

A Case Study in Software Instruction over the World Wide Web: The Michigan-CMU Control Tutorials for Matlab

Dawn Tilbury
Mechanical Engineering
and Applied Mechanics
University of Michigan
Ann Arbor, MI 48109
tilbury@umich.edu

William Messner
Mechanical Engineering
Carnegie Mellon University
Pittsburgh, PA 15213
bmessner@andrew.cmu.edu

Abstract

The World Wide Web (WWW) has given educators unprecedented opportunities to provide information to students both within their own classes and around the globe. This paper describes the development and implementation of a set of Web-based tutorials for Matlab, a popular computation and visualization software package. The potential impact of the WWW on pedagogical objectives is discussed, and the tutorials are presented as a case study of Web-based software instruction. Their structure and implementation are described, and feedback and statistics from the first semester of use at the University of Michigan are shown. We also describe our efforts at disseminating information on the tutorials to increase their use. The paper concludes with a discussion of future work planned for the tutorials and potential future applications of the concept of Web-based tutorials.

1 Introduction

Engineering education (as well as the engineering profession in general) has become increasingly reliant on ever-more-powerful software tools to assist in solving ever-more-complex problems. Computations which once took pages of algebra now require only a press of the ‘return’ key. Instead of presenting computational techniques in the classroom, it is now possible for the professor to focus on both fundamental and advanced *concepts* and let the software packages do the computation.

Teaching students to effectively use these new software tools can be problematic, however. Software is best learned by “playing” with it, but the user interfaces to many powerful tools are cryptic at best. The manuals are often written for advanced users and can be difficult for beginning engineering students to follow. The professor often will bring a computer into the classroom to demonstrate the software, but it can be difficult to take notes on the commands that are used.

In this paper, we present a new methodology for teaching software tools. We have developed a set of tutorials for teaching students to use Matlab, a popular computational software package, in the context of Automatic Control.¹ Student users are expected to run Matlab in one window of their computer and a Web browser (such as Netscape) in another. The tutorials contain Matlab code along with explanatory text to describe the objective of the commands along with the Matlab output (text or graphics). The students are encouraged to copy the commands from the browser window into Matlab (using copy and paste with the mouse), run the code, and compare the output they get with that given in the tutorials. In this way, they are able to quickly use the information presented to them.

Because the tutorials are on the World Wide Web, they are immediately available to students whenever they are at a computer with a network connection. There is never the danger that they forget their handout at home, the manual is already checked out of the library, or the notes taken in class have been eaten by the dog.

Throughout the tutorials, we assume that the users are currently taking a course on Automatic Control. The theoretical treatment of the subject matter is cursory; the emphasis is on how to use a software tool (Matlab) to analyze and design control systems. With this design, the tutorials can not only be easily integrated into an existing undergraduate controls curriculum, but they can also be used by community colleges, industry, and individuals for continuing education. With these tutorials, it is easy for engineers in academia and industry with access to the WWW, who know some classical control theory, to learn how to use Matlab to apply their knowledge and to improve their skills and understanding. In fact, our current statistics show that approximately one-fourth of the US users are from companies.

The application here differs somewhat from other uses of the WWW that we have found. Other educators have built Web sites for illustrating concepts and techniques of dynamic systems and controls [9]. Still others have used the Web as a distribution method for reading material [2]. The tutorials described in this paper are used to teach a skill in using a particular software package.

The outline of this paper is as follows. Pedagogical considerations are discussed in Section 2. The structure and content of the tutorials, including unique features of the tutorials, are presented in Section 3. Assessments of the use and utility of the tutorials are discussed in Section 4. Efforts to disseminate the tutorials are described in Section 5, and future work is discussed in Section 6.

2 Pedagogical Considerations

The pedagogical basis on which the tutorials are built is that students learn most quickly when they are presented with material that they can quickly use to solve problems. With these tutorials, students progress rapidly because they “learn by seeing and doing.” Any student with access to both the WWW and Matlab (on either a personal computer or a workstation) is able to follow a tutorial while running Matlab, and can easily switch between the two programs. Commands shown in the tutorials can be copied from the Web browser window and run in the Matlab command window with a simple point and click of the mouse.

¹The tutorials can be accessed on the WWW at <http://www.engin.umich.edu/group/ctm> or <http://www.me.cmu.edu/Matlab/html>.

The elimination of time-consuming and error-prone typing and editing greatly facilitates the learning process by reducing frustration and decreasing the time before something useful can be done. Students can immediately see the result of an actual computation, compare it to the result shown on the tutorial, and quickly experiment with modifications of the commands and changes of parameters. Compared with currently available books for teaching students how to use Matlab for control system design [5, 6, 8, 10], the proportion of time spent learning usage compared to the time spent learning to debug a program is greatly increased with the Web-based tutorials.

Reinforcement and practice are also essential to the learning process. Consequently, the tutorials contain many examples for students to study and compare. In addition to the main tutorial pages, four examples of physical systems are incorporated into the tutorials. For each physical system, modeling and four control design examples are given. These twenty examples are interconnected “vertically” by the physical systems and “horizontally” by control design method.

