

Control Tutorials for Software Instruction over the World Wide Web

Dawn M. Tilbury, *Member, IEEE*, and William C. Messner, *Member, IEEE*

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 academic year of use 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.

Index Terms—Control engineering education, educational technology, software tools instruction, Web-based tutorials.

I. 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 Web pages 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 lost.

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 just over 1/4 of the users in the United States are from companies (.com domains).

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 [12]. 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 original version of the tutorials went on-line in September of 1996; this paper describes the tutorial concept and overall structure as well as an assessment of the original version. The tutorials were updated over the summer of 1997 by adding one more tutorial (for digital control), three more examples, and four interactive animations; the overall structure remains unchanged.

The outline of this paper is as follows. Pedagogical considerations are discussed in Section II. The structure and content of the tutorials, including unique features of the tutorials, are presented in Section III. The use of the tutorials and their

Manuscript received November 3, 1997; revised August 19, 1999. This work was supported by the National Science Foundation under Grant DUE-9554819.

D. M. Tilbury is with the Department of Mechanical Engineering and Applied Mechanics, University of Michigan, Ann Arbor, MI 48109 USA.

W. C. Messner is with the Department of Mechanical Engineering, Carnegie Mellon University, Pittsburgh, PA 15213 USA.

Publisher Item Identifier S 0018-9359(99)09007-X.

¹The tutorials can be accessed on the WWW at <http://www.engin.umich.edu/group/ctm> or <http://hpme12.me.cmu.edu/matlab/html>.

integration into the curriculum is described in Section IV. Assessments of the use and utility of the tutorials are discussed in Section V. Efforts to disseminate the tutorials are described in Section VI, and future work is discussed in Section VII.

II. 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. 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], [10], [13], 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, seven examples of physical systems are incorporated into the tutorials. For each physical system, modeling and five control design examples are given. These 42 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 Fig. 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], [3], [4], [9]. In addition to the main tutorial pages, there are seven 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 seven root locus examples. A student interested in the DC motor might start with its model and view all five motor control design examples.

The hypertext markup language on the Web is also well-suited for allowing background and supporting information to

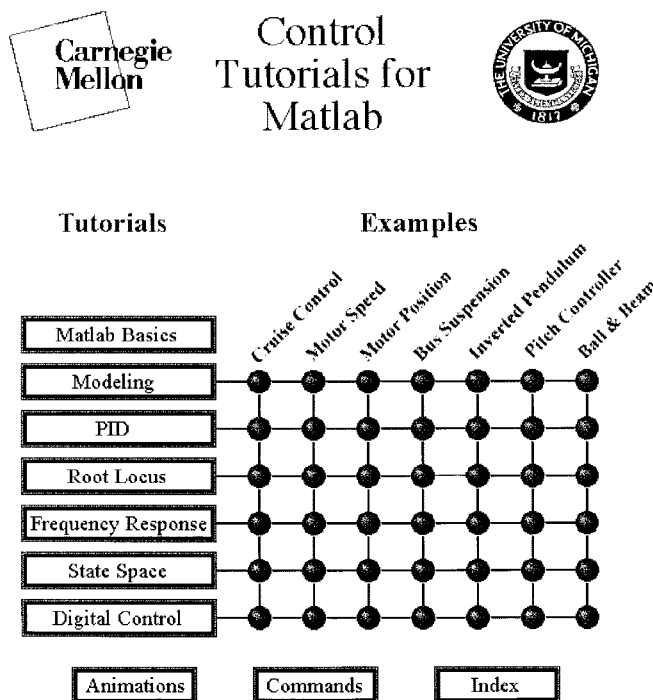


Fig. 1. The image map on the home page of the tutorials. A mouse click on the image leads to the corresponding Web page or site. Each tutorial is represented by a box, and the flow of the tutorials is from top to bottom. There are also seven examples which are followed through the tutorials (each example 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 page describing the animations which are implemented. A link to the National Science Foundation, which provided funding for this project, is not shown.

