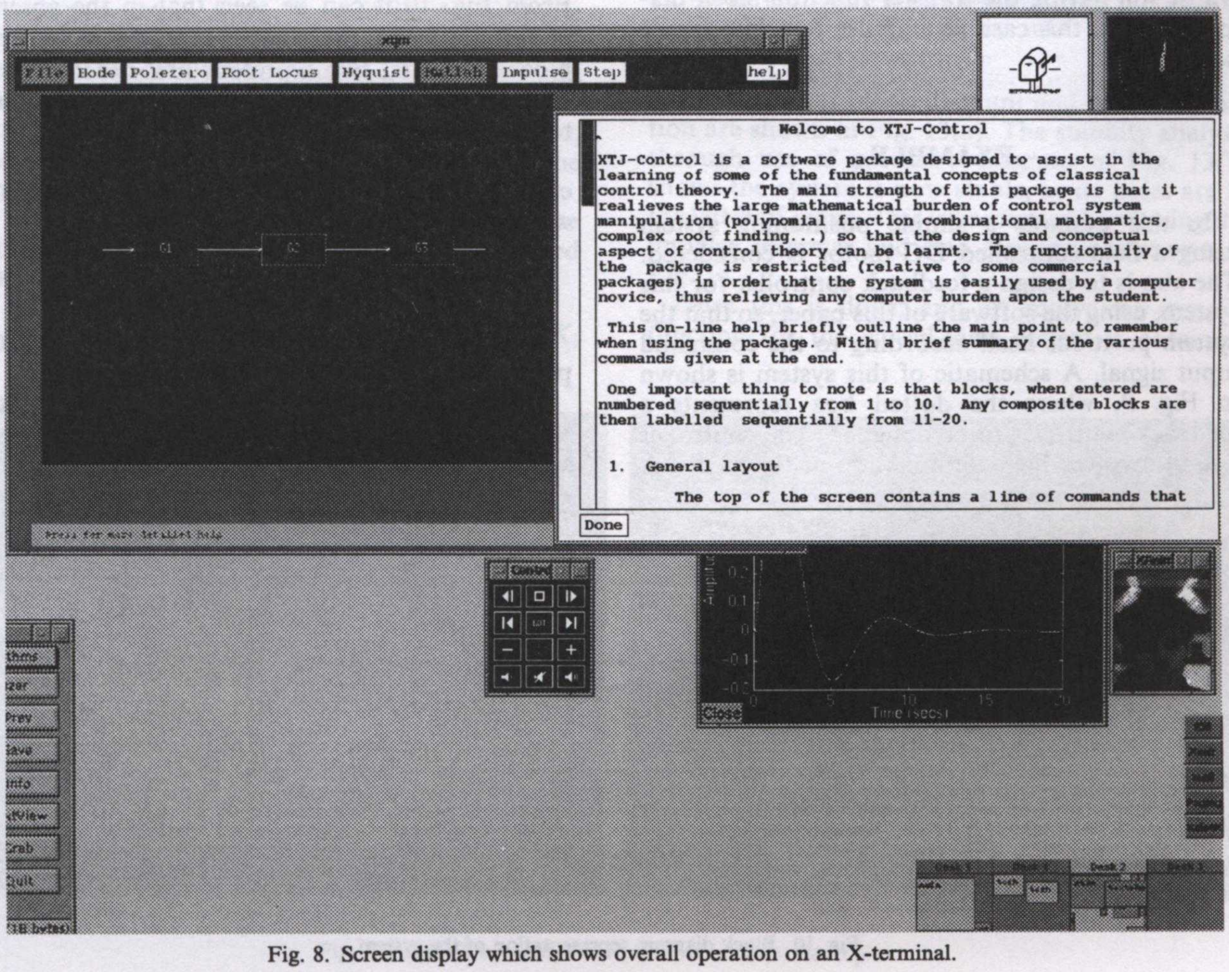


Fig. 8. Screen display which shows overall operation on an X-terminal.

