An Online Homework Generation and Assessment Tool for Linear Systems

Yong Yang, M.S., Department of Electrical & Computer Engineering
Andrew Bennett, Ph.D., Department of Mathematics
Steve Warren, Ph.D., Department of Electrical & Computer Engineering

Kansas State University, Manhattan, KS, 66506, USA

Abstract

Of the students enrolled in upper-level Electrical & Computer Engineering (EECE) courses at Kansas State University (KSU), a percentage consistently struggles with concepts from earlier calculus and differential equations courses. This raises issues regarding how much mathematical knowledge students retain and how they transfer this knowledge to follow-on courses. In recent semesters, the KSU Department of Mathematics has utilized automated online tools to generate homework problems and assess student performance. This paper describes an extension of that approach to the Linear Systems course in the KSU Department of Electrical & Computer Engineering. This online suite utilizes PHP, HTML, Java, and PostgreSQL to generate and assess homework problems in the areas of complex numbers, signals, transient response, Fourier series, and Fourier transforms. Features and benefits of this approach include a visually appealing user interface, custom problem sets for each student, online help, immediate score feedback, problem solutions, practice problems, and the opportunity for a student to rework categories of problems until they receive their desired score. From an assessment standpoint, the resulting database offers opportunities to correlate module scores with scores received on other online modules, projects, or exams, where scores can be aggregated or associated with specific problems. Cross-semester comparisons can also be performed. Additional parameters such as completion date/time, the number of attempts per module, the location of the student's machine, and the time required to complete an exercise provide a rich information set for understanding student work habits. The ultimate goal is to close the assessment loop and improve course content based upon previous semester analyses. Early surveys and anecdotal results indicate that student response is generally positive but is subject to software problems typical of a new software release.

Introduction

Linear Systems (EECE 512) is an upper-level engineering course taken by Electrical Engineering and Computer Engineering students at Kansas State University (KSU). This course addresses the mathematical and computational tools necessary to analyze signals in both the time and frequency domains. While calculus and differential equations courses are prerequisites for Linear Systems, a considerable percentage of the students enrolled in this class consistently struggles with concepts that rely upon these mathematical foundations. This raises issues of how much students actually learn in their earlier mathematics courses and what portion of that knowledge they retain as they transition into their upper-level engineering courses.

Transfer of knowledge from semester to semester is difficult to track and assess for multiple reasons. First, detailed records of student performance, especially on a per-problem basis, rarely exist. It is difficult to assess whether a student has retained a reasonable working knowledge of, for example, integration by parts, if the mathematics problems that addressed this subject were graded but not individually recorded. Overall exam and semester scores can usually be obtained, but their granularity is not such that they provide useful information regarding performance in individual subject areas. Second, different departments (indeed different faculty) have different standards for how they assemble and maintain academic performance data. Assimilating these data into a consistent picture of academic accomplishment is a daunting task, even with the assistance of today's computational database and data mining tools. Third, overall student populations change from semester to semester. Students may take courses that require math skills at different stages in their curriculum, and long-term retention can be more of a factor for some students than for others. This makes aggregate assessments of mathematics knowledge retention difficult. This situation is exacerbated by the fact that student learning in Linear Systems is not simply a result of how much mathematical knowledge students retain: it also depends on the interpersonal dynamics between students and faculty and the resultant learning environment that these foster.

To understand semester-to-semester retention of mathematics knowledge, improvements are needed in two areas: (1) tracking systems for both homework and exams that offer better granularity than current systems and (2) formalized, consensus-based plans for how these data will be acquired and stored so that they are beneficial for follow-on assessments. Linear Systems courses at KSU typically support 50 students per semester, while mathematics lecture courses can involve several hundred students per semester. These numbers imply that tracking and recording student performance on a per-problem basis would be greatly facilitated by the use of computer-based learning tools. Complementary goals include the desire to increase the variety of educational resources and venues offered to students so as to keep them engaged. This speaks to interactive learning tools, Internet-based resources (e.g., for students that prefer to work at home), active learning in the classroom,[1] and learning communities that allow students with common academic and personal needs to seek out one another. Additionally, with the increased pressure on academic institutions to provide services given reduced financial resources, tools are needed that automate the homework assignment, distribution, assessment, and recording process.

