

Mastery Learning with the MATLAB Webserver

Roger A. Green
North Dakota State University

Abstract

Mastery learning, as introduced in the 1960's and 1970's, includes three distinguishing features: (1) the use of proctors, which allows tutoring, repeated testing, and immediate feedback; (2) the unit-perfection requirement, which requires students to document subject mastery before advancing to new material; and (3) the self-paced feature¹. Significant resource demands, particularly for proctors, has discouraged the widespread use of traditional mastery learning methods. New technologies and software, however, permit the practical implementation of mastery learning models. Motivated by the ubiquity of MATLAB in most engineering disciplines, this paper develops a general framework for using the MATLAB Webserver to accomplish mastery learning in engineering courses. Essentially, the MATLAB Webserver serves as a replacement for human proctors. With minimal programming, instructors can effectively harness MATLAB's web interface and delivery tools to promote mastery learning and improve student education.

To help illustrate mastery learning with the MATLAB Webserver, this paper details a pilot study involving electrical engineering students enrolled in a junior-level signals and systems course at North Dakota State University. Students access material through HTML documents and forms using standard web-browser technology. The advantage of web delivery is 24-hour system access. Dynamic information exchange occurs between the client and server machines through the Matweb CGI interface. Developed features include individualized student assignments, instantaneous feedback, mastery through assignment resubmission, and automated grading. Statistical analysis, student comments, and instructor perspectives document the effectiveness and limitations of the approach. Initial student response has been positive.

I. Introduction

Ever since reading Keller's article "Good-Bye, Teacher..." I have been intrigued with the idea of mastery learning. Mastery learning has three fundamental components¹:

- The use of proctors, which allows tutoring, repeated testing, and immediate feedback.
- The unit-perfection requirement, which requires students to document subject mastery before advancing to new material.
- The self-paced feature, which allows students to proceed through the material at their own pace.

Mastery learning is quite flexible, but significant resource demands, particularly for proctors, has discouraged the widespread use of traditional mastery learning methods. Mastery learning is difficult to implement, and it deviates significantly from the traditional approach of lecture, homework, and examination.

Amid the many advances in education research, the fact remains that the practice of education remains quite static. Few education innovations are adopted on a widespread basis. Despite its many advantages, mastery learning, for example, is not in common use. For many instructors, the cost and effort of implementation outweigh any potential benefits. In the end we typically educate our students with the traditional components of lecture, homework, and examination.

Still, significant changes are being realized within the context of traditional education methods, including use of the World Wide Web. According to a recent survey, “nearly nine in 10 college instructors believe that student use of computers enhances their learning.”² However, two-thirds of the instructors also felt stressed keeping up with technology or knowing how to effectively utilize technology to improve education. Not surprisingly, few instructors regularly utilize computer technology in the classroom.

We extol the student benefit of education innovations, but the point is moot if the technologies for education improvement are inaccessible to instructors. If educational methods are time-consuming or complicated, then they will not be widely adopted. Technology enhanced education must be easy for the instructor as well as the students.

Versatility is equally important. Impressive examples of web-enhanced courses abound, but the majority of us abandon the notion of using these tools because they don't quite fit our courses and we lack the resources or skills to adapt them to our particular needs. John Hopkins University has an excellent web site for the demonstration of signals, systems, and controls topics^{5,6}. Because the applets are fixed and cannot be customized, however, their use is necessarily limited.

Other times, the tools are easy to customize and implement but possess insufficient power for engineering applications. Jong and Muyschondt, for example, recently developed a set of CGI and PERL scripts for interactive web-based testing with feedback⁷. Unfortunately, these tools are restricted to multiple choice questions, and instructors must fix each specific question and answer set prior to program execution. As a result, open-ended or individualized assignments are impractical. A powerful, easy-to-operate toolbox is needed that allows instructors to develop individualized, open-ended, and custom applications.

Given these thoughts, it seems wise that improvements for web-based education follow three basic precepts:

- They should be easy to implement and easy to use.
- They should be flexible and customizable.
- They should complement the traditional approach of lecture, homework, and examination.

During the summer of 1999, I was introduced to the MATLAB Webserver through a programming contest. In a loose sense, the contest incorporated several aspects of mastery learning such as resubmission, immediate evaluation, tutoring, and self-pacing. I began to wonder if the MATLAB Webserver could be used to incorporate mastery learning in my own courses. I learned, in fact, that not only could I use the MATLAB Webserver to augment my courses, but so could nearly anyone else who was familiar with MATLAB as well as some very basic web programming using the HyperText Mark-up Language (HTML). The prevalence of MATLAB in engineering suggests that many engineering professors can benefit.