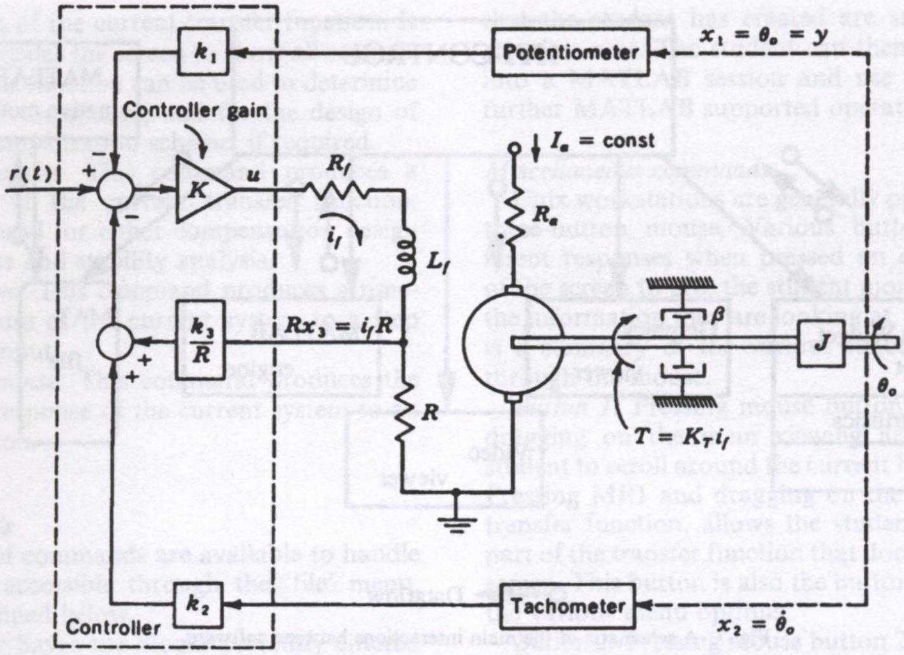


Fig. 9. Schematic of positioning system of field-controlled DC motor.

libraries (such as root-locus). The second method is indirect communication, where data is written to an intermediate file (xtjmdata.mat).

Figure 8 shows a typical session on an X-terminal that a student might have, demonstrating how a student can be listening to and reading help, whilst still having the transfer function/block diagram and in this case an impulse response in the background.

EXAMPLE

In this example a simple positioning system using a field-controlled DC motor is considered. The aim is to design a feedback controller for this system, using the software of this paper, so that the system positions itself according to the command input signal. A schematic of this system is shown in Fig. 9, where the dotted box represents a feedback controller to be designed. The controller design involves the determination of the feedback parameters k_1, k_2 and k_3. Figure 10 shows a block-diagram representation of the positioning system, with the controller.

In order to simplify the discussion, let us assume the following numerical values or the plant parameters: $L_f = 0.5$; $R_f = 2$, $K_T/J = 2$, and $\beta/J = 1$. From Fig. 10 it can be seen that in the absence of the controller, the plant is represented by three blocks, the first two represent first-order transfer functions with a simple gain and lag and the third represent an integrator.

To start the analysis and design process, we first enter the three transfer functions as G_1, G_2 and G_3, where

$$G_1 = \frac{2}{s+4}; \qquad G_2 = \frac{2}{s+1}; \qquad G_3 = \frac{1}{s} \tag{1}$$

The open-loop impulse and step responses of the plant, without a controller, are shown in Fig. 11.

To produce the open-loop transfer function the three blocks given in equation (1) are connected in series (multiplying transfer functions) The impulse

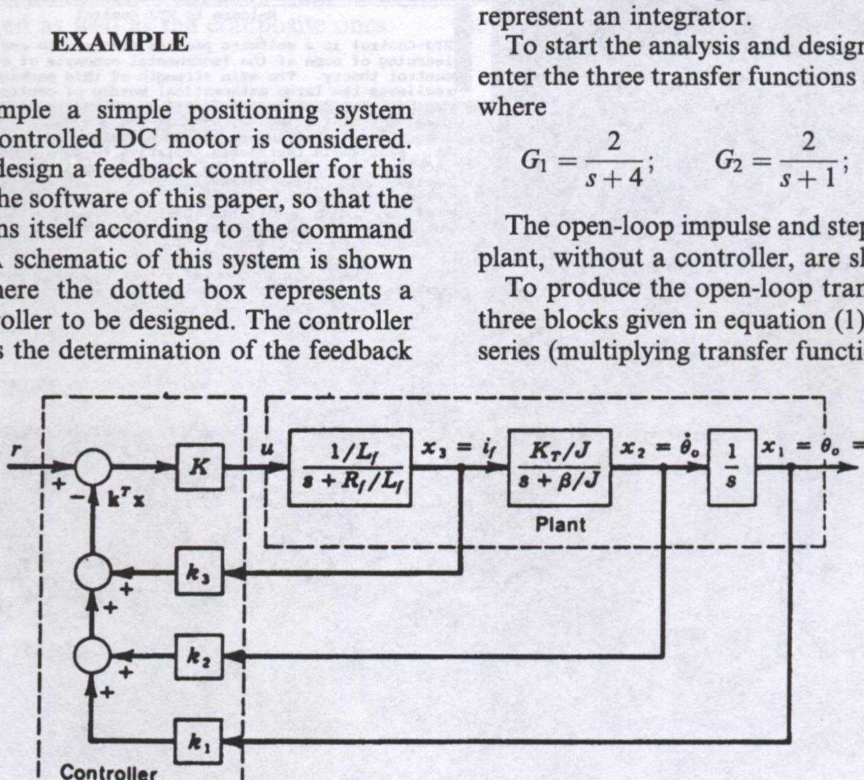


Fig. 10. Block diagram representation of the system.