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 as a hyperlink which 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 indexes. As discussed above, the home page shows the overall structure of the tutorials. At the top of every page is an outline with subheadings 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 to not only other types of software, but also to the use of laboratory equipment. Some possibilities in this direction are discussed in Section VII.

III. 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) and digital control techniques. 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 are able to run any 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.

A. 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, and plotting. Because transfer functions are usually 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, taking the Laplace transform of a differential equation to find the transfer function, converting equations into state-space form, entering the transfer function or state-space equations into MATLAB, and converting between state-space equations and transfer function forms.

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 next 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.

A digital control tutorial was added to the revised version of the tutorials by popular request of the users; this is another topic which has not traditionally been covered within the undergraduate curriculum. The digital control tutorial includes some of the basic concepts used in digital control, including sampling, zero-order-hold, and Z -transforms. The stability region for digital control is demonstrated using the root locus. Digital PID and state-space control are covered in the examples.

B. Example Pages

In addition to the six main tutorials, there are seven examples which reinforce and extend the concepts and techniques introduced in the tutorials. The original version of the tutorials contained four examples; three more were added during the revision process. The examples are ordered in terms of increasing complexity, and are all based on physical system models (not abstract equations or transfer functions).

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

The first example, cruise control, is a simple example of the modeling and control of a first order system, taking inertia and damping into account.

The dc motor has second-order speed dynamics when mechanical properties such as inertia and damping as well as electrical properties such as inductance and resistance are taken into account. The controller’s objective is to maintain the speed of rotation of the motor shaft with a particular step response. This electromechanical system example demonstrates slightly more complicated dynamics than does the cruise control example, requiring more sophisticated controllers. The model of the position dynamics of a dc motor is third order. In this example, however, the motor parameters are taken from an actual dc motor used in an undergraduate controls course. This motor has very small inductance, which effectively reduces the example to second order. It differs from the motor speed example in that there is a free integrator in the open-loop transfer function. Also introduced in this example is the compensation for a disturbance input. This requires a free integrator in the controller, creating instability in the system which must be compensated for.

The bus suspension example looks at the active control of the vertical motion of a bus suspension. It takes into

account both the inertia of the bus and the inertia of the suspension/tires, as well as springs and dampers. An actuator is added between the suspension and the bus. This fourth-order system is particularly difficult to control because of the existence of two zeros near the imaginary axis, requiring careful compensation.

The inverted pendulum is a classic controls demonstration where a pole is balanced vertically on a motorized cart. It is interesting because without control, the system is unstable. This is a fourth-order nonlinear system which is linearized about the vertical equilibrium. In this example, the angle of the vertical pole is the controlled variable, and the horizontal force applied by the cart is the actuator input.

The pitch angle of an airplane is controlled by adjusting the angle (and therefore the lift force) of the rear elevator. The aerodynamic forces (lift and drag) as well as the airplane's inertia are taken into account. This is a third-order nonlinear system which is linearized about the operating point. This system is also naturally unstable in that it has a free integrator.

The ball and beam is another classic controls demonstration. A ball is placed on a straight beam and rolls back and forth as one end of the beam is raised and lowered by a cam. The position of the ball is controlled by changing the angular position of the cam. This is a second-order system, since only the inertia of the ball is taken into account, and not that of the cam or the beam, although the mass of the beam is taken into account in the fourth-order state-space model. The equations are linearized by assuming small deflections of the cam and beam.

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

C. "Extra" Pages

There are many topics covered in the tutorials that do not 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 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.

