

A PROTOTYPE COMPUTER-AIDED-LEARNING PACKAGE FOR SYSTEM DYNAMICS AND CONTROL

J.V. RINGWOOD and M.F. COLLINS

School of Electronic Engineering, Dublin City University, Glasnevin, Dublin 9, Ireland

Abstract. This paper reports on the development of a CAL package, offering undergraduate students a computer-based supplemental learning facility in the areas of system dynamics and control and providing a user-friendly entry to computer-aided dynamical system analysis. The prototype package has been evaluated by a mixed set of users with positive results.

Keywords. Computer-aided instruction, computer-aided design, system identification, stability, state space.

1. INTRODUCTION

The successful teaching of techniques in the analysis and control of systems involves a considerable amount of problem solving and real world examples. In many cases, setting problems and exercises which are tractable for the student requires reducing the problems to almost trivial proportions. With the advent of user-friendly computer-aided-design (CAD) tools for control systems, more difficult and meaningful problems may be set, given that the student may use these tools in the problem solution. This however, means that the student must master the CAD package (possibly a useful by-product in itself), but does not really lend itself to the exam-type situation, where even if the package is available, examination performance may depend to a considerable degree on the familiarity and aptitude with the CAD package, rather than the subject matter itself.

The computer-aided-learning (CAL) philosophy provides an effective mechanism for integrating powerful control systems CAD tools (such as the MATLAB/SIMULINK suite) into a user-friendly environment which focuses the student's learning in a specific area or set of areas. Not only does this provide a self-paced study facility, but also uses the computing tools to their best possible advantages in a learning environment. Traditionally, CAL packages have been little more than electronic textbooks, where control of the pace of learning is divided between the instructor and the student. The philosophy adopted in the development of this package is to give much greater flexibility in how the CAL package can be integrated with existing courses and examination procedures.

Needless to say, the production of a professional CAL package is a time consuming undertaking. To evaluate the idea of a CAL package based around a CAD tool, a prototype package was developed, which implements a subset of common systems analysis and control techniques in a CAL environment. The package is based around the MATLAB/SIMULINK suite and uses VISUAL BASIC for the front end, allowing programming within the WINDOWS environment. A variety of mechanisms may be used to transfer data (both graphical and textual) between applications.

2. CAL PACKAGE SPECIFICATION

In order to provide a package which would closely address the needs of students, while providing a teaching environment which is appropriate for various topics, a scientifically administered questionnaire was completed by 3rd and 4th Year undergraduate students, postgraduate students and course lecturers. This questionnaire attempted to determine the most appropriate set of topics to incorporate in the package as well as to get a general feeling as to what features users would like to see in the package. Responses to the multiple choice part of the questionnaire were weighted with regard to the level of agreement or disagreement and also weighted with respect to category of respondent. The questionnaire responses, along with practical and appropriateness considerations, led to the following choice of topics for inclusion in the package:

- State-space <--> transfer function conversion

- Stability analysis
- System identification.

These would be sufficient to evaluate the effectiveness of the package format. Most of the qualitative comments on the questionnaire favoured the adoption of a friendly, colourful and exciting user interface, with varying levels of detail and easy movement throughout the package.

3. SOFTWARE ENVIRONMENT

A number of components were required to form the integrated CAL package. They include:

- A Graphical/textual front end
- A Computational engine
- A Data exchange mechanism

Each of these important components is discussed in the following subsections.

3.1 The Graphical / Textual Front End

This component dictates the overall effectiveness and impact of the CAL package. Several possible choices for this component exist, from basic high-level programming languages to courseware development tools. At the highest level, DOS-based courseware development tools, such as Tencore and SAM were evaluated, but found to be very restrictive in the format of material which could be presented. Their operation conforms to the general "electronic textbook" idea.

At the bottom end, high level languages such as Pascal and C were considered. While newer versions (particularly those which are Windows-based) provide some useful graphical and presentation tools, they basically concentrate on detailed processing and computational functions. A very useful compromise was found in Visual Basic for Windows, which provides the necessary logical and procedural structure for program development, while giving the programmer excellent control over the user interface. This is achieved through the provision of a number of powerful display objects, such as command buttons, text boxes, scroll bars, etc. In addition, Visual Basic has especially effective facilities for the development of forms (screens), can invoke other applications, such as CAD packages, and is event driven.

Visual Basic has some limitations, including some of the structural restrictions associated with Basic and the fact that it is essentially an interpreted language, requiring a Dynamic Link Library (DLL) in order to run. However, overall, Visual Basic was found to be the most suitable environment in which to develop the CAL package.