Another advantage of the tutorials is their interconnectedness, which is easy to implement on the WWW using the hypertext markup language. In contrast to a book, which begs to be read in a linear sequence, the tutorials admit many equally appropriate paths. Depending on her interest and available time, a student may wish to look at one or many tutorial pages addressing one or more topics. Each page stands alone, but offers hyperlinks to related topics.

A map of the tutorials is given in Figure 1. The main flow of the tutorials is from top to bottom; this outline mimics the table of contents of many of the popular textbooks on automatic control [1, 4, 3, 7]. In addition to the main tutorial pages, there are four examples which are followed through the tutorials. For a quick review of classical controls, a student could follow only the tutorials without going to the examples. A student studying root locus might begin at the root locus tutorial, and then follow all four root locus examples. A student interested in the DC motor might start with its model and view all four motor control design examples.

The hypertext markup language on the Web is also well-suited for allowing background and supporting information to be provided in an unobtrusive but readily-available manner. For example, every time a function is used in the tutorials, the word ‘function’ is highlighted; that hyperlink leads to the Web page describing how functions work in Matlab. The complete description of functions, although not included on every page, is never more than a mouse-click away.

Review of previously-seen material is made easy through the use of many hyperlinks and indices. As discussed above, the home page shows the overall structure of the tutorials. At the top of every page is an outline with subheading which are hyperlinks leading to the different regions on the page. There is also an ‘Index’ page which gives an outline view (with hyperlinks) of every tutorial and extra page. At the bottom of every page are links to all of the tutorials as well as links to similar examples. Links are always given to come back to the home page, to the complete index, and to the list of Matlab commands. Within the Web browser, students can set their own bookmarks to return to particularly useful pages of the tutorials.

Essentially, the tutorials provide ready access to information when and where it is needed—at the computer. The generality of these pedagogical principles can be applied

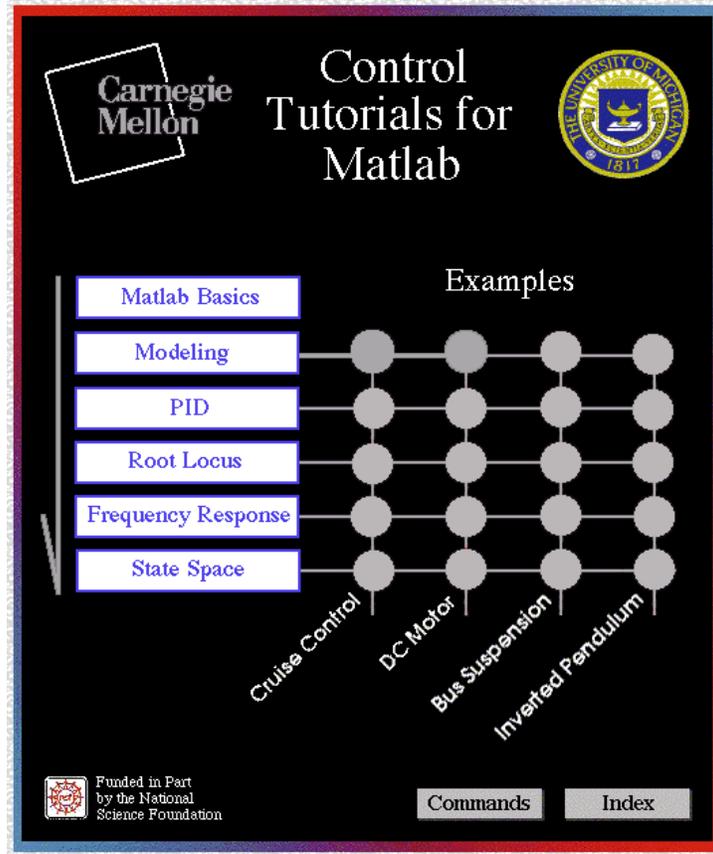


Figure 1: The image map on the home page of the tutorials. A mouse click on the image leads to the corresponding Web page or site. Each tutorial is represented by a box, and the flow of the tutorials is from top to bottom. There are also four examples which are followed through the tutorials (each example page is a dot). The examples are linked vertically by the physical systems and horizontally by control design method. The headers lead to the home pages for the University of Michigan and Carnegie Mellon University. At the bottom are links to the complete index of the tutorials, the list of Matlab commands, and the National Science Foundation, which provided funding for this project.

to not only to other types of software, but also to the use of laboratory equipment. Some possibilities are discussed in Section 6.

3 Structure of the Tutorials

The tutorials are designed to help students learn how to use Matlab for the analysis and design of automatic control systems. They cover the basics of Matlab, the most common classical control design techniques (PID, root locus, and frequency response), as well as some modern (state-space) control. A “page” here refers to one Web page; each page will typically takes up a few screenfuls on a monitor.

The intention is that students will follow along with the tutorials by running Matlab in

one window and the tutorials in another. They should be able to run most of the Matlab programs by copying and pasting between windows. Students in ME461 at the University of Michigan found a prototype set of tutorials helpful as an on-line reference while doing homework assignments using Matlab as well as for reviewing key concepts before exams.