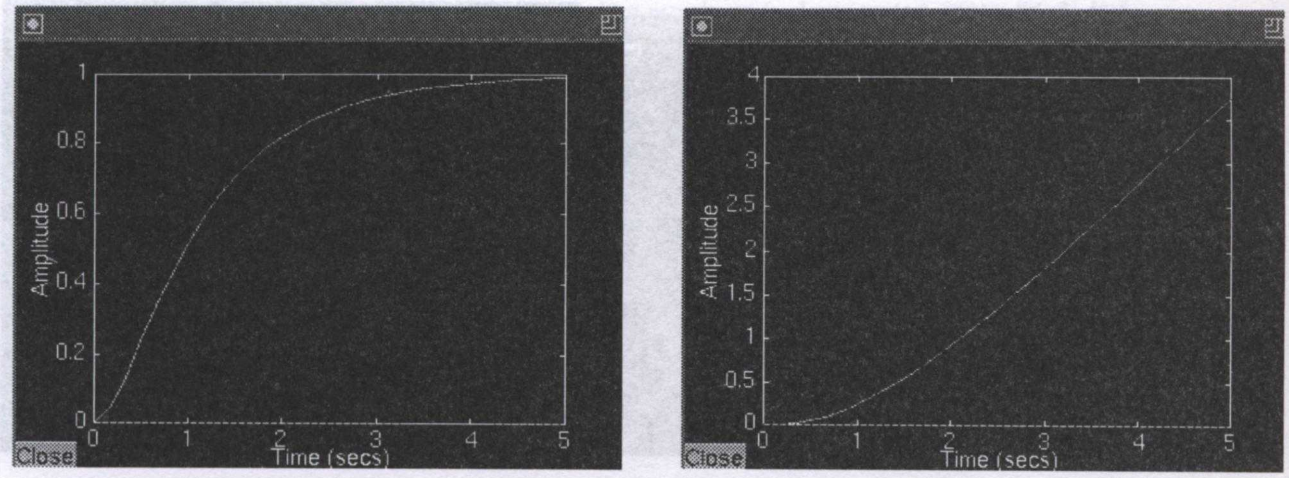


Fig. 11. Open-loop response to (a) impulse, (b) step.

and step response functions (Fig. 11) are obtained by using the respective function buttons provided. The root locus and Bode plot of the open-loop plant are shown in Fig. 12. These two figures indicate that the open-loop response is only marginally stable, does not follow the command signal and has a small gain margin. Therefore it is desirable to design a controller to improve its performance.

The controller design involves the determination of the parameters K, k_1, k_2 and k_3. Using basic control theory, a satisfactory controller can be designed to satisfy the tracking, transient and gain margins requirements. If, we choose, for example, $K = 1.5$, $k_1 = 1$, $k_2 = 2/3$ and $k_3 = 0$, then the closed-loop system will satisfy all the requirements.

In order to use the software to analyse the closed-loop system, the following steps are carried out:

1. Enter blocks G_1, G_2, G_3, as well as values for K, k_1, k_2 and k_3 (G_4, G_5, G_6, G_7 respectively). This is done by pressing the 'Enter Block' button.
2. Combine K, G_1 and k_3 into an equivalent block, G_{11} (numbering for composite blocks starts at 10). This is done by first pressing the 'multiply' button to multiply K and G_1 (G_{10}), then the 'fb' button to carry our the internal closed-loop operation (G_{11}).
3. Combine G_{11}, G_2 and k_2 into an equivalent block, G_{13}, as done in (2).
4. Combine G_{11}, G_2 and k_1 into an equivalent block G_{15}, as done in (2).

The resultant block diagram and transfer function are shown in Fig. 13(a). The stability analysis through use of pole-zero diagram of Fig. 13(b) shows the system is now stable, as its poles are in the left-half plane. This is confirmed by viewing the impulse response and step response, as shown in Fig. 14.

