Creating Analytic Online Homework for Digital Signal Processing

H. J. Trussell and D. Baron

An article by W. L. Everitt in the 1962 50th Anniversary Issue of the Proceedings of IEEE, entitled, "Engineering Education - Circa 2012 A.D.," was one of many predictive articles that appeared in that issue [1]. One of Everitt's observations was the distinction between training and education. He then predicted that in the future training will be done primarily with computers. He then remarks, "Relieved of the necessity of spending most of their time on the training function, devoted teachers will be able to concentrate their efforts on 'education.' "

The distinction Everitt makes can be briefly presented to define what we are trying to accomplish. Training is concerned with learning methods, vocabularies, computational skills and mathematical manipulations. It is reinforced by solving problems with definite solutions or measures of performance. Education is broader and is concerned with creativity, understanding and the ability to apply the trained skills to meet new situations.

We are now reaching the point of Everitt's prediction, where we can create and grade problems on the computer that are similar to those training problems used in engineering texts. With the increasing use of the internet for course content delivery, both for standard on-campus courses and massive open online courses (MOOCs), this capability becomes critically important.

For engineering, there are several online course/learning management systems that are in use. Included among these are Moodle, WebAssign [2], LON-CAPA [3], WeBWorK [4], MasteringEngineering (Pearson Publishing) [5] and Connect Engineering (McGraw-Hill Publishing) [6]. All of these systems manage the administrative aspects of classes, including maintaining gradebooks, reporting grades, posting assignments and messages, tracking student progress, presenting and grading problems. Currently, only WeBWorK, WebAssign, and LON-CAPA allow the creation of the analytical problems that are required for engineering courses.

Of the three systems that can create the required problems, WebAssign is in most widespread use, at about 2300 institutions. WeBWork and LON-CAPA are used at about 300 and 150 institutions, respectively. It is unclear how many instructors use which features of the systems. None of the systems have found extensive application to electrical and computer engineering education. We will give a brief comparison of the systems later.

We used WeBWorK to help realize Everitt's prediction. This system was available at NC State University (NCSU) and was cost free for both students and instructors. WebAssign is available at NCSU, but entails costs to the students. Since WeBWorK was used by our Department of Mathematics, there was good local experience with problems similar to those for electrical and computer engineering. LON-CAPA is not available at NCSU. Since the administrative capabilities of the systems are about the same, we limit our discussion to introducing the types of problems that can be presented to electrical and computer engineering students in general and signal processing students in particular.

Realistic Problems

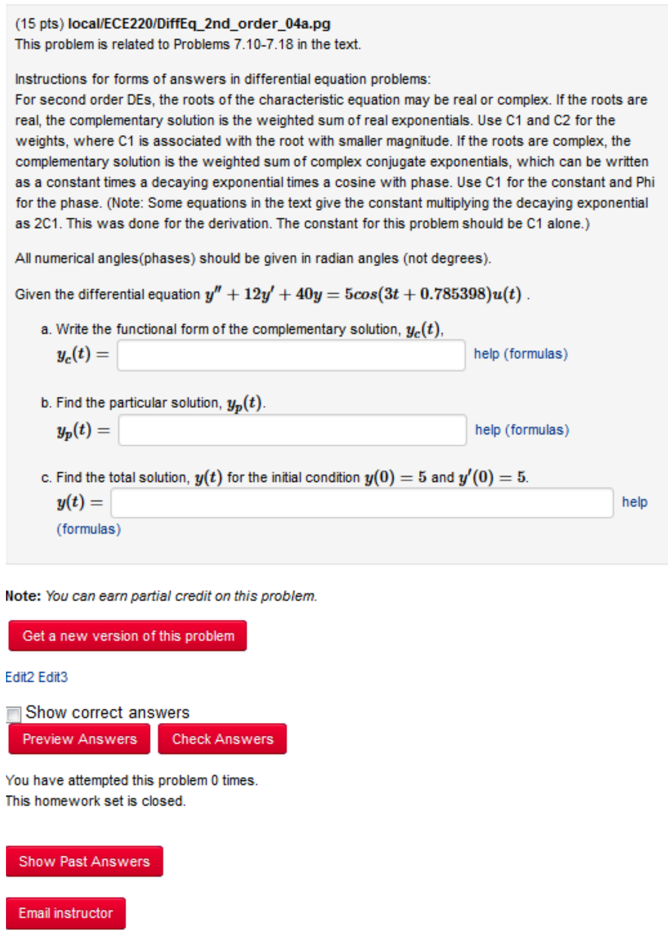
All of the most widely used online homework and testing packages allow the common types of questions, including true/false or yes/no, simple calculated numerical answer, multiple choice or multiple select (several possible correct answers), matching, fill-in-the-blank, etc. However, a problem form of major interest to most engineers is one that asks the student to find a mathematical relationship between the input and output of a system. This relationship is easily written as a formula or equation.