Computer-aided instruction tools offer the potential to address the issues of problem generation (for both homework and exams), performance assessment, record storage, performance tracking, and decreased resource availability. Computer aided instruction is already widely used in secondary engineering education,[2, 3] and tools from universities and book publishers are now available that offer online alternatives to traditional homework,[4-6] which is the thrust of the online system discussed in this paper. Various studies have reported research results, for example, on the design and development of Internet-based instruction tools. As Alexander[7] stated, the Internet provides an opportunity to realize previously unattainable learning experiences for students, whether these tools embody interactive tutoring systems[8-10] or passive delivery methods.[11,12] Mohamed and Rinky[13] studied distributed passive learning (DPL) versus distributed interactive learning (DIL) web-based environments and showed that the DIL environment is generally superior to the DPL environment in terms of both the learning process and the learning outcome. In addition to the interactive element offered by computer-based environments, they also have the potential to immediately assess student learning, which is consistent with the American Association for Higher Education's nine principles of good practice for assessing student learning.[14]

In recent years, the KSU Department of Mathematics has developed and utilized online tools to generate and assess homework problems in trigonometry, calculus, and differential equations courses.[15] This paper presents an extension of that work applied to a Linear Systems course in the KSU EECE Department. As time progresses, an increasing percentage of linear systems students (~50% in the most recent Linear Systems class) have also utilized the homework generation modules in earlier KSU mathematics courses. As implied earlier, the broad goals of this work are two-fold: (1) to provide computer-based education tools that improve learning and (2) to generate assessment data that can be correlated with data from present and previous semesters. These data may shed some light on what mathematical knowledge students most readily retain and what topics require greater emphasis in prerequisite courses. This paper addresses the first broad goal, describing how the online system is designed and summarizing student responses from its first two semesters of use (Spring 2004 and Fall 2004).

Environment Description and Development Approach

Process. The automated homework system addresses eight Linear Systems topic areas and follows the notation used in the course textbook.[16] Problems are similar to those that would normally be assigned as written exercises. When a student logs in with a user name and password, they request a new problem set, which is then uniquely created for them using randomly generated but bounded parameters. Once their problems are generated, the student can either work on the exercises at the computer or save their session and take the exercises home, returning to enter and submit their answers in a follow-on session. When a student submits answers for questions that do not require a multiple choice format (i.e., numerical quantities or mathematical expressions), a parser checks the syntax of each field. If the syntax cannot be understood by the system, the user must fix the offending expression(s) before the module will be graded. After the answers are graded, the computer shows the student which problems were correct or incorrect and provides a total score for the module. Detailed instructions for problem solutions (i.e., worked problems) are provided if desired. A student can repeat a module as many times as desired before the due date, although new problems are generated with each repeated module. The highest score received is stored in the database as the final homework grade. For most modules, a student does not need to repeat the entire module if a perfect score is not obtained on a previous attempt. Only the types of problems which were incorrectly solved must be repeated to receive additional credit. Once the due date for the module has passed, students can continue to work problems for practice. Figure 1 depicts two Linear Systems students interacting with the online homework system.

Environment Characteristics. These modules are designed in such a way that they take between 30 minutes and two hours to complete. The answers can be entered via an Internet browser on or off campus, since the computer server resides outside of the EECE Department security firewall. With a few minor exceptions, numbers or expressions entered by students must be completely correct in order for them to receive any points on the problem. This is driven by the need for the parser to have a mistake-free expression to analyze, but it has an additional benefit in that the student must understand the problem completely to receive any credit for the solution. All expressions entered by the students must incorporate rational expressions that utilize integers in their numerator and denominator. The parser supports simple mathematical constructs such as *, /, sqrt, pi, cos, sin, and tan. Numerical values such as a half must be entered as “1/2” and not 0.5. The standard programming order of operations is applied to nested groups of characters separated with parentheses.

Figure 1. Linear Systems students interact with the homework generation system.

Module Topics. This automated homework generation environment creates and assesses problems in the following areas:

- **Complex Numbers:** The complex number module addresses mathematical operations such as $(a+jb)\times(c+jd)$ and $(a+jb)/(c+jd)$ that students should have practiced in earlier differential equations and circuit theory courses. Problems also include conversions between Cartesian ($a+jb$) and polar representations $\left(ce^{j\theta}\right)$.
- **Signals:** This module is a multiple choice module where students must choose the graphical representation (from a set of four) that matches the mathematical expression at the top of the page. A signal can be any combination of impulse functions, rectangular functions, exponential waveforms, sinusoids, and unit step functions.
- **Zero Input Response (ZIR):** The ZIR module is the first of two transient response modules. This module seeks the output expression for a system described by a 2nd-order differential equation, where the system contains initial stored energy but has no input

forcing function. Systems with three types of characteristic root pairs are generated: (1) distinct real roots (overdamped), (2) repeated real roots (critically damped), and (3) distinct complex roots (underdamped). For this module and others that follow, a student need only repeat the type of problem that they were unable to solve in the prior session. They do not need to redo all three problems correctly to receive full credit.