Following the contest, I began earnest attempts to adapt the MATLAB Webserver to the classroom. The goal is web-deliverable material that incorporates elements of mastery learning such as individualized student assignments, instantaneous feedback, mastery through assignment resubmission, and automated grading. If necessity is the mother of invention, then laziness must be a close relative. I want to improve my teaching and spend less time doing it. The MATLAB Webserver allows both. Except for small classes, automated grading alone can offset the time required for system development. Because the system is MATLAB based, implementation is flexible and relatively easy.

This paper is organized as follows. Section II develops a general framework for using the MATLAB Webserver to incorporate mastery learning in engineering courses. A detailed yet basic example is included to demonstrate the simplicity of system development. Section III details a pilot study involving electrical engineering students enrolled in a junior-level signals and systems course at North Dakota State University. Sophisticated examples from this course are presented to demonstrate the flexibility and power of the MATLAB Webserver. Finally, Section IV provides statistical analysis, student comments, and instructor perspectives to document the effectiveness and limitations of the approach.

II. Mastery Learning with the MATLAB Webserver

The MATLAB Webserver provides tremendous opportunities for education-based applications. Student specific assignments can be generated, automatically graded, and recorded. Material is accessible 24-hours a day, and feedback is nearly instantaneous. In the context of distance education, clients can utilize MATLAB functionality without the purchase of expensive software. Inter-collegiate activities are possible. Even high-school students can benefit from this software by way of interactive, dynamic web sites.

The MATLAB Webserver enables the development and distribution of web-based MATLAB applications. Through the use of standard HTML documents and forms, the MATLAB Webserver allows MATLAB programmers to develop web-deployable applications from standard MATLAB components. The HTML documents serve as a point-and-click Graphical User Interface (GUI) for the MATLAB application being deployed. One significant advantage is that MATLAB need not be running locally on the client machine. Furthermore, it is possible to develop applications that require end-users to possess little or no MATLAB knowledge or experience.

Figure 1 displays the essential details of the MATLAB Webserver environment. Using client machines, students access material through HTML documents and forms using standard web-browser technology. On the server side, information is received and transmitted by a standard HTTP daemon. Matlabserver is a TCP/IP server that manages communication between web applications and MATLAB. Matweb is a TCP/IP client of Matlabserver that extracts data from HTML documents and transfers information to and from Matlabserver³. Since MATLAB has full drive access of the server machine, it can access sensitive file information, such as class rosters and student grades, that are not accessible through the HTML root path. This helps protect student files and grades from unauthorized web access.

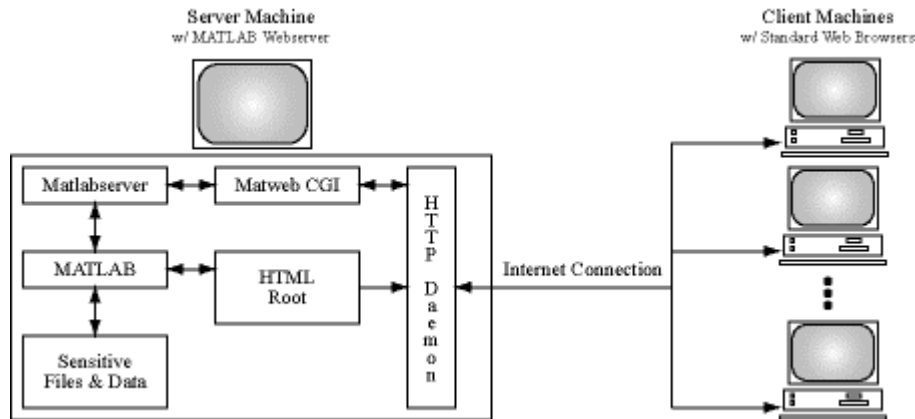


Figure 1: MATLAB Webserver Environment

Fortunately, the vast majority of the components in Figure 1 are transparent to the instructor who is developing class material and to students who are accessing the material from client machines. The instructor is responsible for three basic components:

- An HTML input form to submit student data to MATLAB.
- A MATLAB m-file to process input data and compute results.
- An HTML output document to display feedback information.

Using a simple example, these three pieces are considered in turn to demonstrate the simplicity of system development.

HTML Input Form:

The HTML input document queries the student for information, and passes this information to the server machine. HTML forms make data collection extremely easy, and numerous books are available that summarize the different HTML form elements⁴. It literally takes only a few minutes to write an HTML file that generates a form.