Following is a brief description of the contents of the tutorials.

3.1 Tutorial Pages

The tutorials begin with “Matlab Basics.” As the title implies, this page introduces the Matlab format for vectors and matrices, some standard functions, as well as plotting. Since in controls, transfer functions are often represented as rational polynomials, we show how Matlab uses vectors to represent polynomials. Addition, multiplication, and division of polynomials is also covered.

The modeling tutorial is included for completeness. This page briefly discusses the application of Newton’s laws to mechanical systems, converting equations into state-space form, entering the state-space equations into Matlab, and converting the state-space equations into transfer function form.

The first control tutorial covers the classic three-term PID controller, and describes how to find the open-loop step response of a system, and what effect the three terms (proportional, integral, derivative) will have on the closed-loop behavior. The tutorial also shows how Matlab can be used to find the closed-loop transfer function for unity feedback systems. Examples show how to tune the gains of the PID to get an acceptable response.

The root locus tutorial begins with the definition of the root locus, and the Matlab command for drawing it. It also describes how to choose the proportional gain to achieve a desired closed-loop response, and how Matlab can be used to find the value of the gain. Further topics, such as how to add a lead compensator to improve the characteristics of the root locus, are covered in the examples.

The frequency response tutorial is broken into two parts, one for the Bode plot and another for the Nyquist plot. The gain and phase margin, as well as closed-loop stability, are discussed in each part. The tutorial shows how Matlab can be used to generate and analyze both types of plots, and the fact that the two plots present the same information (the magnitude and phase of the transfer function) in different ways is emphasized.

The final control tutorial covers state-space (modern) control design techniques. Although this topic has not traditionally been included in many undergraduate-level controls courses, the advent of computer-aided control system design using Matlab has made these techniques more accessible. Pole placement using full-state feedback as well as observer design and the reference input are covered.

3.2 Example Pages

In addition to the five main tutorials, there are four examples which reinforce and extend the concepts and techniques introduced in the tutorials. The examples are ordered in terms of increasing complexity, and are all based on physical system models (not abstract equations or transfer functions). The simplest model considered is a cruise-control for an automobile; a DC motor, a bus suspension system, and an inverted pendulum on a cart are also examined.

Throughout the examples, the emphasis is not only on how to use Matlab, but also the physical interpretation of design parameters such as maximum overshoot and steady-state error in the systems under consideration.

For each of the four physical examples, there are five pages corresponding to the five tutorials: modeling, PID, root locus, frequency response, and state-space. The modeling page introduces the system, finds the equations of motion, represents them in Matlab, and defines the control design goals. The four control pages then show how these design goals can be met using the four design techniques.

3.3 “Extra” Pages

There are many topics that needed to be covered in the tutorials that didn’t fit within the framework as defined above. These topics are implemented as “extras;” they are not large enough to deserve their own tutorial, but they are too important to be ignored. Also, many of them relate to more than one of the tutorial topics. The “extra” pages are listed on the main index page, and are linked from the relevant places throughout the tutorials. An attempt is made to take maximum advantage here of the ease of interconnection using the hypertext markup language. For example, there is an extra on lead/lag compensators. Whenever we use a lead or a lag in an example, there is simply a link to the appropriate subheading in the lead/lag extra (i.e. “designing a lag compensator using frequency response”). The steady-state error extra is another popular one; instead of reviewing how to compute the steady-state error in every example, a link is given to this extra. Other extras include plotting, converting between system representations, the step response and simulation, pole/zero cancellation, notch filters, functions, and m-files.

3.4 Implementation Details and Unique Features

Figure 2 shows a small piece of the root locus tutorial. Several things to note are the combinations of explanatory text, Matlab code, hyperlinks, and figures. The convention is to have explanatory prose in a proportional font and Matlab commands in a fixed-width font. The Matlab commands are set off from the text, for ease of copying and pasting into the Matlab window using the mouse. Each tutorial page is divided into several subsections with bold headings; an outline of the subheadings is given at the top of each page and on the index page for easily accessing a certain part of a tutorial.

Graphics are used extensively. Most commonly, graphics are used to show the output of a Matlab command which produces a plot. Students can compare the result that they obtain from running the code with the result shown in the tutorial. If the code is simply copied from the tutorial to the Matlab window without modification, the plots should match. Students can also compare the differences between their results and the tutorial when they change some of the system or control parameters. Graphics are also used to include block diagrams, schematics of physical systems, equations, and other drawings to illustrate concepts, such as the shape of the Nyquist contour in the frequency response tutorial.