3.2 The Computational Engine

The MATLAB/SIMULINK software suite was chosen for this function for a number of reasons:

- Many students already have some familiarity with this CAD environment
- It can run on a PC platform
- It has a wide variety of numerically sound mathematical (including matrix) functions
- It has associated toolboxes for special functions, including control systems
- It is extensible and customisable using the M-file facility.

In addition, the MATLAB/SIMULINK suite is available as a Windows-based application, (similar to Visual Basic) allowing it to take advantage of the data exchange mechanisms of Windows.

3.4 The Data Exchange Mechanism

Windows supports two mechanisms for data exchange - the Clipboard and Dynamic Data Exchange (DDE). This allows the two applications to communicate and transfer data, both numerical and graphical. In addition, Visual Basic can be used to send keystrokes to the active Windows application (e.g. MATLAB), as if they were typed at the keyboard, allowing it to control another application just like a user typing at the keyboard. A third communication mechanism utilises file sharing.

4. PACKAGE STRUCTURE

The package is hierarchical in structure, as seen in Fig.1. The package is also modular, each module forming a subtopic containing, the components shown in Fig.1. In addition to the information contained in each section, the user has access to a custom built pop-up calculator and various other section specific pop-up dialog boxes.

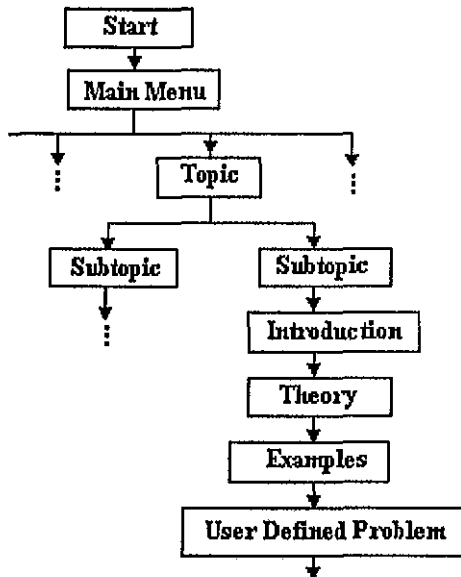


Fig.1. Overall package structure

4.1 Introduction

This section introduces the concept and, ideally, gives some real life examples of the topic area. Such examples may be illustrated with pictures, text, graphs and even animation. An example of the animation used to illustrate the concept of stability is shown in Fig.2. The variation in the ball position with time in response to an impulse (a kick) is shown by a series of pictures (bitmaps).

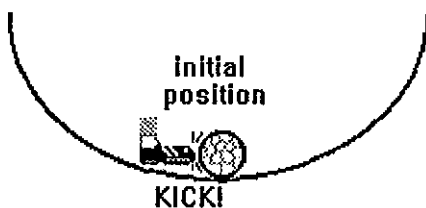


Fig.2. Example of animation used

4.2 Theory

This section presents any theoretical analysis necessary to support the topic. The presentation of this material follows the line of more traditional CAL packages, with the presentation of material in textbook format, with liberal use of colour and graphics and the facility to easily to page forwards or backwards through the material.

4.3 Examples

In this section, the user is presented with a series of predefined examples. Considerable

interaction with the user is invoked and the user required to answer certain questions. To assist with these answers, the user constantly has the facility to review the theory in the previous section. A typical example is shown in Fig.3.

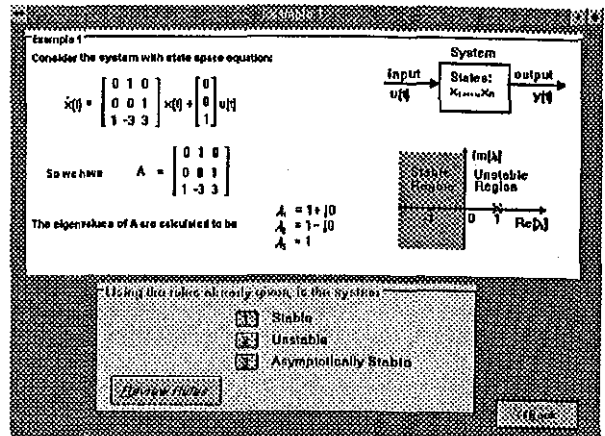


Fig.3. Sample predefined example

4.4 User-Defined Problems

This is similar in format to the previous section, with the exception that the example is defined by the user. This could come from a classroom problem, a standard problem sheet, a problem from a book or simply a problem of the user's definition. The advantage of solving a problem in this environment is that all the computational drudgery can be taken out of the solution, allowing the user to focus on the procedural and logical aspects of the problem. An example of a user defined state-space stability problem is shown in Fig.4.