Once this is done, and an equivalent transfer function for the closed-loop system is obtained,

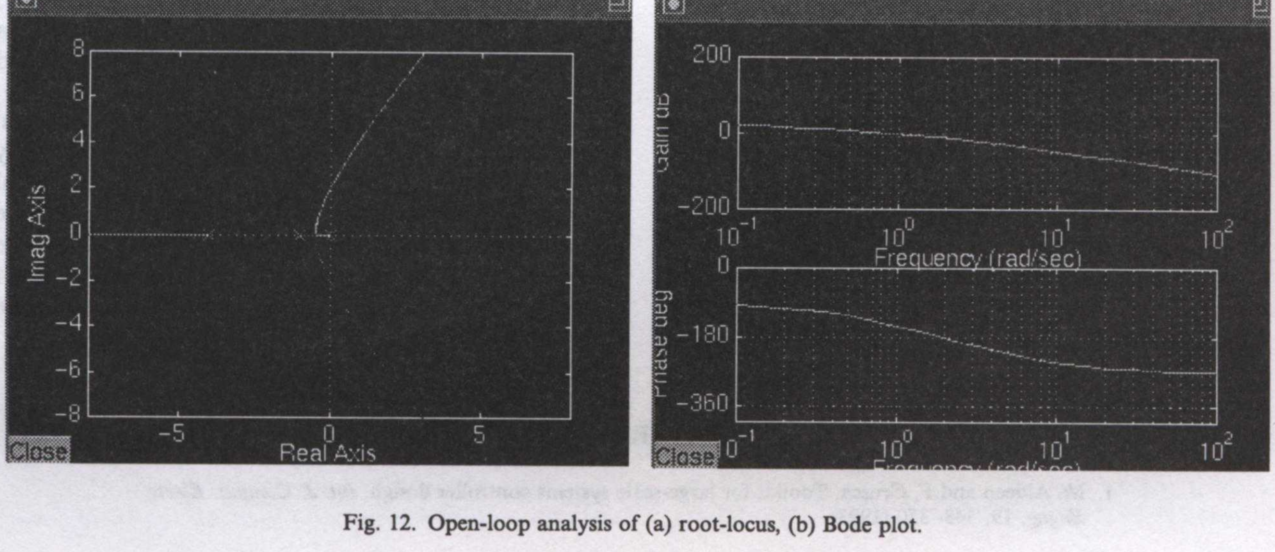


Fig. 12. Open-loop analysis of (a) root-locus, (b) Bode plot.

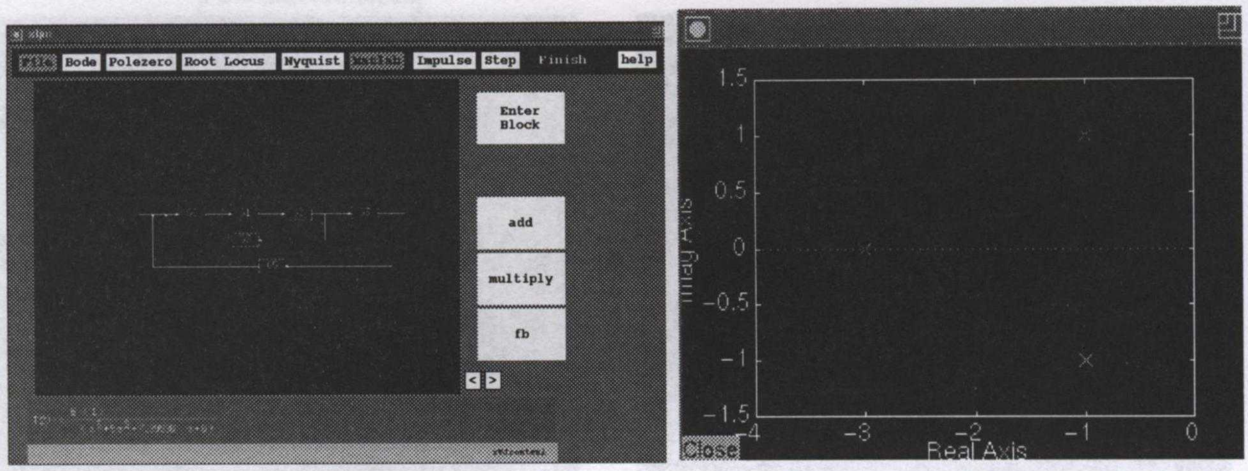


Fig. 13. Closed-loop system: (a) resultant block diagram; (b) pole-zero.

