Teaching Systems and Controls Using a MATLAB-Based Interactive Environment

Mark F. DeHerrera * Armando A. Rodriguez † Richard P. Metzger, Jr. ‡

Department of Electrical Engineering
Center for System Science and Engineering
Arizona State University
Tempe, AZ 85287-7606

October 1, 1996

Keywords: Interactive computer-aided lesson, ICAL, Lesson development, MATLAB.

Abstract

This educational research endeavor describes the development of a MATLAB/Windows/C++ PC environment designed to teach fundamental concepts from classical systems and controls. Interactive computer-aided-lessons (ICALs) form the heart of the environment. In addition to describing the main features of the environment and the process by which interactive multimedia lessons are created, several lessons are presented to demonstrate the environment's utility as an educational tool.

1 Introduction

This paper is motivated by a need to provide engineering and mathematics instructors with a tool to quickly and easily create interactive computer-aided lessons (ICALs) for their sophomore, junior, and senior level students. With MATLAB as its computational engine, the proposed environment is especially geared toward teaching fundamental systems and controls concepts. While MATLAB has proven itself as an invaluable tool for analyzing and designing systems, and while it provides an excellent environment for exploration, it does not provide a guide to the theory of systems and controls [5], [8], [11]. Although such guides do exist [3], [6], [14], [16], they typically come with a set of problems that force the student to use MATLAB and develop very specific MATLAB-utilization skills. In this paper, a different approach is taken. The paper describes a MATLAB-based environment which uses multimedia ICALs to demonstate and reinforce fundamental concepts. Such an environment, if properly designed and used, accelerates and enhances the learning process. ICALs are beneficial for a variety of well known reasons [1], [4]:

- ICALs are good for individualized instruction,
- ICALs permit students to set their own pace,
- ICALs facilitate customization of material,
- ICALs are particularly beneficial in situations that involve students with diverse backgrounds or ability levels.

The remainder of this paper is organized as follows. Section 2 describes the ICAL environment structures and features. In Section 3, the ICAL development process is discussed. Section 4 then demonstrates the utility of the environment as an educational tool. Finally, Section 5 summarizes the paper and presents directions for future research.

*Currently completing a Bachelor of Science in EE.

†This research has been supported, in part, by the National Science Foundation (NSF) through the Coalition to Increase Minority Degrees (CIMD), the ASU Center for Innonvation in Engineering Education (CIEE), the Boeing A.D. Welliver Faculty Fellowship Program, and the Intel Corporation. For additional information, please contact aar@asu.edu.

‡Currently completing a Master of Science in EE.

2 Description of Environment

Upon entering the environment, a master menu is provided from which the user can select a starting ICAL. The user is then led through a sequence of ICALs - each carefully designed to teach a specific concept. Logical sequence of interactive multimedia screens make up each lesson. Although other structures are readily accommodated, each screen is divided into three *child windows*:

1. a Goal Window,
2. an Introduction, Question, Answer, Explanation (IQAE) Window, and
3. a Graphics Window.

Each window may accommodate a variety of media. This includes, for example, text, graphics, animation, video, and audio. Each window is now described.

Goal Window. The main purpose of the goal window is to provide a clear statement describing the goal or objective of the current lesson. Within this window, a block diagram - for eaxmple - may be used to describe the system under study; e.g. a first order stable linear time invariant (LTI) system.

Introduction, Question, Answer, Explanation (IQAE) Window. Within the IQAE window, concepts are introduced and questions are asked. This, to an extent, is done as in traditional texts - only with multimedia. The user specifies answers via interactive menus. If the user specifies an incorrect answer, an explanation may be provided and the user is given another chance. When the correct answer is given, a detailed explanation is displayed.

Graphics Window. Graphics displayed in this window may be part of an introduction to a question or used to shed light on an incorrect answer given by the user. Graphics are also used to support explanations provided when correct answers are specified.