- **Unit Impulse Response (UIR):** The UIR module, the second transient response module, seeks the unit impulse response for a system described by a 2^{nd}-order differential equation. The same types of problems are generated as are used in the ZIR module: overdamped, critically damped, and underdamped systems.
- **Fourier Series:** This is actually a collection of three separate modules that address trigonometric, compact trigonometric, and exponential Fourier series, respectively.[16] In each module, a student is given a signal (see the next section) and asked to determine the Fourier coefficients for the given type of series representation. Of the modules presented here, the students find these more difficult because they involve a fair amount of handwritten work prior to entering their coefficient expressions. Once these calculations are complete, they must carefully check the syntax of these expressions prior to submitting their answers for a score.
- **Fourier Transforms:** This module has been developed but not yet used with students in the classroom. Given an analytical function chosen by the computer, this module requires the student to choose reasonable sample rates and signal durations that retain the important information in the signal. It will be used for the first time in the Spring 2005 section of Linear Systems.

Development Technologies. The main page that displays the problem set, receives students' answers, and provides the help links is written mostly in PHP[17] embedded into HTML[18] codes. PHP, or Hypertext Preprocessor, is a server-side scripting language that performs operations such as gathering data from the database or creating on-the-fly images. The grading parser is programmed in Java,[19] and the database is built with PostgreSQL.[20] Some JavaScript[21] code is integrated into the main page; it calls the Java parser function that checks expression syntax.

Sample Interaction: Trigonometric Fourier Series Module

The following example illustrates the procedure involved when working with the online homework generation and assessment modules. A student with a computer at home accesses the online homework system via the Internet using the HTTP link given in class. Each time she logs into the system, her user name and password are required to receive the problem set. For this session, she selects the link for the trigonometric Fourier series module and receives a problem set that includes a signal $f(t)$ like that illustrated in Figure 2. The parameter values for this signal were chosen randomly from a list of reasonable values, so her problem differs from the problems addressed by the other students in the course. Looking through her class notes and textbook, she recalls that any periodic signal, $f(t)$, can be decomposed into a sum of sinusoids, each with a different magnitude, phase, and frequency. She reads the following in her text:

"This trigonometric Fourier series, $f_{TFS}(t)$, is expressed as

$$f_{TFS}(t) = a_0 + \sum_{n=1}^{\infty} a_n \cos(2\pi n f_0 t) + b_n \sin(2\pi n f_0 t)\,.$$

> Here, a_0 is the DC, or average, value of the signal over a given time interval of duration $T_0 = 1/f_0$ seconds ($f_0 = \omega_0/2\pi$ is referred to as the "fundamental" frequency).
>
> $$a_0 = \frac{1}{T_0}\int_{t_1}^{t_1+T_0} f(t)dt$$
>
> The coefficients a_n and b_n represent the magnitudes of the cosines (even functions) and sines (odd functions) that constitute the signal. These coefficients are determined using the following expressions
>
> $$a_n = \frac{2}{T_0}\int_{t_1}^{t_1+T_0} f(t)\cos(n\omega_0 t)dt,\ n = 1,2,3,\ldots$$
>
> and
>
> $$b_n = \frac{2}{T_0}\int_{t_1}^{t_1+T_0} f(t)\sin(n\omega_0 t)dt,\ n = 1,2,3,\ldots,$$
>
> where n is an integer that represents the number of harmonics (in addition to a_0) used to reconstruct the signal. If the original signal, $f(t)$, is not periodic, the Fourier series approximation assumes periodicity outside of the original time range (e.g., for $t < t_1$ and $t > t + T_0$). …"

With these notes in mind, she proceeds to calculate the value for a_0 and the expressions that represent the a_n and b_n coefficients. She notes from inspection that $a_0 = 0$ (the average value of the signal is zero) and $a_n = 0$ (the signal has odd symmetry). She calculates the b_n formula to be

$$b_n = \left(\frac{-4}{n\pi}\right)\cos(n\pi) + \left(\frac{12}{n\pi}\right)\sin\left(\frac{2n\pi}{3}\right) - \sin\left(\frac{n\pi}{3}\right)$$

and carefully types the following expression into the b_n field:

```
-4/(n*pi)*cos(n*pi) + (12/(n*pi))*sin((2/3)*n*pi)-sin((1/3)*n*pi) .
```

When she clicks 'Submit,' the Java parser checks her expressions, finds them to be syntactically adequate, and sends these answers to the grading system. Her scores and the solutions for each problem are immediately returned in a new web page. To her delight, her answers are correct, and her score is immediately saved in the module database. She can now check this task off of her action item list and move on to homework for another course.