Hyperlinks to other pages with more information on the commands show up in the page as underlined and in a different color. The interested student can follow them, but the main body of text is not cluttered with superfluous explanations. At the bottom of every page

Plotting the root locus of a transfer function

Consider an open loop system which has a transfer function of

$$H(s) = \frac{Y(s)}{U(s)} = \frac{s+7}{s(s+5)(s+15)(s+20)}$$

How do we design a feed-back controller for the system by using the root locus method? Say our design criteria are 5% overshoot and 1 second rise time. Make a [Matlab file](#) called rl.m. Enter the transfer function, and the command to plot the root locus:

```
num=[1 7];  
den=conv(conv([1 0],[1 5]),conv([1 15],[1 20]));  
rlocus(num,den)  
axis([-22 3 -15 15])
```

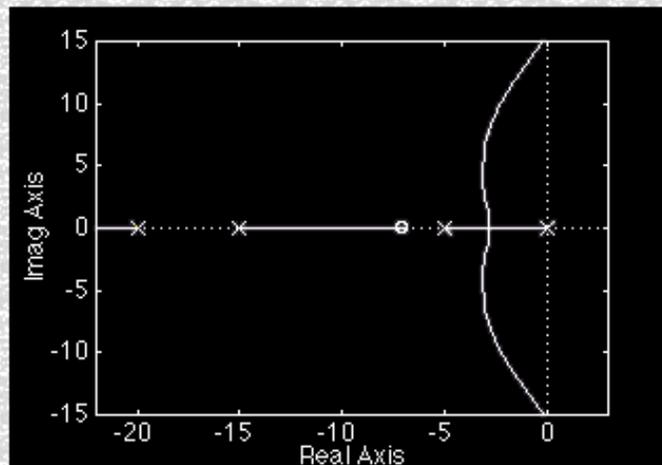


Figure 2: Snapshot from a root locus tutorial. Note the juxtaposition of explanatory text (in proportional font), Matlab code (in fixed font), equations (included as in-line gifs with transparent background), and figures (the figure is in color if viewed on a color monitor). The underlined words (Matlab file, axis) appear on-screen in blue, and are hyperlinks to pages with more information (about m-files and plotting options).

are “horizontal” hyperlinks to all the examples using the techniques covered in the tutorial, “vertical” hyperlinks to the other tutorials, and buttons linking the user to the home page, index page, and command page.

One of the great advantages of using computers as an education medium is that motion can be easily depicted. In the current version of the tutorials, movies are included to illustrate aspects of the Nyquist criterion. The first animation shows the relationship between the polar plot of a the frequency response of a transfer function and the gain and phase of the Bode plot. Another shows how encirclements of -1 are counted on a Nyquist plot. Both of these concepts are difficult to explain in a textbook but are easily illustrated using a movie.

We have designed the tutorials to be extremely portable. Local file references for all links are used, and a client-side image map was used on the home page, so the entire package of tutorials can be transported to another site and installed on their Web server with a minimum of effort. In an effort to retain backwards-compatibility and readability by varied Web browsers, many of the latest functions of html (such as frames) were not used. Realizing that many users have small monitors, the images are as small as was determined to be reasonable. The entire set of tutorials was created by four mechanical engineering undergraduate students, working at the University of Michigan and Carnegie Mellon University over one summer, under the direction of Professors Tilbury and Messner.

4 Assessment

The novelty of these education aids made a means for receiving feedback from users desirable. A feedback form is included at the end of each tutorial; each form contains four multiple-choice questions and a section for free-form comments. The form is submitted (anonymously) by electronic mail to the authors for processing and evaluation. An email address where students can send questions is also given; the users are asked to use this email address if they wish to receive a reply.

Two means are available for the assessment of the utility of the tutorials. The first is direct feedback available from the users through a forms interface described above. The email responses were tabulated and the comments summarized. We also have acquired statistics from the University of Michigan’s Web server, detailing every ‘hit’ (request) for a page of the tutorials. This data was processed with Analog to get summary information on the number of pages accessed, which pages were most popular, and what sites the users came from. A perl program was written to extract the number of pages that each user requested, the time that was spent per page, and the total time spent looking through the tutorials.

4.1 User Feedback

Direct feedback is available from the users through a forms interface at the bottom of every page. During the period Sept. 1996–Jan. 1997, we received 191 email responses; these responses are tabulated in Table 1. Nearly all (181) respondents thought the tutorials were at least somewhat useful. In addition, the overwhelming majority (149) of the respondents felt that the level of the tutorials was “about right.” In their feedback to us, the users were also able to include comments. Of the comments, 46 were kudos or thanks, 13 were

questions (such as “what are dc-motors” from Cyprus), 22 were suggestions for improvement (the most common requests were for digital control and reduced-order observers), 8 were problems encountered (such as printing problems), and 4 were other comments, (including some in a foreign language that we couldn’t understand).

The respondents were from around the country as well as around the world. Comments received from users include (with their locations, as determined by the WWW server, in parentheses):