In the Fall 2013 semester, we used WeBWorK extensively for a sophomore electrical and computer engineering class at NCSU, ECE220. The course covers basic math, beyond calculus, needed for Electrical and Computer Engineering (ECE). This online homework approach was used in subsequent semesters of this course with quite positive evaluations from the students. To follow on this success, we developed problems for our senior level digital signal processing course, ECE421.

To illustrate the type of problems that we created, let us consider the solution to second order differential equations, which could represent a variety of circuit, control or signal processing problems. We often want to see the various steps that a student makes to obtain a solution. For example, with given initial conditions, we ask the student for the form of the complementary solution, the particular solution and the total solution. All of these are functional forms and the complementary solution has unknown constants that are determined only in the final stages of solving for the total solution. Such a differential equation problem is shown in Figure 1.

The student enters the answers using a format that resembles MATLAB®. Figure 2 shows the student's input and its interpretation by the parser. The package will give hints as to problems with the format, e.g., unbalanced parentheses, unknown variables.

Problems can be created in a tutorial mode that gives the student step-by-step instructions on solving the problems. Alternatively, they can be written to check only the final answer. It is common that instructors wish to evaluate the students' written solutions to determine their understanding and pinpoint where mistakes are made. The automated approach can check the steps of the solution, if desired, as in the case of Figure 1. It cannot indicate what error the student made, if she missed a part of the problem. However, we have found that by giving the students immediate feedback that answers are incorrect, students are usually able to determine their mistakes and proceed to the correct answer. This would seem preferable to giving detailed feedback on the error several days after the student hands in a written solution.



(15 pts) **local/ECE220/DiffEq_2nd_order_04a.pg**
This problem is related to Problems 7.10-7.18 in the text.

Instructions for forms of answers in differential equation problems:
For second order DEs, the roots of the characteristic equation may be real or complex. If the roots are real, the complementary solution is the weighted sum of real exponentials. Use C1 and C2 for the weights, where C1 is associated with the root with smaller magnitude. If the roots are complex, the complementary solution is the weighted sum of complex conjugate exponentials, which can be written as a constant times a decaying exponential times a cosine with phase. Use C1 for the constant and Phi for the phase. (Note: Some equations in the text give the constant multiplying the decaying exponential as 2C1. This was done for the derivation. The constant for this problem should be C1 alone.)

All numerical angles(phases) should be given in radian angles (not degrees).

Given the differential equation $y'' + 12y' + 40y = 5cos(3t + 0.785398)u(t)$.

a. Write the functional form of the complementary solution, $y_c(t)$,
$y_c(t) =$ help (formulas)

b. Find the particular solution, $y_p(t)$.
$y_p(t) =$ help (formulas)

c. Find the total solution, $y(t)$ for the initial condition $y(0) = 5$ and $y'(0) = 5$.
$y(t) =$ help (formulas)

Note: *You can earn partial credit on this problem.*

Get a new version of this problem

Edit2 Edit3

Show correct answers

Preview Answers Check Answers

You have attempted this problem 0 times.
This homework set is closed.