Note that, for problems of this nature, students can generate many different types of expressions depending on the technique they use to solve the integral. Rather than checking the correctness of the expression, the grading system evaluates the solution by computing its result for several different values of n. If these results are close to the expected results, the expressions are considered to be correct. Furthermore, in a module of this type, the score for each problem in the module varies according to its difficulty level. In addition, points are assigned to each field based upon how much work is required to generate the response for the field. For instance, as noted above, the a_0 and a_n coefficients for the example shown in Figure 2 are very easy to determine. The student needs to spend much more time solving for the b_n expression. Therefore, b_n is worth more points than a_0 or a_n.

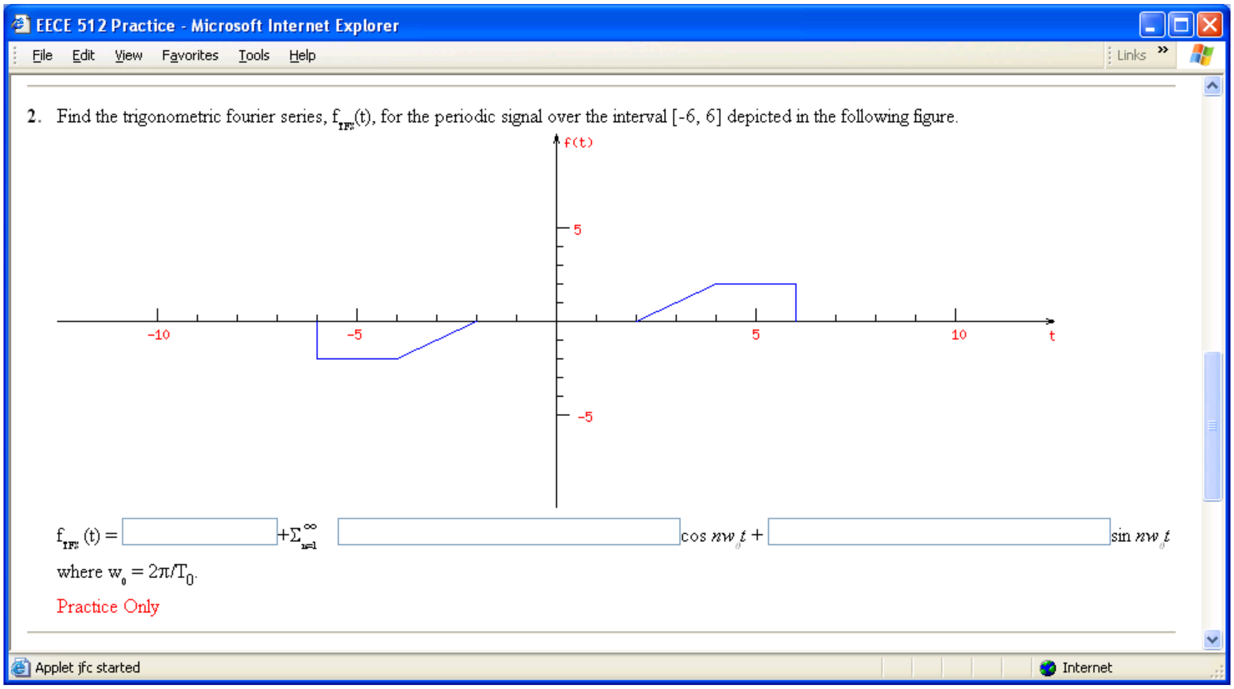

Figure 2. An example trigonometric Fourier series problem.