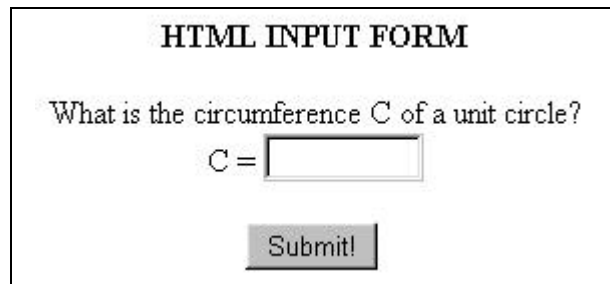
Consider the example of asking students the circumference of a unit circle. Figure 2 presents an example HTML file. Line numbers are added for clarity and are not part of the file.

1	<HTML><BODY>
2	<form action="/cgi-bin/matweb.exe" method="POST">
3	<input type="hidden" name="mlmfile" value="example1">
4	<CENTER>HTML INPUT FORM
5	What is the circumference C of a unit circle?
6	C = <input type="text" name="C" size=9>
7	<input type="submit" value="Submit!">
8	</BODY></HTML>

Figure 2: Example HTML Input Form (example1.html)

In Figure 2, lines 1 and 8 are needed for all HTML documents. Line 2 establishes this HTML document as a form, and tells the browser where to route data. In this case, information is passed to the Matweb CGI. Line 3 instructs the MATLAB Webserver to use the m-file named

“example1” to process data. Typically different forms use different m-files so it is convenient to use the same name for the input HTML file and corresponding m-file. Lines 4 and 5 are used to format text on the screen. The input text box is established by line 6. The contents entered into this box are assigned a variable name “C”. Other input types, such as checkboxes and radio buttons, are also possible⁴. Finally, line 7 creates the submit button, which is used by clients to initiate data transfer to the server machine. Figure 3 shows the appearance of this form using a standard web browser.



HTML INPUT FORM

What is the circumference C of a unit circle?

C =

Figure 3: Browser Display of Example HTML Input Form

MATLAB m-File:

Once data is passed to the MATLAB Webserver, an m-file function is required to process the data. Due to the prevalence of MATLAB in engineering, many professors already possess the necessary programming skills to construct these script files.

Continuing with the circle circumference example, Figure 4 presents an m-file that evaluates the user’s submission. Once again, line numbers are included for clarity.

```
1 function retstr = example1(input);
2 cd(input.mldir);
3 C = str2num(input.C);
4 if (fix(C*100)==fix(2*pi*100)),
5     output.message = ('Great! Your answer is correct.');
```

```
6 else,
7     output.message = ('Sorry, your answer is incorrect.');
```

```
8 end
9 retstr = htmlrep(output,'example1out.html');
10 return
```

Figure 4: MATLAB m-file (example1.m)

In Figure 4, lines 1 and 10 establish the file as a MATLAB function. The file and function name should match the name referenced in the input form, line 3 of Figure 2. Form data are passed into the function using the structure array named “input”. In this example, the circumference variable is passed into the function as “input.C”. That is, “C” is a field of the structure array “input”. Line 2 is used to set the proper working directory for the m-file. Directory management is particularly important when sensitive information such as class rosters and grade files are involved.

The input parameter “input.C” is an ASCII string, and line 3 converts this string to a MATLAB double-precision floating-point number. Lines 4 through 8 are used to evaluate whether the answer is correct and to generate an appropriate output message. For this problem the circumference of a unit circle is an irrational number (2π), and it is impossible to numerically express the exact answer. Thus, any answer within two decimal places is considered correct by the m-file. Accommodating inexact but correct answers complicates the program and demonstrates one limitation of computer evaluation of data.

Line 9 uses the “htmlrep” function supplied with the MATLAB Webserver to combine output data with the output HTML file, which in this case is named “example1out.html”. This output is assigned to the variable “retstr” which is then passed to the Matweb CGI for export to the client machine. More details of how output variables are combined with output HTML documents are included in the next subsection.

HTML Output:

Once form data is collected and evaluated, it is important to provide feedback to the user. Figure 5 provides an HTML output file to display feedback information.

1	<HTML><BODY>
2	<CENTER>HTML OUTPUT
3	\$message\$
4	</BODY></HTML>

Figure 5: HTML Output (example1out.html)

Again, lines 1 and 4 are common to every HTML document. The second line is used to put a title on the page. During execution, the MATLAB Webserver function “htmlrep” replaces the variables enclosed by dollar signs with the corresponding data. This permits dynamic page generation that adapts to individual user parameters. Figure 6 displays the appearance of the output using a standard web browser.