Animation, video, and audio may also be used to support material at any stage of the lesson. While the existing environment is based on MATLAB 4.2 and and runs under any Windows operating system, current developments are focussing on MATLAB 5.0, Microsoft Visual C++, and Windows 95/NT. The ongoing developments significantly impact the lesson creation process. This process is now described.

3 Creation of ICALs

While ICALs are easy to create in theory, in practice this creation process may be very tedious [1]. During the past year, we have been working on facilitating the creation of ICALs so that an entire lesson may be designed, implemented, and refined rapidly. These developments - which are still evolving - are now discussed. Within this discussion, two environments are described: an old environment and a new evolving environment.

The Old Environment. Within the old environment, lessons are implemented within MATLAB macros - C++ is not used at all. Because of this, the lesson creation process is very tedious, time consuming, and simply unacceptable for developing an ICAL-based environment. While MATLAB offers multimedia object management capabilities, they are not very flexible. Video and animation, for example, are played within Windows but external to the MATLAB environment. Although MATLAB provides a state-of-the-art computation engine, its text creation, manipulation, and interface design facilities are weak. These defficiencies motivated the development of a new environment - one which exploits MATLAB 5.0, Microsoft Visual C++, and Windows 95/NT.

The New Evolving Environment. ICALs within the new evolving environment are constructed using a *Lesson Development Interface (LDI)* written for Windows 95/NT in Microsoft Visual C++. Microsoft Visual C++ will facilitate object-oriented programming. More specifically, it will provide us with Microsoft Foundation Classes (MFCs) as well as an OLE capability. MFCs facilitate the development of the LDI. The OLE capability facilitates the manipulation and management of objects.

The user-friendly windows-style LDI allows the lesson designer to very easily manipulate multimedia objects (i.e. text, graphics, animation, video, and audio) in accordance with a lesson plan. Creating these objects is a critical step, manipulating them is easy [8], [10], [11] - especially with the LDI. The following discusses the creation of multimedia objects.

Creation of Multimedia Objects. The text portion of a lesson may be developed using a typesetter such as LATEX[1] or a word processor such as MS Word. An image of the text is then "captured" using, for example, CorelCapture. Graphics may then

[1] LATEX easily handles equations.

be added in CorelDraw.[2] The resulting object is then saved as a compressed GIF file[3]. Graphics may be created in much the same way, using any Windows-compatible drawing utility capable of saving files in the GIF format. Audio files may be created using a microphone and saved as a WAV file. Video from a camcorder or VCR may be imported using a video capture card and saved as an AVI [4] file. These files are then organized around a lesson plan using the *lesson development interface*.

Lesson Development Interface (LDI). A lesson plan consists of a sequence of lessons. Each lesson consists of several interactive multimedia screens. Each screen contains the basic child windows discussed earlier. The LDI provides structures to facilitate the organization and manipulation of lessons, screens, and the multimedia objects which make them up. These structures include: a lesson list dialog box, a screen list dialog box, and a screen edit dialog box. Each structure is now described.

Lesson List Dialog Box: The Lesson Plan. A lesson plan, or sequence of lessons, may be specified using the lesson list dialog box shown in Figure 1. This dialog box may be used to specify the name and order of individual lessons. Additionally, this dialog box may be used to select existing lessons for editing, alter the lesson sequence, and create new lessons.

Screen List Dialog Box: The Lesson. The screens which make up a lesson are specified using the screen list dialog box shown in Figure 2. This dialog box defines the number of screens, order of screens, and screen names which make up a particular lesson. Additionally, this dialog box may be used to select existing screens for editing and creating new screens.

Screen Edit Dialog Box: Child Windows and Multimedia Objects. Screens are edited with the screen edit dialog box shown in Figure 3. This dialog box may be used to edit the format and content of a screen's individual child windows. This dialog box contains three important sections (see Figure 3): a screen format section, a goal window options section, a Q/A window option section, and a graphics window section.

[2]CorelCapture and CorelDraw are trademarks of Corel Corp. CorelDraw is a general purpose drawing/graphics manipulation utility.