If the parser finds an expression syntax problem in one or more fields, the student sees a pop-up window like that depicted in Figure 3. Once the student corrects these errors, the expressions are sent to the grading module for assessment. This results in a new web page, as illustrated in Figure 4. As depicted in this figure, if the student had entered values of zero into every expression field and then selected 'Submit,' they would have received credit for a_0 and a_n, but not b_n. Note that the page with the grading results has links to each worked problem. For this Fourier series example, choosing that link would result in another new web page, illustrated in Figure 5. These worked solutions are very detailed and are arranged in two parts: (1) a general set of guidelines for solving a problem of this nature and (2) a set of specific instructions for working the problem. This has tremendous value for the student and can result in time savings for the instructor, since the method presented online would likely be the method that the student and instructor would discuss were the student to come to see the instructor during office hours.

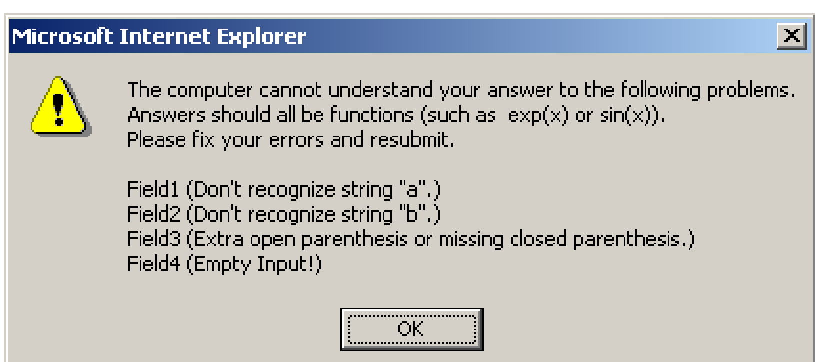

Figure 3. JavaScript syntax pop-up window.

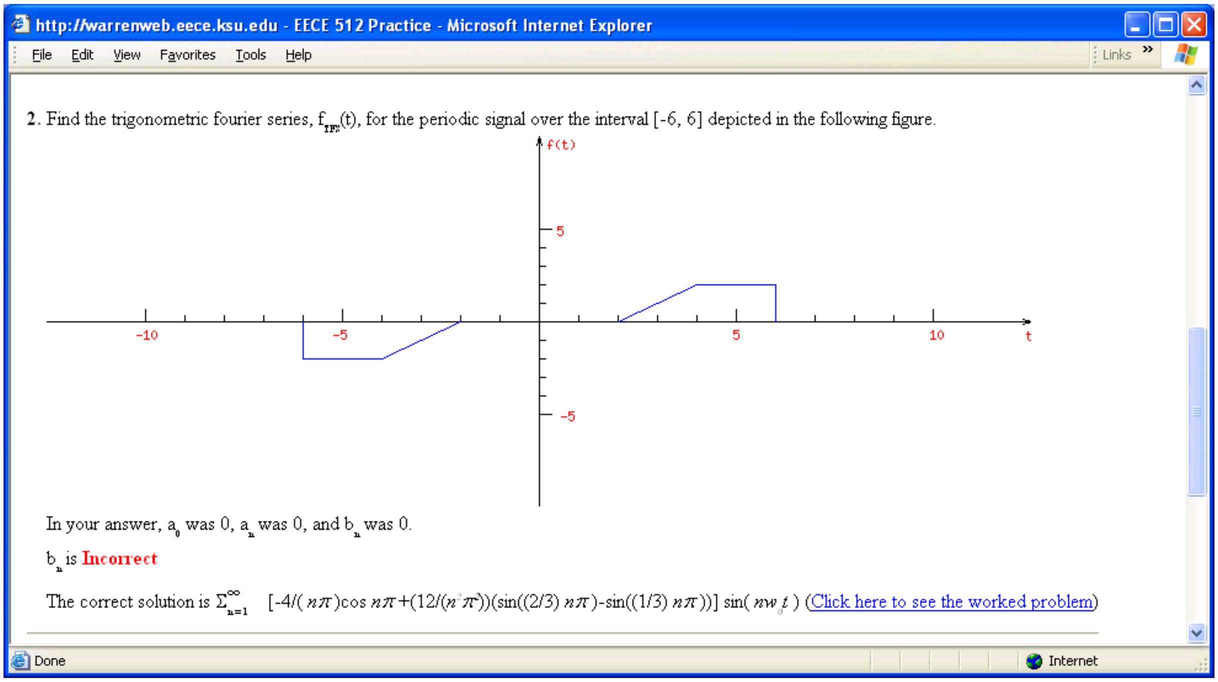

Figure 4. The grade page for the trigonometric Fourier series problem.