HTML OUTPUT
Great! Your answer is correct.

Figure 6: Browser Display of Example HTML Output

It should be emphasized that this instructional example does not utilize or demonstrate the full power of MATLAB. It does, however, provide a starting point to build more sophisticated examples. For example, MATLAB can easily utilize Fast Fourier Transforms and Monte-Carlo simulation to evaluate student designs. MATLAB can also generate sophisticated graphics to pass to client machines. These features are simple using MATLAB, but they would be very difficult to implement using standard HTML, CGI, PERL, or JAVA programming. Section III details some sophisticated and appropriate applications of the MATLAB Webserver from an engineering education context.

III. Signals and Systems: An NDSU Pilot Study

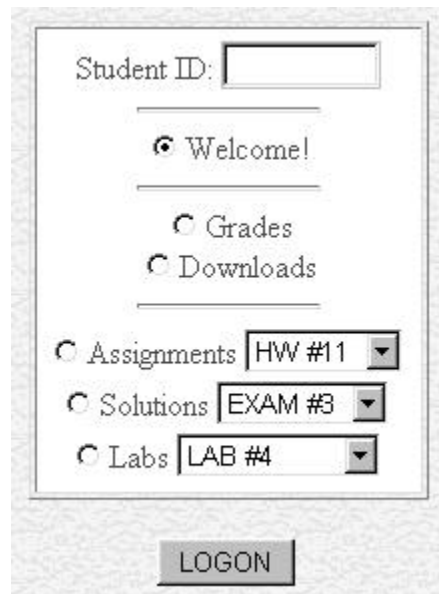
The use of traditional homework sets is ubiquitous in the university environment. Assignments provide a reasonable means of developing student competence, but unfortunately they suffer from several deficiencies. Typically, students are all given the same assignment. Often, there is a significant time delay between submission of an assignment and feedback from the instructor. Finally, students are rarely afforded the opportunity to revise and resubmit their work. The roots of these policies are more practicality and convenience rather than sound educational pedagogy.

The MATLAB Webserver and mastery learning techniques can improve homework set effectiveness without disrupting the lecture component of a course. Consider how the MATLAB Webserver accommodates some “assignment aphorisms”:

- Learning can happen anytime, anywhere. As long as the student has an Internet connection, the MATLAB Webserver is available on demand. With no clock or fixed schedule, students proceed at their own pace.
- Learning is individual. The MATLAB Webserver makes individualized student assignments feasible. Unless a student can convince someone else to do their assignment, they must work the material themselves.
- Feedback should be immediate. The MATLAB Webserver can evaluate student submissions within seconds, allowing feedback to be concurrent with the work performed. Students know what they did correctly and what they did incorrectly.
- Practice makes perfect. By configuring the MATLAB Webserver to allow assignment resubmission, students can rework problems to achieve material mastery.

An NDSU pilot study involving a junior-level signals and systems course demonstrates these features.

Figure 7 displays the main menu for the signals and systems course. To help maximize clarity and simplicity, the entire course web-site, from passwords to assignments to grades, is managed by the MATLAB Webserver. The result is an integrated site with consistent content and style.



The screenshot shows a web interface for a course. At the top, there is a text input field labeled "Student ID:". Below this, there are several radio buttons for navigation: "Welcome!" (which is selected), "Grades", and "Downloads". Further down, there are three more radio buttons, each followed by a dropdown menu: "Assignments" with "HW #11" selected, "Solutions" with "EXAM #3" selected, and "Labs" with "LAB #4" selected. At the bottom of the interface is a "LOGON" button.

Figure 7: Signals and System Main Menu

Each student is assigned a password to access the course web-site. This helps secure the site against unauthorized access and ensure student privacy. Additionally, the logon password establishes the identity of the user. This allows all page requests to be customized and automated grades to be credited to the proper user. Following password entry, the user selects the desired page and presses the “LOGON” button.

To help demonstrate mastery learning models in action, two assignments will be presented. First, consider the online homework assignment shown in Figure 8. Each student receives problems of the same form, but with different parameter values w_1 , w_2 , and w_3 . In this case, $w_1 = 2.8 - i8.6$, $w_2 = 7e^{-i2.85}$, and $w_3 = (i5.4)^i$. Once the form is completed, the student submits his or her answers with the press of a button.