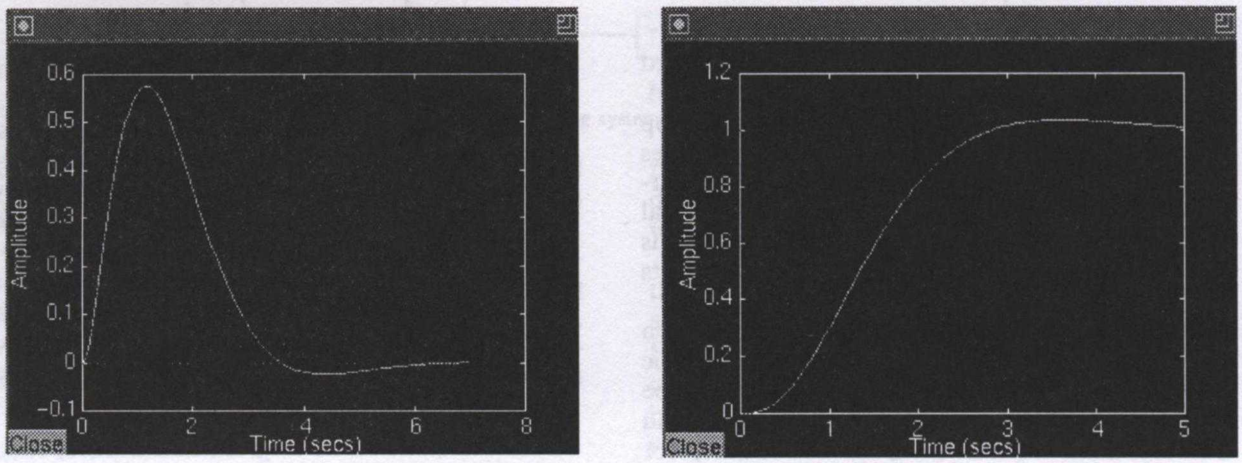


Fig. 14. Closed loop system: (a) impulse response; (b) step response.

local or MATLAB routines can be used for further analysis as described above.

CONCLUSION

In engineering disciplines such as control, networks and communications, transfer function or block diagram analysis constitutes an essential part of the system analysis and design. This paper reports on the development of a software package to facilitate such operations. It is shown that provision of this package presents junior undergraduate students with a simple and user-friendly tool which can be used either in their laboratory sessions or as a learning aid.

The usefulness of this package is demonstrated not only by its excellent transfer function manipulation capability, but also by its other operations such as time- and frequency-domain analysis. These features enable students to free up valuable laboratory time, which would otherwise have been spent on performing tedious and laborious computational tasks, and allows them to concentrate on theory and analysis.

As the software package is interfaceable with MATLAB, it draws on the more advanced and sophisticated operations of MATLAB, providing a wider usage of it by student at a more advanced stage of their studies.

Such a package must enhance the students' understanding of the material the laboratory session seeks to illustrate. As well as its useful usage in the laboratory the package can also be used in tutorial classes and as a learning aid where the students can use it in their own time and at their own pace. Our experience with this package confirms this.

REFERENCES

1. M. Aldeen and F. Crusca, Toolkit for large-scale systems controller design, *Int. J. Comput. Electr. Engng*, **19**, 343–350 (1993).

2. M. Aldeen, Development of a computer-aided experiment in classic control theory, *Int. J. Appl. Engng Ed.*, **8**, no. 2 (1992).
3. M. Aldeen and K. E. Forward, A new electrical engineering course structure in the University of Melbourne, *Int. J. Electr. Engng Ed.*, **29**, 169–181 (1992).
4. M. Aldeen, GPACS based undergraduate teaching laboratory in control engineering, *Australasian J. Engng Ed.*, **2**, 17–27 (1991).
5. M. Aldeen, R. Obbrow and F. Crusca, Computer-based control laboratory, *Proc. 4th AaeE Annual Convention and Conference*, University of Queensland, Australia, pp. 543–548 (1992).
6. F. Crusca, M. Aldeen and P. Bohn, Computer-aided learning facility for power system stability studies, *Proc. 3rd AaeE Annual Convention and Conference*, University of Adelaide, Australia, pp.175–180 (1991).
7. M. Aldeen, D. Lim and J. Lim, A real-time computer software for the measurement and display of phasors, *Proc. Association for Engineering Association in South East Asia and the Pacific (AEESEAP) Conference*, Chistchurch, New Zealand, pp. 429–433 (1991).
8. T. J. Morgan and M. Aldeen, Computer software for transfer function and block-diagram manipulation, *Proc. Int. Conference in Computer Based Learning in Science*, Vienna, Austria, pp. 642–651 (1993).