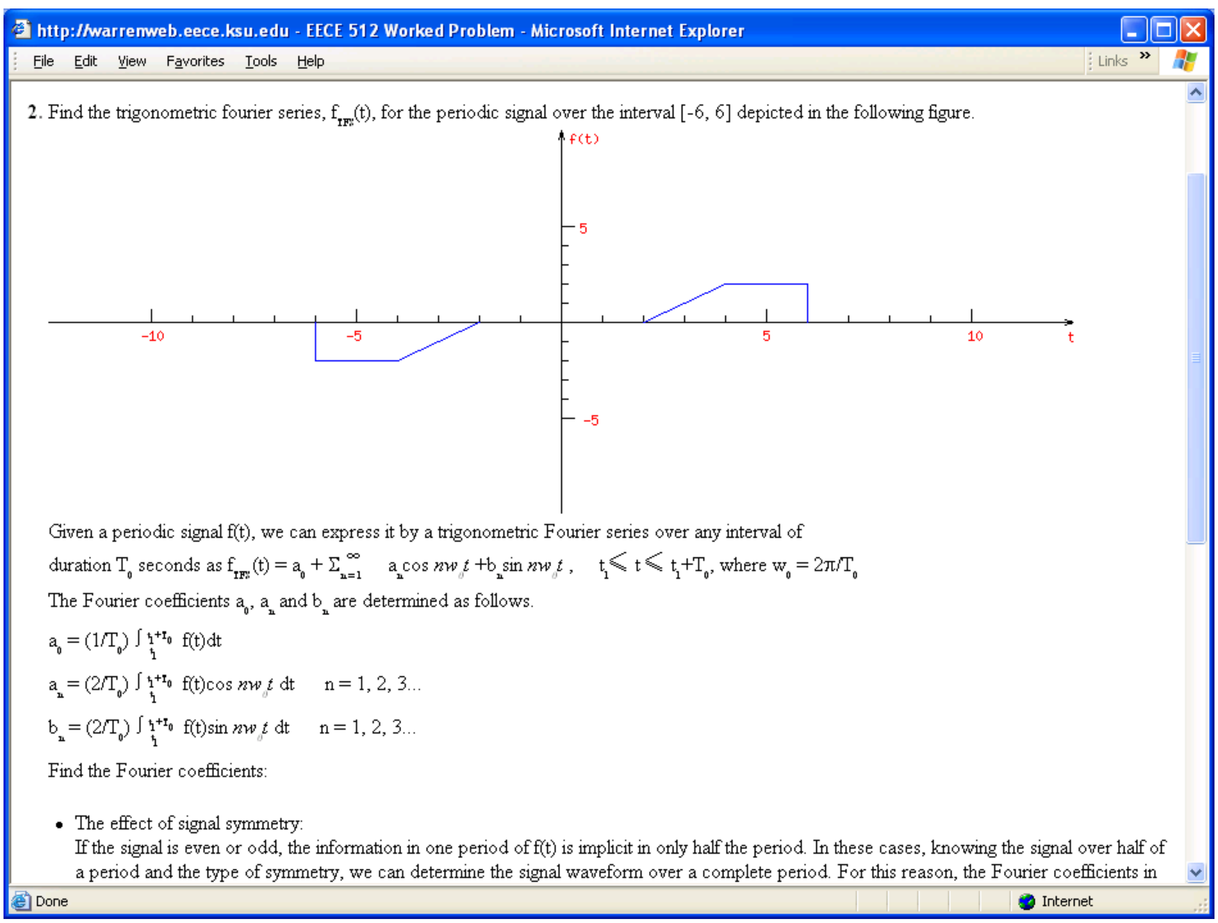

Figure 5. The solution (help) for the trigonometric Fourier series problem.

Early Feedback from the Interactive Learning Modules

Efforts to quantify the impact of this online homework system on learning, academic performance, and knowledge retention are underway, but validated conclusions have not been formulated. However, based upon teaching experience, experience with similar modules in KSU mathematics courses, anecdotal feedback from students, and research already published by others working in this area, we believe that this approach has a high likelihood of success when applied to this secondary student population. To begin the process of assessing the value of these online experiences, students in the Linear Systems class were asked to complete surveys that address the environment presented by these tools as well as their perceptions of the resulting learning experiences. Table 1 and Table 2 list a subset of the results of this survey, which was completed by students in the two back-to-back semesters. The first survey (Spring 2004) was required, and almost every student in this class of 50 students submitted their feedback. In fall 2004, it was not required, so only 60% of the surveys were completed in a class of 40 students.