5. Express the w_1/w_2 in polar coordinates.	$r =$ <input type="text" value="1.292"/> $\theta =$ <input type="text" value="1.594"/>
6. Express the w_3 in rectangular form: $w_3 = x + i*y.$	$x =$ <input type="text" value="-0.024"/> $y =$ <input type="text" value="0.2065"/>
7. Find real, positive constants, r and θ , for all time, t , such that: $6\cos(3*t) + -4.4\sin(3*t + 2.25) = r*\cos(3*t + \theta)$	$r =$ <input type="text" value="3.78"/> $\theta =$ <input type="text" value="0.82"/>
8. Express the primary value of the natural logarithm of w_1 in rectangular coordinates: $\ln(w_1) = x + i*y.$	$x =$ <input type="text"/> $y =$ <input type="text"/>
9. Consider the expression $(w-w_1)^2 = w_2$, where w is a complex variable. In rectangular coordinates, determine the solution to this equation that is closest to the origin.	$x =$ <input type="text"/> $y =$ <input type="text"/>
10. The magnitude squared of a complex number is easily computed by multiplying the number by its complex conjugate. Express the magnitude squared of w_2 in rectangular coordinates.	$x =$ <input type="text"/> $y =$ <input type="text"/>

Figure 8: Online Homework Assignment

Within seconds of submission, the student’s assignment is automatically graded and feedback is provided, as shown in Figure 9. The MATLAB Webserver automatically records the student’s score, which in this case is almost perfect. Only problem 7 was missed, and this was only due to a simple sign error. Notice that answers need to be within one percent of the true answer to be considered correct.

To encourage topic mastery, students can resubmit the assignment as many times as they like before the set due date. For this assignment, correct answers are provided with each submission. Unless problem values change, assignment resubmission is trivial! Thus, the MATLAB

Websserver changes the problem parameters each time the student retries the assignment. This provides students with additional practice in areas of personal difficulty.

SCORE: 9/10

Problem	Correct Answers		Your Answers		Credit
1	9.04434	-1.25604	9.04	-1.26	1
2	11.3078	-1.92335	11.31	-1.92	1
3	-36.0788	52.0242	-36.09	52.02	1
4	0.142857	2.85	0.1429	2.85	1
5	1.29205	1.59396	1.292	1.594	1
6	-0.0239779	0.206492	-0.024	0.2065	1
7	3.77859	-0.820491	3.78	0.82	0
8	2.20214	-1.25604	2.2021	-1.26	1
9	2.41562	-5.98232	2.416	-5.98	1
10	49	0	49	0	1

Figure 9: Homework Evaluation and Feedback

The second example presents a non-trivial open-ended design problem that truly utilizes the power of MATLAB. The essential details of the design are presented in Figure 10. Although every student is required to design a Butterworth lowpass filter using second-order Sallen and Key circuit stages, each student has a unique cut-off frequency. Although three filter design methods are provided in the pre-lab, each student must decide which design method to use. The assignment forms the backbone for a laboratory exercise where each student must realize his or her design in hardware.

Each design is submitted electronically, as shown in Figure 11, and evaluated according to the following criteria:

- The design must be Butterworth. To evaluate this criteria, the "Q" for each stage of the filter is compared to theoretical Butterworth "Q" values.
- Using nominal values, the mean-square-error (MSE) between each filter's actual and ideal frequency response is computed from DC to four times the cut-off frequency. A lower MSE indicates better performance. Gains are normalized prior to computations.
- Using 10-percent resistor and 25-percent capacitor tolerances, the worst frequency response MSE is recorded from 100 simulated circuit realizations. Small variation is desired, which indicates low sensitivity to component parameters. Gains are normalized prior to computations.

Students are constrained to choose resistors and capacitors from pull-down lists that contain only nominal component values. This reflects the real-life design decisions necessary in analog filter design. The constraint to use nominal component values also complicates the design procedure. As with any open-ended design, it is unwise to penalize students for trying something new. The goal is to allow students to discover what works best. Thus, there is no penalty for multiple submissions. Students are encouraged to submit new designs that may improve their circuit's performance. Any improvement is automatically recorded.

To prepare for upcoming Lab #2, you need to design a filter using a cascade of second-order stages. Design specifications are provided in Table 1, and Figure 1 shows the Sallen and Key circuit required to implement each stage. Four resistor and two capacitor values must be specified for each stage of your filter.