[3]The Graphics Interchange Format (GIF) is a platform independent graphics format.

[4]The Audio Video Interleave format for digital video files.

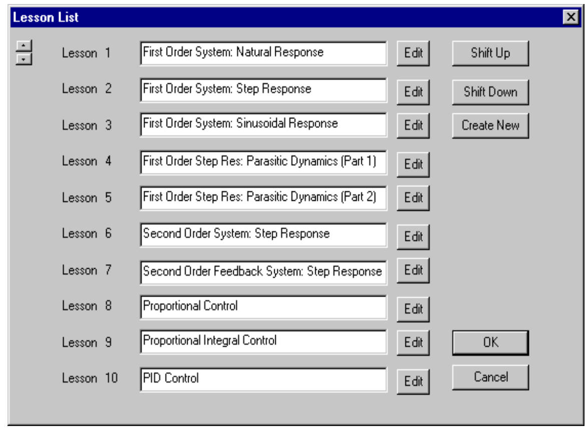

Figure 1: Lesson List Dialog Box

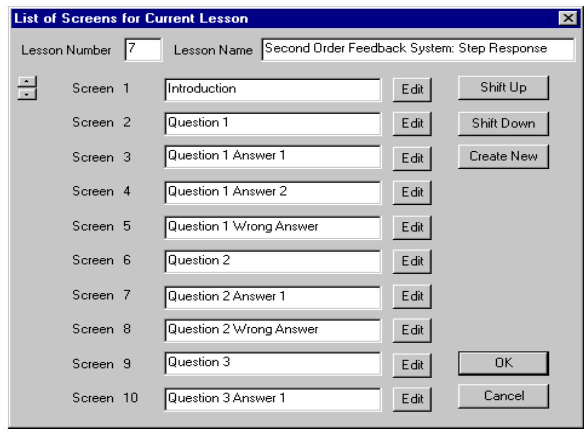

Figure 2: Screen List Dialog Box

Each of these are now discussed.

- The screen format section is used to specify the position of individual child windows within a particular screen.
- The goal window options sections is used to select pictures, captured text, or animation files for display.
- The Q/A window options section is used to specify a picture file for retrieval, specify a captured text file for retrieval, define multiple choice questions, and specify interactivity links with other files (e.g. audio, picture, animation, video, etc.) for questions, answers, explanations, or additional hints.
- The graphics window section is used for specifing a picture file for display or a MATLAB macro to be processed. The MATLAB macro may