From both surveys, it is clear that many of the students appreciate instant problem scoring and feedback. Online answers and instant problem help (worked solutions) make students efficient. The ability to attempt modules as many times as they like prior to the deadline also appeals to them. It is interesting to note that a greater percentage of students used the modules for practice in the second semester. One explanation for this could be the additional software bugs present in the online system during the first semester: it simply was not as easy to use. Accessibility and ease of use did not generate the highly positive response that was expected, although a few students mentioned this feature. In both semesters, few students said they liked the random generation of problem sets, a feature that requires students to work individually and is therefore helpful to the instructor regarding the assessment of individual student performance. For the second semester, the modules were updated so that problems of different difficulty levels were clearly separated. Additionally, the feature was added that allows students to only rework the types of problems that they get wrong (rather than resubmitting answers for an entirely new set of problems). This was clearly a good addition to the environment.

Table 1. Features students like the most.

Feature	Spring 2004	Fall 2004
Instant grading and feedback	41.3%	31.5%
Online solutions and worked problems	39.1%	26.3%
Multiple attempts to achieve the desired score	34.7%	31.6%
Practice problems for test preparation	4.3%	26.3%
Accessibility and ease of use	10.8%	10.5%
Randomly generated problem sets	4.3%	0%
Partitioning of problems into different types and difficulty levels	N/A	10.5%

It is also helpful to note the features of the system that the students did not fully appreciate. Some of these are listed in Table 2. As one can imagine, the strict computer grading scheme requires that answers be entered precisely and in a manner the program can interpret. This is unavoidable, since the programmer cannot anticipate the myriad number of ways students might attempt to format expressions if left unguided. The JavaScript syntax checker has helped somewhat in this regard, since it locates mismatched/missing parentheses and unrecognizable variables. Multiple choice questions can address this issue, but they are not as effective from a learning standpoint. Since problem sets can be submitted until a student achieves their desired score, it would simply be too easy for a student to obtain scores that do not clearly reflect their level of understanding. In fact, there is not currently a good way to make computer homework feel like the same experience as handwritten homework. This speaks to one feature about which students consistently complain: an "all or nothing" grading system. In handwritten homework, when a student gets little or nothing correct but still shows some work, some graders will assign partial credit for the problem. However, in the computer-based system, no interim work is sought, so no partial credit is given. Note that some problems with multiple fields do offer the possibility for 'partial credit.' Complaints regarding software bugs and confusing question wording have been significantly reduced in the second semester.

Table 2. Features students like the least.

Feature	Spring 2004	Fall 2004
Picky syntax	45.7%	57.9%
No partial credit	26.1%	37%
Software bugs	26.1%	5.5%
Question wording and help (can be confusing)	10.8%	0%

In addition to offering the potential for improved student learning, automated systems of this nature also allow the educator to track all student interactions with the system via database queries. Instructors can learn how many attempts students made on a given module, when students started doing their homework relative to the deadline, how long students took to finish a problem set, and other good information which is traditionally unavailable. The online system also frees the instructor from the burden of designing and assessing problem-solving homework: this can be especially appealing for very large classes. In addition, every problem set is randomly generated, making the student assume more responsibility for their work. Finally, the detailed help offered by the worked problems can alleviate some of the burden normally imposed on instructors during office hours.

Conclusions

This paper presented an automated, online system for generating and assessing homework in a linear systems course. This capability is an extension of computer-based environments already utilized in the KSU Department of Mathematics. The overall goals of this effort were two-fold: (1) to provide computer-based education tools that offer the potential to improve learning and (2) to generate assessment data that can be correlated with data from present and previous semesters. We hypothesize that these data may shed light on what mathematical knowledge students most readily retain and what topics require greater emphasis in prerequisite courses.

Early indications are that students have a generally positive experience with the online homework process but dislike the picky nature of the system and the accountability that it imposes. From an educator's perspective, automated assessment tools offer the ability to track student performance in ways that could not be realized previously. The cost incurred is the upfront code development time and the maintenance required during the semester, although these drawbacks could be easily offset by the benefits achieved with large course sections.

Future work will involve a better quantified assessment of the learning improvement offered by these resources, which will require more numerous correlations between module scores and test performance and further analysis of information gathered from database queries. Other work will include inter-semester performance comparisons as well as the extension of this capability to other engineering courses.