Design	Butterworth
Filter Order	Fourth
Filter Type	Lowpass
Cutoff Frequency	1576 Hz

Table 1: Design Specifications

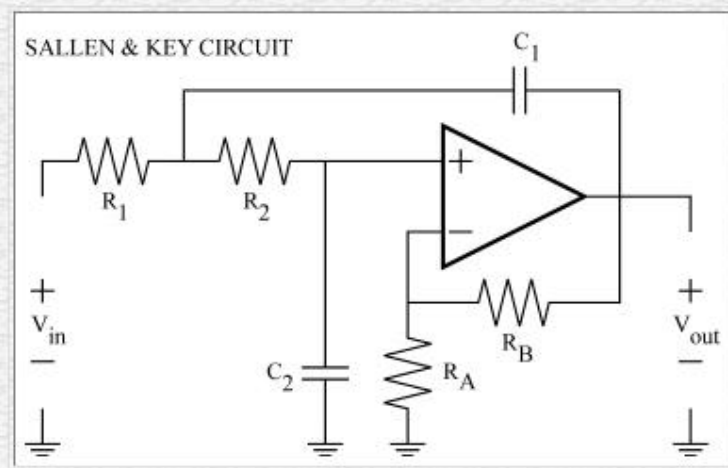


Figure 1: Sallen and Key Circuit

Figure 10: Filter Design Specifications

By requiring electronic submission, students receive immediate feedback regarding their designs. Each submission is anonymously ranked against the other designs in the class, including the instructor's design! This information helps students evaluate their performance relative to their peers and motivates continued improvement. Additionally, each evaluation criteria is detailed to help students identify the deficiencies of their design.

Figure 12 displays the overall design evaluation. Score, rank, and number of submissions are provided, as well as a customized message indicating relative performance of the current design compared with previous submissions. Figures 13, 14, and 15 display how the submission meets each of the three design criteria. Overall the presented design is good, but Figure 14 in particular illustrates how the filter differs from the theoretical Butterworth response. Based on the student's rank of 8 out of 25, improvements are still possible.

Select component values and then press the evaluate button. A value of "inf" represents an open circuit and a value of "0" represents a short circuit.

STAGE 1		STAGE 2	
R1 =	2.7 exp 5 ohms.	R1 =	4.7 exp 5 ohms.
R2 =	2.7 exp 5 ohms.	R2 =	4.7 exp 5 ohms.
RA =	inf exp 0 ohms.	RA =	inf exp 0 ohms.
RB =	0.0 exp 0 ohms.	RB =	0.0 exp 0 ohms.
C1 =	4.7 exp -10 farads.	C1 =	4.7 exp -10 farads.
C2 =	3.9 exp -10 farads.	C2 =	6.8 exp -11 farads.

Evaluate!

Figure 11: Electronic Submission of Filter Design

Total Score: 3.84
 Rank: 8 out of 25
 Number of Submissions: 16

Congratulations! You improved your score by 18.50 points.
 Your new submission has been recorded.

Figure 12: Filter Design Score and Rank

Criteria 1: Butterworth Design
 Q-Score = 2.03

	Theoretical	Actual	Percent Difference
Q, Stage 1	0.5412	0.5489	0.0142
Q, Stage 2	1.3066	1.3145	0.0061