Show Past Answers

Email instructor

Figure 1: WeBWorK Problem Format

Examples and Practice

The Digital Signal Processing (DSP) course at NCSU uses the text by Proakis and Manolakis [7]. We developed problems based on those in the text and have referenced the text problem number in the problem description. This will also allow the reader to observe our translation from text problems to realistic online problems.

A typical z-transform problem is shown in Figure 3.

A discrete Fourier transform problem is shown in Figure 4.

A problem that has a purely symbolic answer is shown in Figure 5.

In these problems, the various parameters are randomized, but guaranteed to give reasonable formulations and answers. The randomization of the parameters

Entered	Answer Preview
C1*exp(-6*t)*cos(2*t+Phi)*u(t)	$C_1 \exp(-6t)\cos(2t + Phi)u(t)$
0.1052*cos(3*t-0.07449)*u(t)	$0.1052\cos(3t - 0.07449)u(t)$
[17.857*exp(-6*t)*cos(2*t-1.2931)+0.1052*cos(3*t-0.07449)]*u(t)	$(17.857\exp(-6t)\cos(2t - 1.2931) + 0.1052\cos(3t - 0.07449))u(t)$

Figure 2: WeBWorK Problem Format

This problem is related to Problem 3.7 (page 215) in the text.

Consider the signal given by

$$x(n) = \begin{cases} (\frac{1}{2})^n & n \geq 0 \\ (\frac{2}{8})^{-n} & n < 0 \end{cases}$$

and the impulse response

$$h(n) = \begin{cases} (\frac{6}{9})^n & n \geq 0 \\ 0 & n < 0 \end{cases}$$

Compute the convolution of the signal with the impulse response, $y(n) = x(n) * h(n)$.

For $n \geq 0, y(n) =$ help (formulas)

For $n < 0, y(n) =$ help (formulas)

Determine its region of convergence (ROC). Write the ROC as an interval.

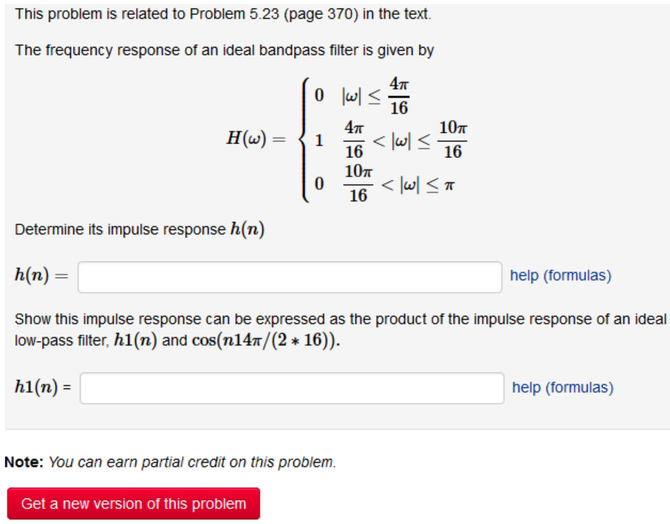
ROC = help (intervals)

Figure 3: z-transform Problem

has the advantage of presenting each student with a different problem. The randomization also allows the students to use the same problem for additional practice. Note the "Get a new version of this problem" button at the bottom of Figure 4. This button appears after the student has submitted his or her answer to the assignment. The online solutions for the problems are written to use the parameters appropriate to each student. This allows students to see exactly how their problems are solved without having to translate from a common example in the text.

Each answer is noted as correct or not. The instructor sets the number of attempts that the student may use prior to submission of the answer for grading. This checking has been well received by the students and helps them to identify their own errors. The students can use the "preview" button, Figure 1, to check their answers for typos.

At the bottom of the problem, see Figure 1, there is a button to "email the instructor." This allows the student who is stuck or has questions to ask the instructor for help. The student explains his problem in the email, which is


This problem is related to Problem 5.23 (page 370) in the text.