Acknowledgements

Portions of this material are based upon work supported by the National Science Foundation under grant DUE–0206943. Opinions, findings, conclusions, or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the NSF.

References

[1] David W. Johnson, Roger T. Johnson, and Karl A. Smith. *Active Learning: Cooperation in the College Classroom*. Edina, MN: Interaction Book Company, 1998, 0-939603-14-4.
[2] Striegel, A. and D.T. Rover. "Problem-Based Learning in an Introductory Computer Engineering Course," presented at 32nd Annual Frontiers in Education Conference, 2002, 2002.
[3] Jones, Sandra. "Designing Flexible Learning Opportunities Through a Non-Linear Collaborative Process," *ultiBASE*, vol. November, pp.
[4] "Homework Service Overview," The University of Texas at Austin, https://hw.utexas.edu/overview.html.
[5] Bonham, Scott and Robert Beichner. "Online homework: Does It Make a Difference?," http://physics.wku.edu/~bonham/Publications/HomeworkCompare.pdf.
[6] Hayt, William H., Jack E. Kemmerly, and Steven M. Durbin. *Engineering Circuit Analysis*,
[7] Alexander, Shirley. "Teaching and Learning on the World Wide Web," 1995, http://ausweb.scu.edu.au/aw95/education2/alexander/.
[8] II, Burks Oakley. "A Virtual Classroom Approach to Teaching Circuit Analysis," *IEEE Transactions on Education*, vol. 39, Aug 1996, pp. 287-296.
[9] Lu, Han-Pang Huang and Chiou-Hwa. "Java-Based Distance Learning Environment for Electronic Instruments," *IEEE Transactions on Education*, vol. 46, Feb 2003, pp. 88- 94.
[10] Titcomb, S.L. "Computer-Based Interactive Tutorials for Electrical Engineering Laboratory Instruction," presented at 27th Annual Conference Frontiers in Education Conference, 1997. Proceedings 'Teaching and Learning in an Era of Change'. 1997.
[11] Gronhovd, Sudhir I. Mehta and Sandy M. "Instrumentation and Communication Modules on CD-ROM's for Enriching Engineering Education," *IEEE Transactions on Education*, vol. 39, Aug. 1996, pp. 304-308.
[12] Schodorf, J.B. Yoder, M.A. McClellan, J.H. Schafer, R.W. "Using Multimedia to Teach the Theory of Digital Multimedia Signals," *IEEE Transactions on Education*, vol. 39, Aug 1996, pp. 336-341.
[13] Lam, Mohamed Khalifa and Rinky. "Web-Based learning: Effects on Learning Process and Outcome," *IEEE Transactions on Education*, vol. 45, Nov 2002, pp. 350- 356.
[14] "9 Principles of Good Practice for Assessing Student Learning," American Association for Higher Education, http://www.aahe.org/assessment/principl.htm.
[15] Bennett, Andrew G. "Mathematics Education and the Internet," 1998, http://www.math.ksu.edu/meati/.
[16] Lathi, B.P. *Signal Processing & Linear Systems*, First ed: Berkeley-Cambridge Press, Carmichael, CA, 1998, 0-195219-17-1.
[17] "PHP," 2004, http://www.php.net/.
[18] "HyperText Markup Language (HTML) Home Page," W3C, 2004, http://www.w3.org/MarkUp/.
[19] "Java Technology," Sun Microsystems, 2004, http://java.sun.com/.
[20] "PostgreSQL," 2004, http://www3.sk.postgresql.org/.
[21] "JavaScript.com," Sun Microsystems, 2004, http://www.javascript.com/.

Biographical Information

Yong Yang is a Masters student in the Department of Electrical & Computer Engineering at Kansas State University. He received his B.S. in Mechanical Engineering and Automation from Shanghai Jiao Tong University in July, 2000. His areas of research interest include computer aided education, signal processing and control, networking, and robotics.

Andrew Bennett is a Professor in the Department of Mathematics at Kansas State University. He teaches undergraduate and graduate courses in mathematics, including trigonometry, calculus, and differential equations. Dr. Bennett's current research focuses on enhancing education and assessing student learning and mathematical knowledge retention via emerging technology tools.

Steve Warren is an Associate Professor of Electrical & Computer Engineering at Kansas State University. He teaches courses in linear systems, computer graphics, biomedical instrumentation, and scientific computing. Dr. Warren manages the KSU *Medical Component Design Laboratory*, and his research focuses on plug-and-play, wearable systems for telemedicine.