- Pretty darn nifty (Duke)
- Great tutorial.....!!!!!!! —> THANKS.....
(University of Florida)
- I love this page...especially all the very handy *.m files that you offer. I will certainly visit you very often. (Delft University of Technology)
- Thanks for the tutorial. I needed to learn Matlab basics for a project I was working on and I found the tutorial to be very helpful. I had no Matlab experience so I thought that the tutorial was at the right level. (Lockheed Martin Corporation)
- I am taking a class now on Controls and the examples you provide on this page are as good or better than anything I have found. Also, being able to snap code off the Web page and then run in Matlab is also very useful! (University of Wisconsin-Milwaukee)
- This tutorial was useful in that instead of giving just a random list of commands, it gave stuff to work with which made it more worthwhile. (Carnegie Mellon)
- first of all i would like to take the chance to say how useful and great ur site (matlab tutorials) and i would like to say that it will be grater if you include a tutorial for a reduced order observer (Kuwait)
- Because I have no manual this pages are really a good help for me to get familiar with Matlab (University of Cape Town, South Africa)
- thanks for the tutorial on Matlab basics. something like this was much more useful to me than having to use Matlab’s on-line help reference. (Northwestern)
- I’m currently taking a controls course and I was confused on how the zeros and poles effected the root-locus - your tutorial was very helpful. Thanks... (Virginia Tech)
- This is a very nice initiative and I hope you will be able to expand it. (University of Texas)
- Very good idea to teach using WWW. I hope more people will use your concept. I think the whole course or other courses could be taught in that way. (Quebec)

Industrial representatives to whom we have shown the tutorials have also been very impressed; they see the tutorials as an excellent way to bring their employees up to speed on the latest software tools. For retraining purposes, the Web-based tutorials are much more attractive than books with the equivalent content because they eliminate the need for large amounts of typing.

How useful was this tutorial?	
Very useful	101
Useful	60
Somewhat useful	20
Not very useful	6
A waste of time	4
How was the level of the tutorial?	
Too simple	35
About right	149
Too difficult	7
How much time did you spend on it?	
Less than 30 mins.	133
30-60 mins	44
More than 60 mins.	13
How much of the Matlab code did you run?	
None	77
Some	56
Most	56

Table 1: We received 191 anonymous email responses from the tutorials from September 1996 through January 1997. This questions and answers are summarized here.

4.2 Usage Statistics

When the tutorials went on-line in September of 1996, they were simultaneously installed at the University of Michigan, CMU, and Duke. The tutorials were incorporated into the Automatic Controls course taught in the Electrical Engineering Department at Duke during Fall 1996 by Dr. L. Bushnell. We “advertised” the availability of the tutorials by entering the Web address (at UM) into the most popular Web search engines (Altavista, Lycos and Yahoo).

Access statistics are currently available for the University of Michigan’s Web server for the period Sept. 1996–Feb. 1997. The Web server tracks each request for a page and the domain name of the computer that made the request. We have processed these statistics both by domain (where the users came from) and by page (which pages they looked at). In the statistics, we have excluded requests from the author’s computers at Michigan.

As shown in Figure 3, the traffic continued to grow during the fall semester. Also shown is the breakdown of visitors by their location. As expected, a large portion of the users were from US educational institutions. A surprising number were from foreign countries and US companies. The ‘unresolved’ users are identified only by their numerical addresses; we did not cross-reference them to their domains.

We are also able to break down the number of requests by page; a summary of the total hits during the period September 1996–January 1997 is given in Figure 4. The basics tutorial is by far the most popular; evidently, many of the visitors are not as interested in controls

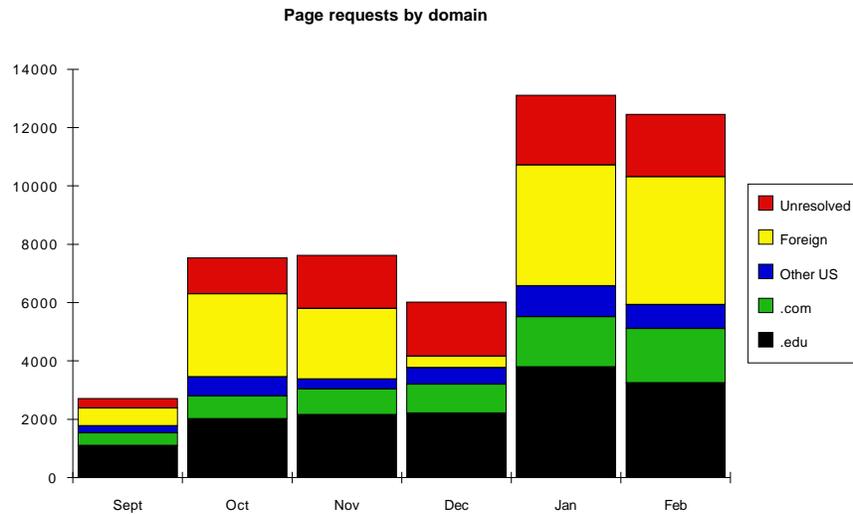


Figure 3: The total number of page requests per month, broken down by domain.

as we are! During this same period of time, the home page was the most popular page, accessed 8358 times, the basics was second-most popular at 3744. Other popular pages not shown in the figure include the list of commands (1196), the complete index (1490), and the 22 “extra” pages (6018 combined)—the most popular extra was the one describing Matlab functions (896).

The Michigan Web server tracks the internet address of the host computer which accesses each page as well as the time of the access. We have used this raw data to estimate the number of users of the tutorials as well as the amount of time that each user spent looking at the tutorials. We assumed that only one user was accessing the tutorials from a particular host at a given time, and that a single user would access a new page at least once every 15 minutes. Using these assumptions, we have summarized the access data in Figures 5–7.