D. Implementation Details and Unique Features

Fig. 2 shows a small piece of the root locus tutorial. Several things to note are the combinations of explanatory text,

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])
```

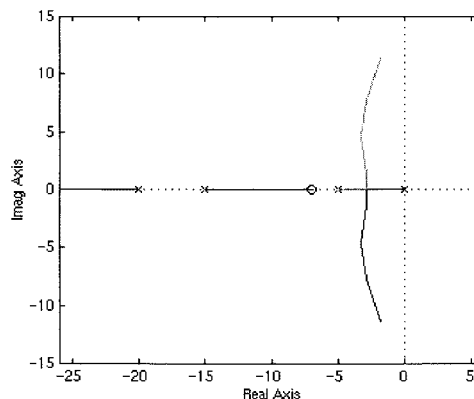


Fig. 2. Snapshot from a root locus tutorial. Note the juxtaposition of explanatory text (in proportional font), MATLAB code (in fixed font), equations (included as in-line gifs with transparent background), and figures (the figure is in color if viewed on a color monitor). The underlined words (MATLAB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

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 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.

In the newest version of the tutorials, interactive MATLAB animations have been included for four of the examples. These interactive animations allow the user to design a feedback controller (by placing poles, choosing weights for an LQR design, or choosing the gain in a root-locus design) and simultaneously simulate and animate the result. The step response of the closed-loop system is drawn on the screen at the same time as the animation, reinforcing for the student the physical meaning of the response. The full nonlinear model can also be simulated if desired.

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 backward-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 undergraduate engineering students, working at the University of Michigan and Carnegie Mellon University over two summers, under the direction of the authors.

IV. EMPLOYING THE TUTORIALS

The original set of tutorials went on-line in September of 1996. They were incorporated into an undergraduate course on automatic control (EE 141) taught by Dr. L. Bushnell at Duke University in the Fall of 1996, and they were used as reference by the undergraduate controls courses at the University of Michigan (Aero 471, EECS 460, ME 461) and Carnegie Mellon University (ME 246) during the academic year 1996 to 1997. The address of the tutorial Web site was also entered into the most popular WWW search engines (e.g., Altavista, Lycos, Yahoo), attracting a global user population. In this section, we will describe the usage of the tutorials, based on statistics from the University of Michigan’s Web server, e-mail feedback from users, and informal contact with students.

A. How the Tutorials Are Used

Our original intention in designing the tutorials was that students would follow along by running the tutorials in one window and MATLAB in another. Because of the ease of moving between the two programs, we expected that they would run the MATLAB programs by copying and pasting

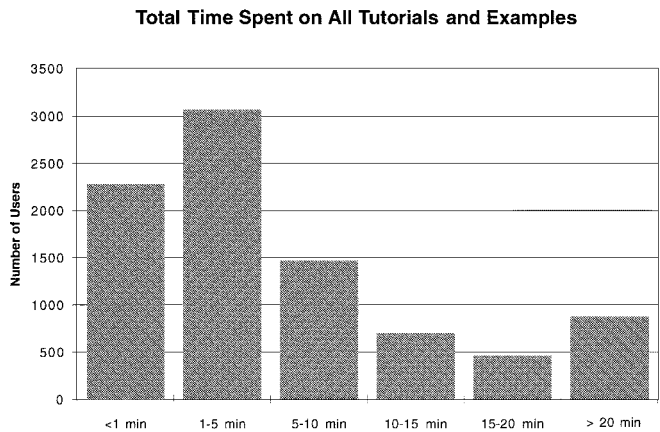


Fig. 3. The distribution of total time spent by users on all pages.

between windows. We have discovered, through conversations with students and through e-mail feedback from users, that this model is not universally followed. Of the 189 e-mail feedbacks that we received in the first six months, 40% of the students stated that they ran none of the MATLAB code (30% ran some and 30% ran most). The tutorials are actually used in many different ways. First and foremost the tutorials are a quick and easy reference when doing homework assignments. Information on the use of a particular command or technique is readily available. The ability to quickly move among pages to get information is an attractive attribute of the WWW. Second, students have found the tutorials to be useful for reviewing key concepts before exams. Although we did not expect the tutorials to ever be printed out, we often get e-mail questions about printing problems or suggestions to change the background color of the plots from black to white to make printing easier (this was done in the revised version). We have also met a practicing engineer who has printed out every tutorial and meticulously typed each program into an *m*-file in an effort to really learn MATLAB.