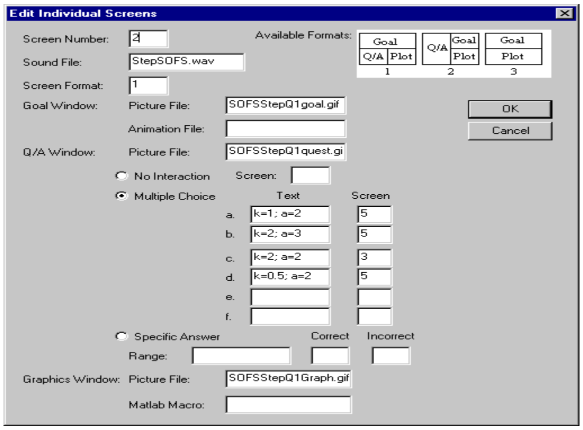

Figure 3: Screen Content and Structure Dialog Box

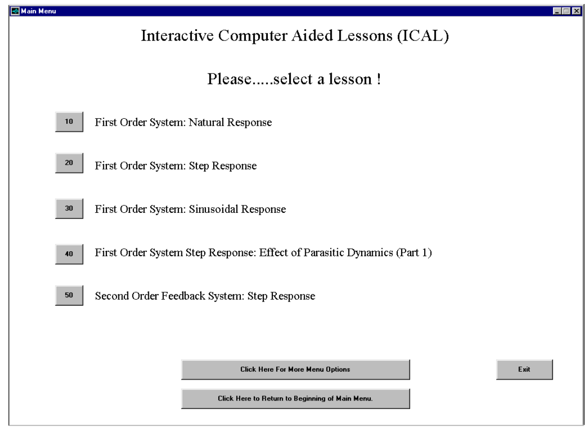

Figure 4: Opening Menu - Lesson Selection Screen

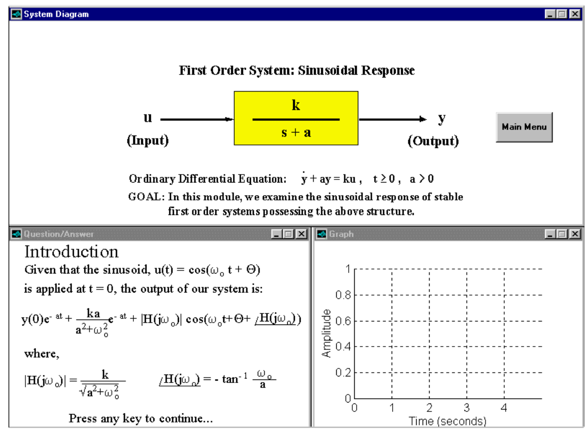

Figure 5: Lesson 3 - Sinusoidal Repsonse of First Order Systems Introduction - Window with System Diagram

perform MATLAB command functions such as the generation of data plots using user-specified parameters in response to lesson questions.

4 Educational Utility

In this section, several ICAL screens are provided to demonstrate the utility of the ICAL environment as a useful educational tool. While these ICAL screens were developed within the old environment, they will be used as templates for future development. Figure 4 shows an opening menu with a few of the lessons that are currently available.

Figures 5-7 consider the sinusoidal response of stable first order systems. Figure 5 depicts an introduction to the subject and displays a block diagram which describes the system under consideration. Figure 6 shows a window that asks the first question on the subject and provides an edit box for user responses. If the user enters an incorrect answer, a window appears that explains why the answer was incorrect and shows a graph of the incorrect response. Figure 7 appears when the user specifies the correct answer. The answer window explains why the response is correct and reinforces the explanation with a graph.

Figure 8 shows the first question in a lesson dealing with the step response of a second order feedback system. A set of multiple-choice buttons are provided for user input. A hint button is also provided which will bring up a screen of useful information pertaining to solving the problem. If a wrong answer is selected, a window appears showing why it was incorrect, as in Figure 9. Figure 10 and Figure 11 step through an explanation of the result after the user selects the correct answer. Figure 11 also demonstrates what happens when the gain of the second order feedback system is increased.

5 Summary and Directions

In this paper, a MATLAB-based ICAL multimedia environment with structures for facilitating the development of lessons was described. The environment is based on MATLAB 5.0, Microsoft Visual C++, and Windows 95/NT. While the environment may ostensibly be used to develop lessons for any subject, it is especially geared toward the teaching of systems and controls concepts. Used properly, ICALs are an excellent supplement to traditional teaching techniques [4]. They not only produce an enjoyable learning environment that can hold a student's attention, but also reinforce concepts with a combination of audio and visual stimuli.