The estimate of the number of users obtained in this fashion is likely to be conservative. If two or more users were working with the tutorials at the same time from the same host, only one user would have been counted. In contrast, the estimates of the time spent per page should be lower than the actual number, since multiple users counted as a single user would be accessing different pages faster than a single user.

Another assumption that we have used in calculating these statistics is that a user was looking at a tutorial page up to the time when he accessed another tutorial page. The Web server can only determine when a user accesses a page; it cannot determine when a user has stopped looking at a page. The time spent on a page was assumed to be equal to the time difference between page accesses. The time spent on the last page accessed by a user could not be determined.

Figure 5 shows the distribution of the total time spent by users on all pages associated with the tutorials including examples and extra pages. The statistics are gleaned from data collected at the University of Michigan server from September 1996 through February 1997. Approximately 13000 distinct users accessed the tutorials during this time. Time statistics could be determined for 8800 of these users. The number of people who spent 1 minute or less on the tutorials seems high, but it is important to remember that this includes many

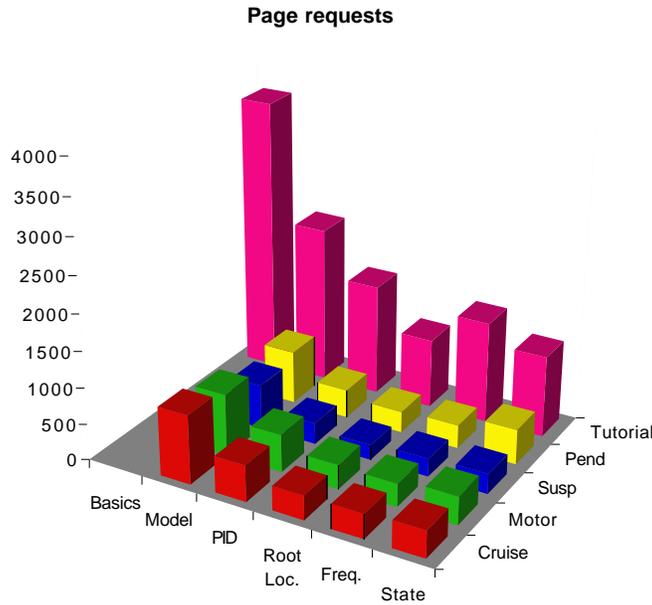


Figure 4: The total number of hits per page over the total period September 1996–January 1997, broken down by page.

Web surfers as well as automatic Web crawlers who just looked at a page or two briefly before moving on. It is encouraging to note that of the 8800 total timed users, about 2000 spent 10 or more minutes examining or using the tutorials. The average total time spent in the tutorials was just over 5 minutes.

Figure 6 shows the average amount of time spent per page for the main tutorial and example pages. The average time per page for all pages was 159 seconds. Figure 7a shows that the basics page was by far the most popular tutorial, followed by modeling, PID, frequency response, root locus, and state space. The mode of all of the distributions was between one and two minutes. One interpretation of the distribution data is that most people used the tutorial just as a quick reference and not as an intensive study guide. Another explanation for the short time duration is that users would leave tutorial pages to examine the examples or extras and would return later. Evidence to support this hypothesis is supplied by the distribution of time spent on the most popular “extra” pages shown in 7b. These pages have similar distributions to the those for the main tutorial pages, yet five of the six most popular extra pages contain only brief explanations of particular Matlab functions. Users would be expected to go in and out of an extra page in a relatively short time—just long enough to get a small amount of reference information.

5 Dissemination

A new education technique or a technology needs to be used by many people to be maximally effective. There are two potential impediments to the use of the Control Tutorials for Matlab. First is the bandwidth of the internet itself. Students will not use a tool if it involves frustrating waits for pages and images to download. To alleviate this bandwidth problem, a

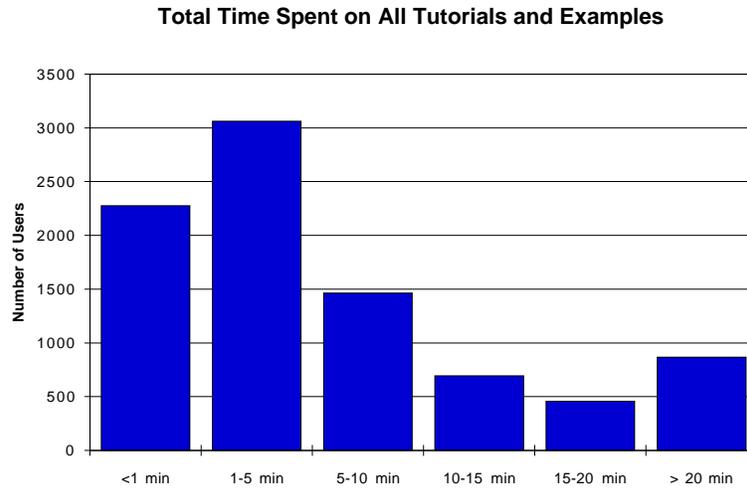


Figure 5: The distribution of total time spent by users on all pages September 1996–February 1997.