The hypothesis that the tutorials are used primarily for quick reference and as study guides is supported by statistics from the University of Michigan’s Web server. The total amount of time that users spent in all of the tutorials is shown in Fig. 3; the mode of the distribution histogram is between 1 and 5 min. This information is broken down by page in Figs. 4 and 5. Although the tutorial pages are much longer and contain more information than the “extra” pages, the mode of all of the distributions was between 1 and 2 min. More details on the data acquisition are given in Section V-B.

B. Integration in the Curriculum

For the tutorials to be most effective as an education tool, they must be integrated in a manner which is most consistent with how they are likely to be used. Thus, they should be employed most often in situations where easy reference is needed. At the University of Michigan and Carnegie Mellon, the undergraduate control course Web pages contain links for easy access to the MATLAB tutorials, and references to the appropriate tutorial pages are included on homework assignments. Reading assignments are supplemented with assigned

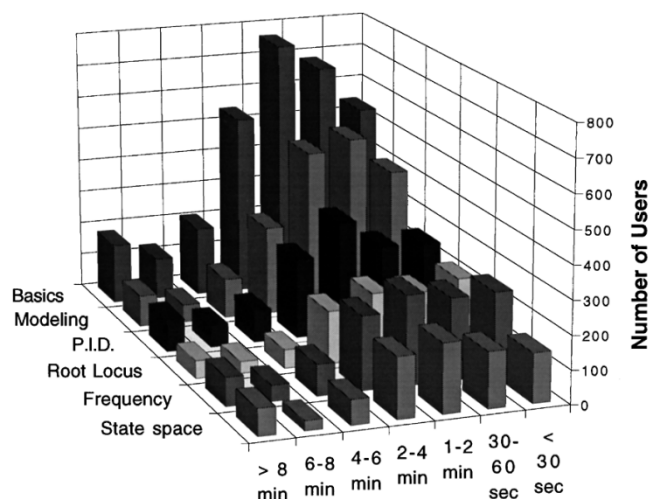
Time Spent Per Page

Fig. 4. The distribution of amount of time spent per tutorial page.

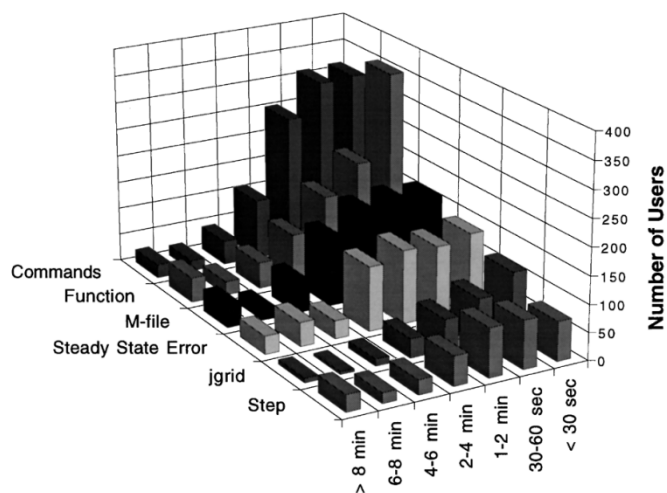
Time Spent on "Extra" Pages

Fig. 5. The distribution of amount of time spent per extra page.

reading of tutorial pages. This is not inconsistent with the use of the tutorials as easy reference, since it familiarizes students with the location of material that they may need at a later time. Students have commented that assignment of the root locus tutorial was particularly helpful when that topic was covered in class.

The Web-based tutorial concept need not be confined to software instruction; we are currently extending the concept for educational laboratories, as described in Section VII.

V. 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. In the original version, each form contained four multiple-choice questions and a section for free-form comments; only the free-form comments section is included in the revised version. The form is sub-

TABLE I
SUMMARY OF USER RESPONSES

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 min	133
30-60 min	44
More than 60 min	13
How much of the MATLAB code did you run?	
None	77
Some	56
Most	56

mitted (anonymously) by electronic mail to the authors for processing and evaluation.

Two different means are available for the assessment of the utility of the tutorials. The first is direct feedback available from the users through the forms interface described above. The e-mail 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 spent per page, and the total time spent looking through the tutorials.

A. User Feedback

Direct feedback is available from the users through a forms interface at the bottom of every page. During the period September 1996 to January 1997, we received 191 e-mail responses; these responses are tabulated in Table I. 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), eight were problems encountered (such as printing problems), and four were other comments, (including some in a foreign language that we could not 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)