The frequency response of an ideal bandpass filter is given by

$$H(\omega) = \begin{cases} 0 & |\omega| \le \frac{4\pi}{16} \\ 1 & \frac{4\pi}{16} < |\omega| \le \frac{10\pi}{16} \\ 0 & \frac{10\pi}{16} < |\omega| \le \pi \end{cases}$$

Determine its impulse response $h(n)$

$h(n) =$ help (formulas)

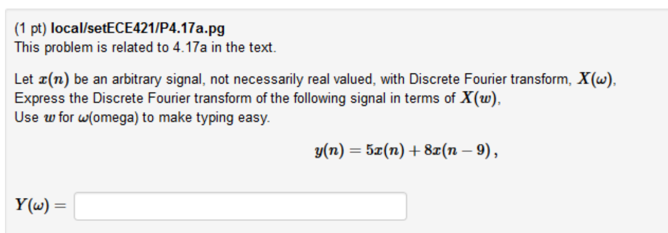
Show this impulse response can be expressed as the product of the impulse response of an ideal low-pass filter, $h1(n)$ and $\cos(n14\pi/(2*16))$.

$h1(n) =$ help (formulas)

Note: *You can earn partial credit on this problem.*

Get a new version of this problem

Figure 4: Discrete Fourier Transform Problem


(1 pt) **local/setECE421/P4.17a.pg**
This problem is related to 4.17a in the text.

Let $x(n)$ be an arbitrary signal, not necessarily real valued, with Discrete Fourier transform, $X(\omega)$,
Express the Discrete Fourier transform of the following signal in terms of $X(w)$,
Use w for ω(omega) to make typing easy.

$$y(n) = 5x(n) + 8x(n-9),$$

$Y(\omega) =$

Figure 5: Purely Symbolic Answer Problem

sent to the instructor and support group (teaching assistants or other faculty). The email contains a link to the student's problem. This allows the instructor to see exactly what the student has entered and to formulate an answer to the student's question.

Graphics

Problems for DSP homework and tests frequently refer to figures, such as block diagrams, time-domain signals, frequency-domain spectra, pole-zero plots, etc. As is common for most packages, WeBWorK allows the insertion of static graphics that are generated off-line. In addition, WeBWorK allows dynamic graphics where the graph depends on the random parameters for a specific problem. This enables many more types of realistic problems that ask the student to obtain information from the graph.

A frequent assignment is to sketch a graph of a function that has been obtained. Plots might be waveforms, frequency responses, regions of convergence and the like. Currently, there is not a simple way for the student to enter a sketch of the function that can be evaluated for correctness. There is an app that can be embedded in WeBWorK that allows the student to input a curve in a graph with labeled axes. The student enters the curve using the mouse or with a tablet. The goodness of the student's sketch is measured by a combination of maximum pointwise error and cumulative error. An example is shown in Figure 6, where the student is asked to sketch the derivative of the function in the top graph. The differences that determine the error are shown in the green and yellow areas. Note the "smooth" button allows entering rough sketches, followed by smoothing to assist in producing better curves. While this indicates a promising capability, we have not yet incorporated it into our problems.

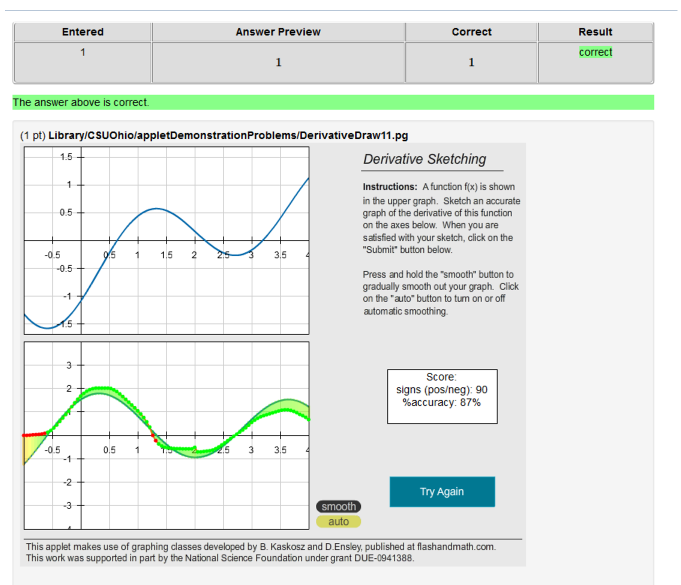


Figure 6: Student Sketches Graph

We note that both WebAssign and LON-CAPA allow student generated graphics. The examples that we have seen are limited to very simple manipulations. These would have difficulty sketching a sinusoid or exponential. This ability will likely improve for all three systems.