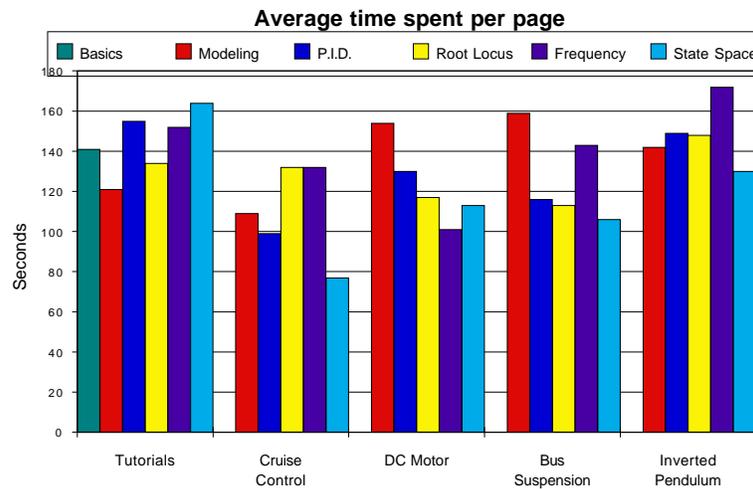


Figure 6: The average amount of time spent per page September 1996–February 1997, broken down by page.

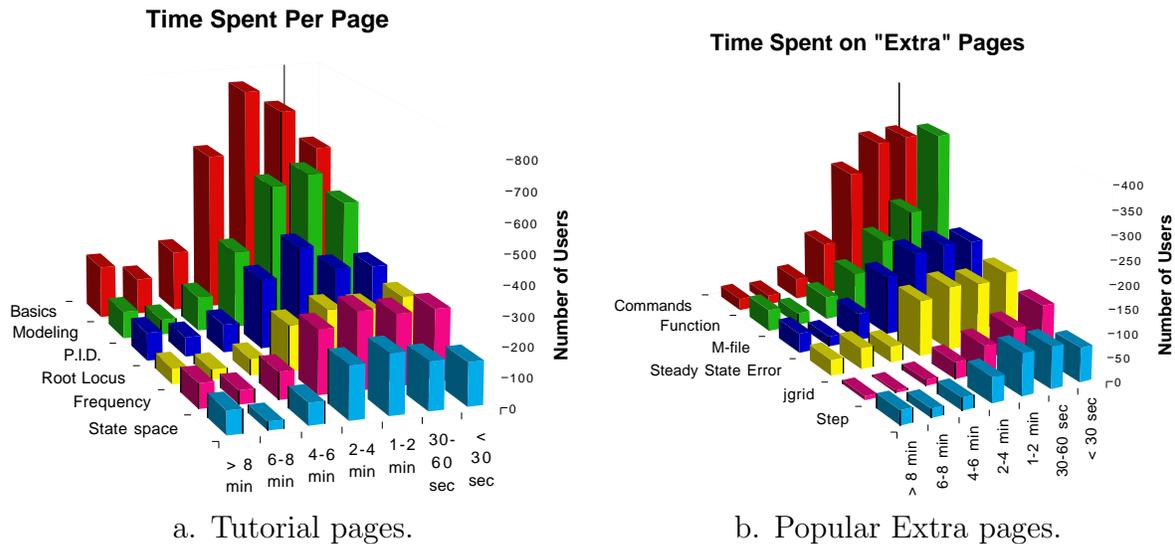


Figure 7: The distribution of amount of time spent per page for tutorial and extra pages. Data taken from September 1996–February 1997.

copy of the tutorials has been made available to interested universities. A compressed tar file (2.4MB) is available for downloading by anonymous ftp; the tutorials can then be installed on any local server (instructions on downloading and installing the tutorials are available by email from the authors). Currently, the tutorials have been installed at 26 sites in the US (e.g. Columbia, University of Illinois, Ohio State, Oregon State) and around the world (e.g. Australia, England, France, Italy, Singapore, Kuwait, South Africa).

The second impediment is simply letting people know of the existence of the tutorials. Savvy students may have discovered the tutorials while surfing the web, but most engineering faculty members don't have time for random exploring. Word of mouth from the authors to colleagues has been the most effective dissemination method so far. We have accomplished this by through face to face meetings and a letter-writing campaign to about 100 of our colleagues. The result was local installations at 12 universities. In December, an announcement describing the tutorials appeared in the "e-letter", an internationally distributed electronic newsletter for people in the systems and controls field, as well as on the Matlab newsgroup (comp.soft-sys.matlab). The number of requests increased dramatically, and another 12 requests to install the tutorials were received. We are currently writing letters and sending brochures to heads of all departments of Mechanical Engineering and Electrical Engineering in the US, and plan to present the tutorial concept this summer not only at ASEE but also at the American Control Conference.

6 Future Work

The success of these tutorials leads to the question "Where do we go from here?" In the short run, improvements to the existing tutorials will be made. Future versions of the tutorials will

incorporate more varied examples, as well as changes to address the feedback received from users of the current version of the tutorial. The authors also plan to extend the tutorials to incorporate digital control system design and Simulink. Java programs will be incorporated into the examples to animate the response of physical systems in the open-loop and with feedback control; the interactive nature of Java will allow the user to change the feedback gains or other control and system parameters.