- Excellent page with heaps of really useful stuff!! Keep it up folks!! (University of Aberdeen, U.K.)
- 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 Corp.)
- 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 your site (MATLAB tutorials) is and I would also like to say that it will be great if you include a tutorial for a reduced order observer. (Kuwait)
- Because I have no manual these pages are a real 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 affected 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 the 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.

B. 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 first academic year that the tutorials were available, September 1996–May 1997. The Web server tracks each request for a page and the domain name of

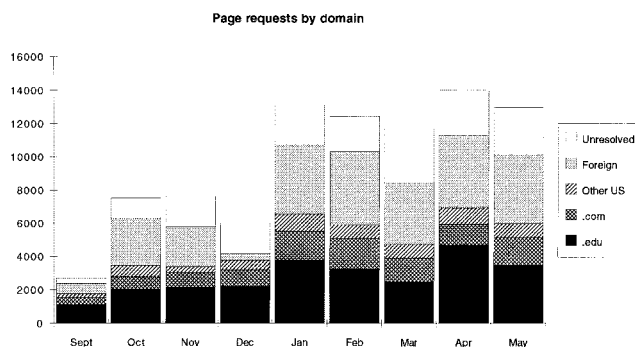


Fig. 6. The total number of page requests per month, broken down by domain.

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 UM.

As shown in Fig. 6, the traffic continued to grow during the fall semester as more and more people learned about the tutorials. The usage statistics follow the academic calendar closely, dropping off over the semester break in December and increasing dramatically in January (this sharp increase coincides with our preliminary advertising efforts as described in Section VI). The average number of users per day in February is actually slightly greater than that in January; because the month is three days shorter, however, it shows up as a decrease on the graph. March shows a slight dip coinciding with spring break, and the usage drops off again in May coinciding with the end of the academic year. We emphasize that these statistics are only for the University of Michigan's Web server. The tutorials were also installed at Duke and CMU during the entire year; statistics are not available for those installations. The first remote sites were installed in December of 1996; we do not have any access statistics for other sites.

Also shown in Fig. 6 is the breakdown of visitors by their location. As expected, a large portion of the users were from United States educational institutions. A surprising number were from foreign countries and United States 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 Fig. 7. The basics tutorial is by far the most popular; evidently, many of the visitors are not as interested in controls 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

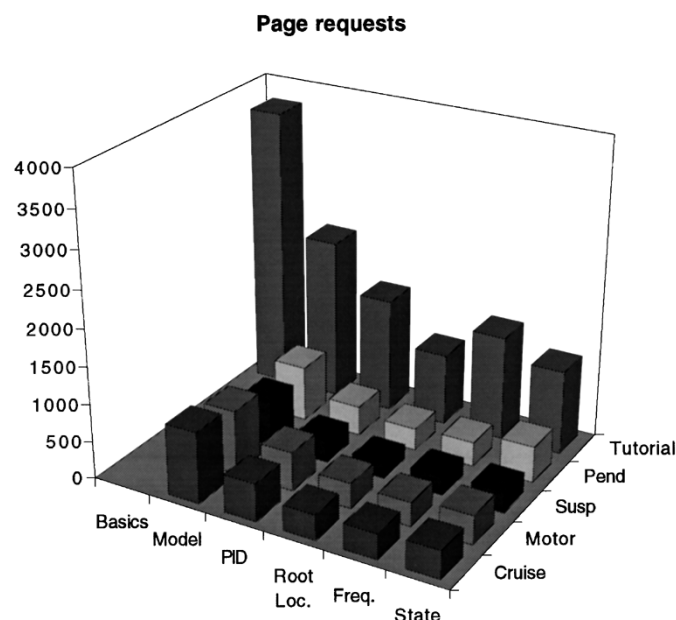


Fig. 7. The total number of hits per page over the total period September 1996–May 1997, showing the most popular tutorial pages. Only the original six tutorials and four examples are included.

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 min.

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.

The statistics summarized in Figs. 3–5 are gleaned from data collected at the University of Michigan server from September 1996 through February 1997. Approximately 13 000 distinct users accessed the tutorials during this time. Time statistics could be determined for 8800 of these users.

Fig. 3 shows the distribution of the total time spent by users on all pages associated with the tutorials including examples and extra pages. The number of people who spent 1 min or less on the tutorials seems high, but it is important to remember that this includes many 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 min.