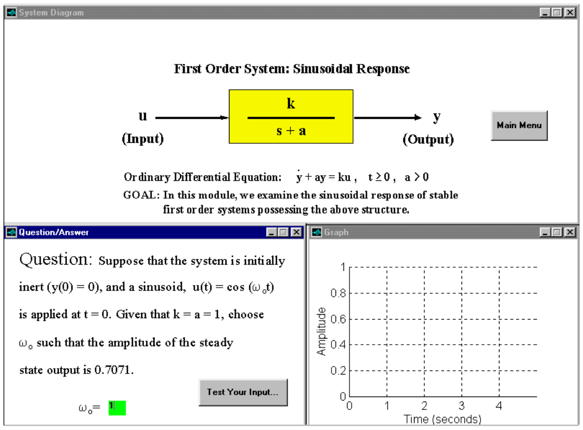

Figure 6: Lesson 3, Question 1 - Question Window

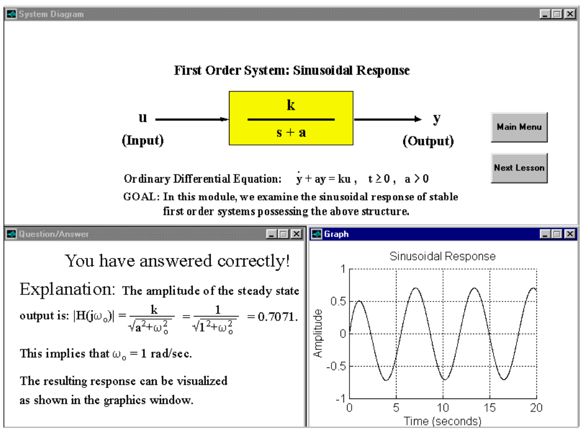

Figure 7: Lesson 3, Question 1 - Answer Window with Accompanying Graphics

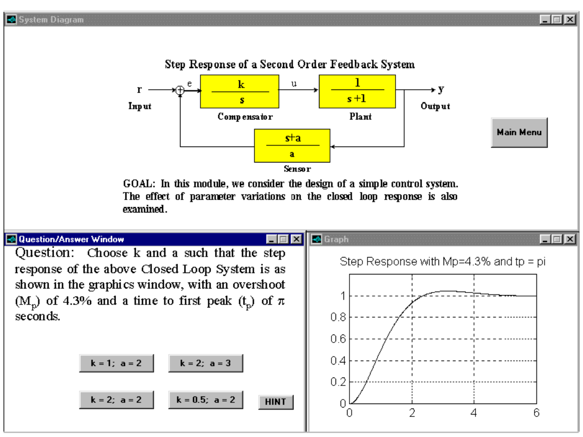

Figure 8: Lesson 5, Question 1 - Step Repsonse of Second Order Feedback System - Question Window with Diagram and Graphics

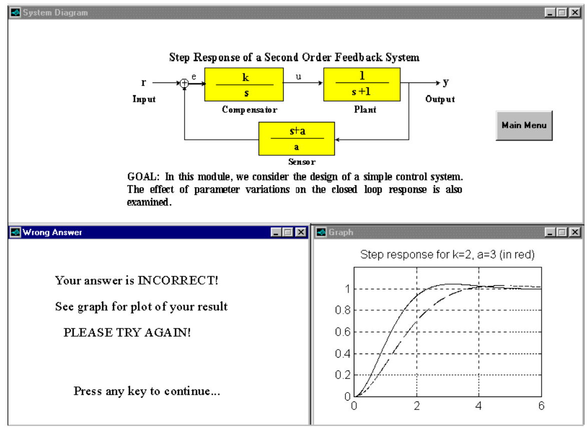

Figure 9: Lesson 5, Question 1 - Incorrect Answer Window with Grpahical Visualization

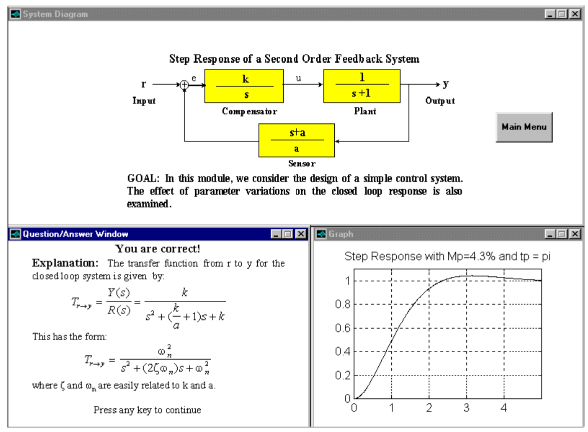

Figure 10: Lesson 5, Question1 - Answer Window 1 with Explanation and Graph of Result