Figure 13: Evaluation of Butterworth Design Criteria

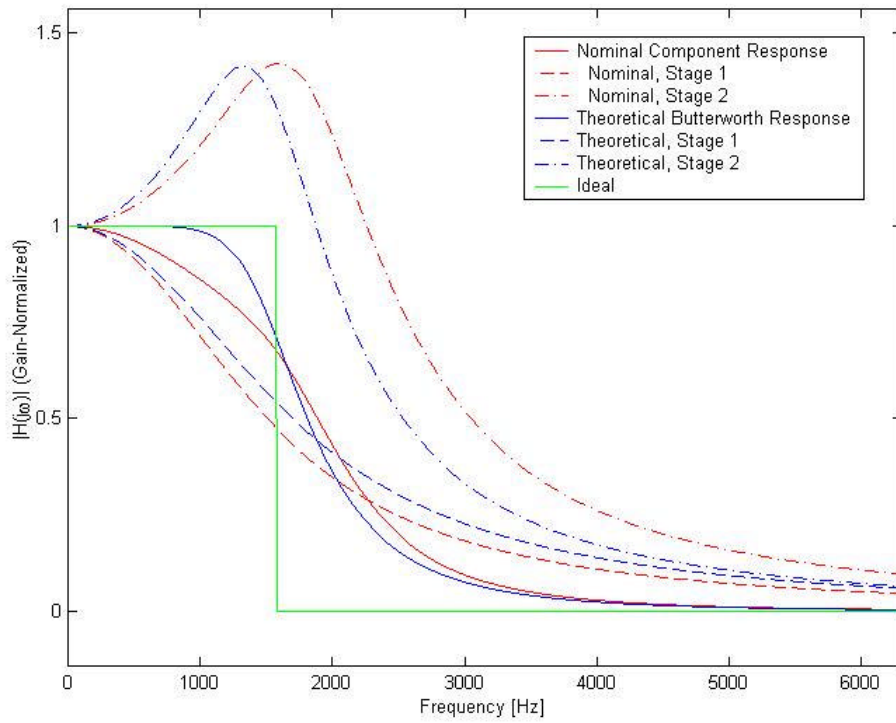


Figure 14: Evaluation of Frequency Response MSE Criteria

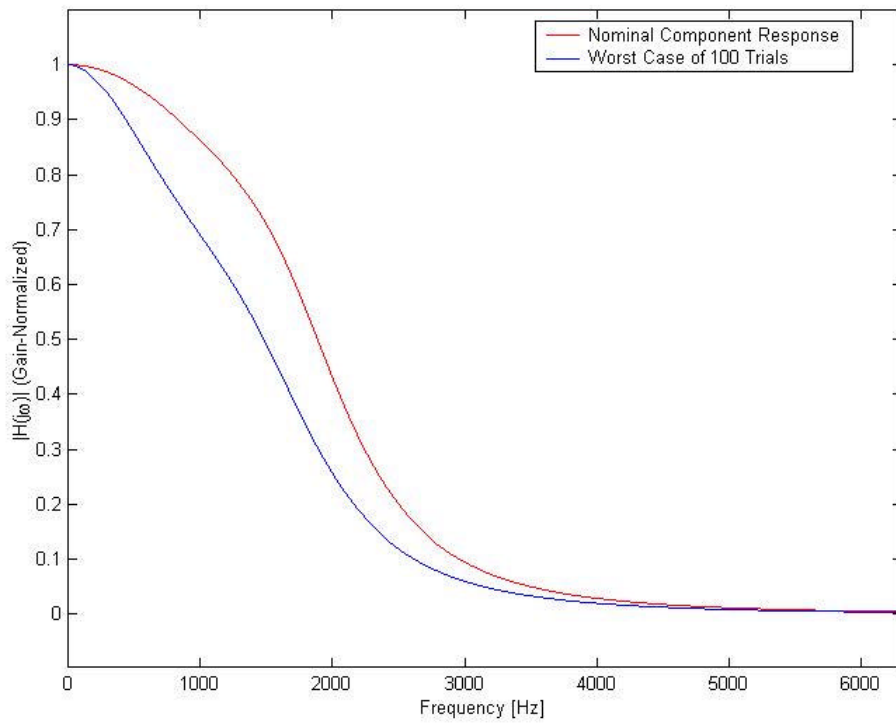


Figure 15: Evaluation of Frequency Response Sensitivity Criteria

IV. Analysis and Conclusions

During the pilot study semester, 23 students accessed the class web-site an average of roughly 100 times each. This translates to nearly one access per student per day. Some students accessed the site as few as 45 times, while others were closer to 200. If each access saved an average of one to five minutes of instructor time, then total savings range from a work-week to over a work-month. Larger class enrollments will obviously realize more savings.

Nearly 48% of student accesses were for homework and laboratory activities. Grade viewing accounted for around 31% of the accesses, while requests for online homework and exam solutions took 13%. Finally, course policies and file downloads each accounted for around 4% of site accesses. These figures are consistent with the goals established for the site, although the percentage for grade viewing was somewhat higher than expected and the percentage for assignment solution viewing was lower than expected. Availability of hardcopy assignment solutions likely reduced the web usage in this latter category.

The success of the two assignments detailed in Section III is somewhat varied. For the online homework assignment, students submitted solutions an average of 3.2 times. This demonstrates elements of mastery learning in action. Resubmission, however, did not necessarily result in unit perfection. The mean score on the assignment was 81.3%. Problems 7 and 9 seemed particularly difficult. In each case, fewer than 50% of the students obtained the correct answer.