Fig. 4 shows again 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 Fig. 5. 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.

VI. 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 copy of the tutorials was made available to interested Universities. The full installation requires less than 5 MB of disk space. The tutorials have been installed at 60 remote sites in the United States (e.g., Columbia, University of Illinois, Ohio State, and Oregon State) and around the world (e.g., Australia, England, France, Italy, Singapore, Kuwait, and 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 do not 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 of 1996, 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 news group (comp.soft-sys.matlab). The number of hits increased dramatically, and another 12 requests to install the tutorials were received. We also wrote letters and sent brochures to the heads of all departments of Mechanical Engineering and Electrical Engineering in the United States, and presented the tutorial concept at several conferences [11], [15], [16]. These efforts helped to bring the total number of remote installations up to 60. Of particular interest is the fact that the tutorials are installed at 40 of the ME and EE departments in the United States, a penetration of greater than 10%.

VII. FUTURE WORK

The success of these tutorials leads to the question "Where do we go from here?" In the short run, the answer was to improve upon and expand the existing set of tutorials. The version released in September of 1997 includes three more examples, one more tutorial (digital control), as well as interactive animations of some of the physical examples [17]. Simulink tutorials were included in a CD-ROM version of the tutorials [8], [14]. Other user comments, such as changing the background of the plots from black to white, and including more emphasis on the transfer function formulation, were also included in the revision.

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 educational laboratories [7]. 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. The MATLAB tutorials will be an essential part of this integration.

VIII. 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.

ACKNOWLEDGMENT

The authors would like to thank undergraduate students J. Pagel and Y. Sun from the University of Michigan for their work, and L. Oms and M. Suri from Carnegie Mellon, whose hard work produced the tutorials over the summer of 1996. The authors also would like to thank undergraduate students C. Caruana and D. Kawano at the University of Michigan, and P. Prempraneerach and B. Nakai at Carnegie Mellon University, for updating and expanding the tutorials over the summer of 1997. Graduate student J. Luntz at Carnegie Mellon also contributed to their development.

REFERENCES

- [1] R. C. Dorf and R. M. Bishop, *Modern Control Systems*, 7th ed. Reading, MA: Addison-Wesley, 1995.
- [2] J. Forrester, "Road maps: A guide to learning system dynamics," 1996. Available <http://sysdyn.mit.edu/road-maps/home.html>
- [3] G. F. Franklin, J. D. Powell, and A. Emani-Naeini, *Feedback Control of Dynamic Systems*, 3rd ed. Reading, MA: Addison-Wesley, 1994.
- [4] B. C. Kuo, *Automatic Control Systems*, 7th ed. Englewood Cliffs, NJ: Prentice-Hall, 1995.
- [5] B. C. Kuo and D. C. Hanselman, *MATLAB Tools for Control System Analysis and Design*. Englewood Cliffs, NJ: Prentice-Hall, 1994.
- [6] N. E. Leonard and W. S. Levine, *Using MATLAB to Analyze and Design Control Systems*. Redwood City, CA: Benjamin-Cummings, 1995.