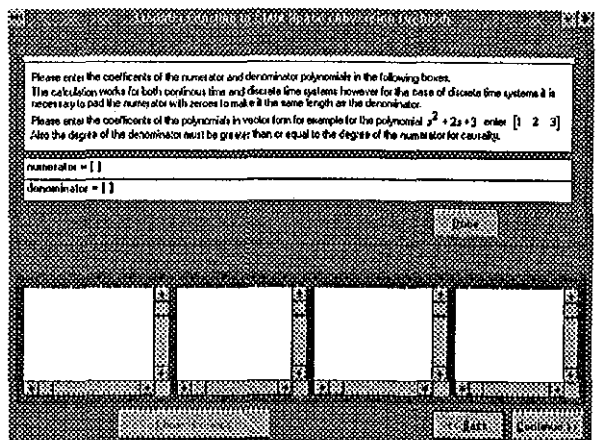


Fig.4. Sample user-defined problem

Upon completion of this section, the user is returned to the module directory and the completed section marked as completed.

5. CAL PACKAGE EVALUATION

The effectiveness of the CAL package prototype was evaluated by approx. 15 respondents, including undergraduate and postgraduate students and academic staff. The responses are summarised in Table 1, with the following key to the point average:

1 - Strongly disagree 2 - Disagree
3 - Unsure 4 - Agree 5 - Strongly agree

My understanding of Systems was improved	3.7
The sequence of content was well ordered	4.4
It was easy to step through the package	4.6
The theory was covered in sufficient depth	3.7
The predefined examples were helpful	4.4
The user-defined examples were helpful	4.4
The package has no advantages over lecture notes	2.0
The screen layout was too cluttered	2.4
The use of colour was appropriate	3.8
It was possible to go through the package without having to really think about the topic in question	2.8
The content maintained my interest	3.7
The package assumed too much prior knowledge	2.5
The mathematical derivations were clear	3.6
Difficult topics were made more understandable	4.3
I would recommend this package to other students	4.5

Table 1 Appraisal of CAL Package

Qualitative comments were also solicited. The positive comments focused mainly on the professional-looking user interface, modularity, use of pictures and animation and interaction with the user. The negative comments focused mainly on the sluggish responses from MATLAB and its non-transparent operation. This was due to the fact that MATLAB did not support DDE and communication had to be performed via the clipboard. This can hopefully be rectified in future versions.

6. FUTURE DIRECTIONS

The development of the CAL package into a full professional system which encompasses a considerable variety of modules will be a time consuming process, possibly requiring a number of man years. However, due to the modular nature of the package, this can be achieved in incremental fashion, providing a usable tool on the way. The most important aspect, at this stage, is to get the definition right so that a consistent approach may be made to each module. To this end, and in the

light of the prototype evaluation, several improvements may be made:

- The screen layouts should be standard across modules.
- Theory sections should involve more compulsory interaction with the user, preventing any glossing over.
- The graphical capabilities of MATLAB should be used to replace stored bitmaps. This would reduce storage requirements and give a more dynamic presentation.
- Ideally, animation should be coded in software, rather than presented as a succession of bitmaps - more complex to code but reducing the storage requirements by approx. 5 Mb per set.
- Though the package itself contains sufficient error checking, error checking for correct MATLAB input format should also be made.
- Visual Basic Version 2 is 20% faster and has significant extra visual features.
- An on-line help facility should be provided.
- A scoring mechanism should be provided. This could be used for giving a performance measure to the student, with a print-out of performance breakdown, or for examination purposes.

7. CONCLUSIONS

The prototype CAL package, though implementing just a small subset of modules, has proved to be an effective mechanism for teaching Linear Systems Theory and Control Systems proper. These subjects are ideal for such an educational medium due to their reliance on a number of graphical constructs and their connection with real world systems. The important aspects of the prototype are the *interaction* which the user is forced to have with the package and the *removal of tedious calculations* in the solution of non-trivial problems, which may represent real-world systems.

The software environment, although not specifically designed for CAL, would appear to be eminently suitable for this application. Another option would be the addition of a package which would allow analytical manipulation, such as Mathematica. However, the package overall should be as self-contained as possible with a reasonable memory requirement.