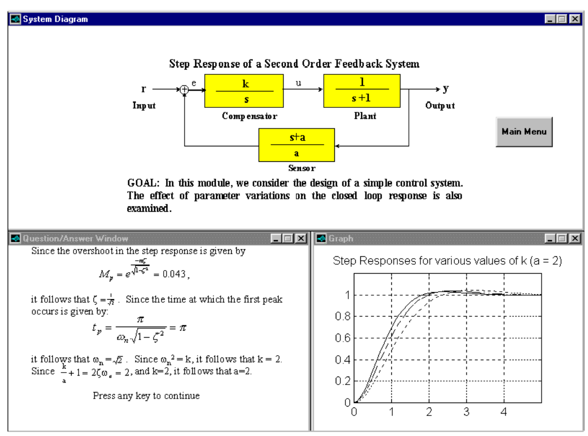

Figure 11: Lesson 5, Question1 - Answer Window 2 with Explanation and Graph Demonstrating Concept

The plan is to soon have a database of over 300 lessons for teaching systems and controls. Coding the interface in C++ allows easy construction of an intuitive Windows environment, while linking the environment to MATLAB allows it to take advantage of the powerful mathematical manipulation functions and graphing abilities that MATLAB has to offer.

References

[1] B.A.A. Antao, A.J. Brodersen, J.R. Bourne, and J.R. Cantwell, "Building Intelligent Tutorial Systems for Teaching Simulation in Engineering Education," IEEE Transactions on Education, Vol. 35, No. 1, February 1992, pp. 50-56.

[2] R. Braham, "Math and visualization: New Tools, New Frontiers," *IEEE Spectrum,* November 1992, pp. 17-36.

[3] C. Borghesani, *Controls Tutor Using MATLAB*, PWS, 1996.

[4] S.R. Cvetkovic, R.J.A. Seebold, K.N. Bateson, and V.K. Okretic, "CAL Programs Developed in Advanced Programming Environments for Teaching Electrical Engineering," *IEEE Transactions on Education*, vol. 37, no. 2, May 1994, pp. 221-227.

[5] D. Etter, *Introduction to MATLAB for Engineers and Scientists*, Prentice Hall, 1996.

[6] D.K. Frederick and J.H. Chow, *Feedback Control Problems Using MATLAB and the Control Systems Toolbox*, PWS, 1995.

[7] L. Geppert, "The New Contenders," *IEEE SPECTRUM: Fast and Powerful*, December 1993.

[8] D. Hanselman and B. Littlefield, *Mastering MATLAB*, Prentice Hall, 1996.

[9] L.B. Jackson, *Signals, Systems, and Transforms*, Addison-Wesley Publishing Company, Reading, MA, 1991.

[10] *MATLAB Based Books*, MathWorks Inc., Natick, MA, Spring 1996, www.mathworks.com, info@mathworks.com.

[11] *MATLAB User's Guide*, MathWorks Inc., Natick, MA, 1985.

[12] R.P. Metzger, K.J. Elliott, and A.A. Rodriguez, "Modelling, Analysis, and Graphical Visualization of a Dual Robot Arm System: A PC BasedEnvironment," *Proceedings of the 1996 International Conference on Simulation in Engineering Education*, San Diego, CA, January 14-17, 1996, pp. 175-180.

[13] K. Ogata, *Modern Control Engineering*, Prentice Hall, Englewod Cliffs, NJ, 1990.

[14] J.C. Polking, *Ordinary Differential Equations Using MATLAB*, Prentice Hall, 1995

[15] C. E. Rohrs, J. L. Melsa, and D. G. Schultz, *Linear Control Systems*, McGraw Hill Inc., New York, NY, 1993.

[16] B.C. Wheeler, "MATLAB Exercises to Demonstrate Principles of Biological Modeling and Signal Processing," *ASEE Annual Conference Proceedings,* Vol. 2, Session 3509, 1993, pp. 1789-1790.