The instructions for this assignment cautioned students against the blind use of calculators to obtain results. A major limitation of the MATLAB Webserver, in fact, is that it cannot verify the procedures that students use to obtain their answers. Consider the eighth problem in Figure 8, which can be solved in seconds with most engineering calculators. Over 90% of the class received credit for obtaining the correct answer for this problem. On the assignment due date, an identical-style problem was given as a quiz with the stipulation that calculators were not allowed. Only one of the 23 students correctly solved the problem. Clearly, online problems must be carefully designed to ensure students actually learn the material.

A subtler problem is identified when the submission times are analyzed. Although homework is assigned one week before the due date, over 40% of the class had not made any submission by 5:00 p.m. the night before the due date. The assumption that students will adapt to new teaching methods is not necessarily correct. It is difficult to affect changes in behaviors that are conditioned by years of traditional education. It should be noted that submission times and resubmission rates improved in subsequent online assignments.

The online filter design pre-lab was more successful. The average number of submissions was 15 and most students began the design well before the actual laboratory time. Furthermore, most students achieved good designs. Since the problem involved selecting 12 components from a fixed list of values, one student recognized that only a finite number of combinations were possible. At this point, he became intent on guessing his way to an optimal solution. As it turns out, he seriously underestimated the number of permutations -- 44.2 billion trillion. Even if he evaluated 100,000 combinations every second of every day, it would take him 14 billion years to exhaust the list of choices.

Throughout the semester, I have been pleased by the student comments regarding the class web site and the MATLAB Webserver. Most students appreciate the availability of class material. Since the class text does not supply answers to any problems, most students appreciated the immediate feedback of the online assignments. Furthermore, several students persistently resubmitted assignments until they received perfect scores, which indicated a sincere desire to learn the material. The online Butterworth filter design was tremendously satisfying, particularly for the students who beat the instructor's design score! Every student came to lab prepared with quality filter designs.

The homework and filter assignments described in Section III took around 8 and 20 hours of development time, respectively. In the former case, automated grading offset the development time. In the latter case, this type of individualized, open-ended problem would be impractical without a tool such as the MATLAB Webserver. The advantages outweighed the cost.

Given my experience with the MATLAB Webserver, the major deficiencies appear to be:

- It can be difficult to design suitable material.
- The MATLAB Webserver cannot evaluate the process used to reach a given solution.
- The MATLAB Webserver is not suited for symbolic problems, such as symbolic integration.
- Since the MATLAB Webserver operates on a single machine, server burden may be unacceptable.
- At the educational rate, the MATLAB Webserver is still a costly \$1000.

The pilot study at NDSU confirmed several advantages of the MATLAB Webserver, namely:

- It is easy to implement and easy to use.
- It is flexible and customizable.
- It is very powerful, particularly for engineering applications.
- It permits mastery learning techniques such as assignment resubmission, immediate feedback and grading, self-paced progression, and unit-perfection requirements.

Given these points, the MATLAB Webserver appears ideally suited to supplement course material. The MATLAB Webserver can improve student education and save time in the process. It is my hope that more engineering instructors utilize the MATLAB Webserver in their courses.

Bibliography

1. Keller, F. "Good-Bye, Teacher..." Journal of Applied Behavior Analysis, No. 1, Spring 1968, pp. 1-11.
2. URL: <http://www.msnbc.com/news/306031.asp>; "Professors Slow to Pick Up Tech Tricks," MSNBC report, 1999.
3. URL: http://www.mathworks.com/access/helpdesk/help/pdf_doc/webserver/webserver.pdf; "MATLAB Web Server," Version 1 documentation, 1999.
4. Oliver, D. Teach Yourself HTML 3.2 in 24 Hours, Sams.net Publishing, 1st Edition, 1997.
5. URL: <http://www.jhu.edu/~signals/>; "John Hopkins University Signals Systems and Control Demonstrations"
6. URL: <http://www.jhu.edu/~signals/CSMpaper.html>; Crutchfield, S. and Rugh, W. "Interactive Learning for Signals, Systems, and Control," to appear in IEEE Control Systems Magazine.
7. Jong, I. and Muyschondt, A. "Simple Scripts to Produce Interactive Web-Based Test with Immediate Feedback: Software and Illustration," ASEE Journal of Engineering Education, Vol. 88, No. 4, October 1999, pp. 509-525.

ROGER GREEN

Roger Green is an Assistant Professor in the Electrical and Computer Engineering Department at North Dakota State University. Dr. Green specializes in digital and statistical signal processing, time-frequency analysis, and array processing. Roger is committed to excellence in teaching through effective communication and classroom innovation. Dr. Green received his Ph.D. in Electrical Engineering from the University of Wyoming in 1998.