However, the success of the concept indicates that other applications are possible. A reasonable first step is extending the tutorials for other software packages used for controls, such as Mathematica. However there is no reason to limit the concept to the controls area. We hope to extend the use of the tutorial concept to other useful engineering packages such as ANSYS and ProEngineer.

In addition, we have already begun to apply the tutorial concept for education laboratories. Lab assignments which appear to be similar to traditional laboratory handouts can be seen on the Web at <http://www.me.cmu.edu/faculty1/messner/classes/24-246/labs/lab1/>. The intent is to provide each laboratory station with a PC with access to Web tutorials, thus giving students information when and where they need it, in this case in the lab. The laboratory instructor can then be less of a technician and more of a coach.

Many of the references to important concepts in the lab assignment, such as step response, frequency response, and modeling, have hyperlinks to the relevant place in the Matlab tutorials. There are many pictures of the experimental set-up in various stages of construction; this allows a student to build the hardware independently of the teaching assistant. Furthermore, some of the pictures themselves are hyperlinks. Clicking on the portion of a photo which shows a piece of equipment brings up a description of that piece and a link to extended documentation. Later on in the assignment, photographs of the expected output of the equipment are shown. These are also live with links to relevant portions of the Matlab tutorials. For example, the picture of the oscilloscope showing the step response of the speed of a DC motor is linked to the DC motor example of the tutorials showing a Matlab simulation of step response. As more labs are developed there will be links to previous labs for review purposes, and perhaps even to those from previous courses.

In the future, we intend to provide data acquisition hardware and software for the PC's and corresponding Web-based instruction. Students will be able to acquire data, analyze it on the PC, and design and construct controllers while in the laboratory. We believe that this idea has applications to continuing education, worker training, and documentation of maintenance and repair procedures in industry.

7 Conclusion

This paper has discussed the design and implementation of tutorials for teaching Matlab for control system design and analysis using the World Wide Web. The primary advantages of these education aids are their wide accessibility and that users are able to learn by seeing and doing. Another advantage is that different sections of the tutorials are highly interconnected using the hypertext markup language. Animation provides a means for illustrating sophisticated concepts which are difficult to explain using textbooks or a blackboard. The tutorials are an example of a new paradigm for teaching the use of software packages over

the Web.

Acknowledgments

We would like to acknowledge the work of undergraduate students Joshua Pagel and Yanjie Sun from the University of Michigan, and Luis Oms and Munish Suri from Carnegie Mellon, whose hard work produced the tutorials over the summer of 1996. Funding for this project was provided by the National Science Foundation under grant number DUE-9554819.

References

- [1] Richard C. Dorf and Robert M. Bishop. *Modern Control Systems*. Addison-Wesley, Reading, Massachusetts, seventh edition, 1995.
- [2] Jay Forrester. Road maps: A guide to learning system dynamics, 1996. <http://sysdyn.mit.edu/road-maps/home.html>.
- [3] Gene F. Franklin, J. David Powell, and Abbas Emani-Naeini. *Feedback Control of Dynamic Systems*. Addison-Wesley, Reading, Massachusetts, third edition, 1994.
- [4] Benjamin C. Kuo. *Automatic Control Systems*. Prentice Hall, Englewood Cliffs, New Jersey, seventh edition, 1995.
- [5] Benjamin C. Kuo and Duane C. Hanselman. *Matlab Tools for Control System Analysis and Design*. Prentice Hall, Englewood Cliffs, New Jersey, 1994.
- [6] Naomi E. Leonard and William S. Levine. *Using Matlab to Analyze and Design Control Systems*. Benjamin Cummings, Redwood City, California, 1995.
- [7] Norman S. Nise. *Control Systems Engineering*. Benjamin-Cummings, Redwood City, California, second edition, 1995.
- [8] Katsuhiko Ogata. *Solving Control Engineering Problems with Matlab*. Prentice Hall, Englewood Cliffs, New Jersey, 1994.
- [9] Wilson J. Rugh. Signals and systems demonstrations, 1996. <http://www.ece.jhu.edu/class/214/demos/index.html>.
- [10] Hadi Saadat. *Computational Aids in Control Systems Using Matlab*. McGraw-Hill, New York, New York, 1993.

Author Biographies

Dawn M. Tilbury received the Ph.D. degree in Electrical Engineering from the University of California, Berkeley, in 1994. She has been an Assistant Professor at the University of Michigan since 1995. During her first semester teaching at Michigan, she developed a

preliminary set of tutorials to help her students learn Matlab; this set of tutorials won an Undergraduate Computational Science Education Award from the US Department of Energy in 1995. Her research interests are in the area of nonlinear control with applications to robotics and manufacturing.

William C. Messner received his B.S. degree in mathematics from M.I.T. in 1985 and his M.S. and Ph.D. degrees in mechanical engineering from U.C. Berkeley in 1989 and 1992 respectively. Since January 1993 he has been an assistant professor at Carnegie Mellon University where he specializes in controls for data storage systems and robotics.