- [7] J. Luntz, W. Messner, and D. Tilbury, "Web technology for controls education," in *Proc. IEEE Conf. Decision Contr.*, Dec. 1997.
- [8] W. C. Messner and D. M. Tilbury, *Control Tutorials for MATLAB and Simulink: A Web-Based Approach*. Menlo Park, CA: Addison-Wesley, 1998.
- [9] N. S. Nise, *Control Systems Engineering*, 2nd ed. Redwood City, CA: Benjamin-Cummings, 1995.
- [10] K. Ogata, *Solving Control Engineering Problems with MATLAB*. Englewood Cliffs, NJ: Prentice-Hall, 1994.
- [11] J. Pagel, Y. Sun, D. Tilbury, L. Oms, M. Suri, and W. Messner, "Control tutorials for MATLAB on the World Wide Web," in *Proc. Amer. Contr. Conf.*, 1997, pp. 3150-3154.
- [12] W. J. Rugh, "Signals and systems demonstrations," 1996. Available <http://www.ece.jhu.edu/class/214/demos/index.html>
- [13] H. Saadat, *Computational Aids in Control Systems Using MATLAB*. New York, NY: McGraw-Hill, 1993.
- [14] D. Tilbury, J. Luntz, and W. Messner, "Controls education on the WWW: Tutorials for MATLAB and Simulink," in *Proc. Amer. Contr. Conf.*, Philadelphia, PA, 1998, pp. 1304-1308.
- [15] D. Tilbury and W. Messner, "A case study in software instruction over the World Wide Web: The Michigan-CMU control tutorials for MATLAB," in *Proc. ASEE Annu. Conf. Exposition*, 1997, Session 3266.
- [16] ———, "Development and integration of Web-based software tutorials for an undergraduate curriculum: Control tutorials for MATLAB," in *Proc. IEEE Frontiers Educ. Conf.*, Pittsburgh, PA, Nov. 1997.
- [17] ———, "Controls education on the Web: Using MATLAB for control design, simulation and visualization," in *Proc. ASME Int. Mechanical Eng. Congr. Exposition*, vol. DSC-64, 1998, pp. 65-70.

Dawn M. Tilbury (S'91–M'95) received the B.S. degree in electrical engineering (summa cum laude) from the University of Minnesota, Minneapolis, in 1989, and the M.S. and Ph.D. degrees in electrical engineering and computer sciences from the University of California, Berkeley, in 1992 and 1994, respectively.

Since 1995, she has been an Assistant Professor in the Mechanical Engineering and Applied Mechanics Department at the University of Michigan, Ann Arbor. Her research interests include trajectory planning for nonlinear systems (including mobile robots), real-time distributed control of mechanical systems, and logic control of manufacturing systems.

Dr. Tilbury received an Undergraduate Computational Engineering and Science Award from the U.S. Department of Energy in 1995 for her work in Web-based software tutorials, and the EDUCOM Medal (jointly with Dr. Messner) in 1997. She received an NSF CAREER award in 1999. She is a member of ASME, SME, SWE, and ASEE.

William C. Messner (S'89–M'91) received the B.S. degree in mathematics from the Massachusetts Institute of Technology, Cambridge, in 1985, and the M.S. and Ph.D. degrees in mechanical engineering at the University of California, Berkeley, in 1989 and 1992, respectively.

In 1993, he joined the Department of Mechanical Engineering at Carnegie Mellon University, Pittsburgh, PA, as an Assistant Professor and was promoted to Associate Professor in 1998. His research interests include the development and applications of controls to data storage systems and robotics. He is the Education Chairman of Carnegie Mellon's Data Storage Systems Center and he leads the center's servo control research effort.

Dr. Messner received the EDUCOM Medal (jointly with Dr. Tilbury) in 1997. He received Carnegie Mellon's George Tallman Ladd Award For Excellence in Research in 1996.