Programming the Problems

Although the new problem generation and answer checking packages are able to handle a large variety of practical problems associated with undergraduate electrical and computer engineering, a major hurdle for the current packages is the steep learning curve required to generate online problems for a particular course.

Our experience in developing problems is with WeBWorK. This package is based on the PERL computer language and has been greatly augmented by the WeBWorK developers to handle the many cases required for mathematical problem development. While using templates from the large online library helped, there were many things that we wanted to do for engineering problems that were sufficiently different from the available math-based problems to cause difficulties. Notations and problem motivations were typical of these. Fortunately, the developers had already solved the j vs. i notation problem for $\sqrt{-1}$. We needed to insert the change into all of our problems.

The difficulty of developing problems is somewhat alleviated by the libraries of problems available from those who have learned the systems and developed their own problem sets. Thus, we often used problems from the WeBWorK Open Library as a starting point for our problems. An example of the code for Figure 4 is given below. The language is basically Perl. The # symbol denotes a comment.

```
# DESCRIPTION
# Problem from 'Digital Signal Processing' Proakis and Manolakis, 4th ed.
# WeBWorK problem written by Joel Trussell, <hjt@ncsu.edu>
# ENDDESCRIPTION

## DBsubject(Digital Signal Processing)
## DBsection(Problems)
## Institution(North Carolina State University)
## Author(H. J. Trussell)
## TitleText1('Digital Signal Processing: Principles, Algorithms and Applications')
## AuthorText1('Proakis and Manolakis)
## EditionText1('4')
## Problem1('5.23)
##############################
#  Initialization

DOCUMENT();
# load macro functions for Webwork
loadMacros(
"PGstandard.pl",
"MathObjects.pl",
"AnswerFormatHelp.pl",
"PGcourse.pl"
```

```
);

TEXT(beginproblem());
#############################
Context("Numeric");  # webwork context for math evaluations
Context()->variables->are( n=>"Real" );
# set function tolerance parameters
# always use absolute tolerance with signal processing problems
# since the random test points may result in values near zero
Context()->flags->set(  tolerance => 0.001,  tolType => "absolute");

$d1 = 16;  # base denominator, the following are numerators.
$n1 = random(2, 8, 2);
$n2 = random($n1+2, 14, 2);
$n4 = ($n2-$n1);
$n3 = ($n2+$n1);

# describe frequency response to take the inverse DFT of
#  formula  int_(-b)^b exp(j*w*n) dw -  int_(-a)^a exp(j*w*n) dw
#   (2/n)*2*sin((b-a)*n/2)*cos((b+a)*n/2)
#  a = n1*pi/d1; b = n2*pi/d1
#  write using fractions

$text_formula = "2*sin(($n2-$n1)*pi*n/(2*$d1))*cos(($n2+$n1)*pi*n/(2*$d1))/(pi*n)";
$answera = Formula($text_formula)->reduce;
# the reduce feature eliminates terms with zero multipliers and
# a coefficient of unity - among other things
$answerb = Formula("2*sin(($n2-$n1)*pi*n/(2*$d1))/(pi*n)")->reduce;

#############################
#  Main text1

Context()->texStrings;
BEGIN_TEXT
This problem is related to Problem 5.23 (page 370) in the text.
$PAR
The frequency response of an ideal bandpass filter is given by
 \[ H(\omega)  = \left\lbrace \begin{array}{ l l }
 0 & |\omega | \leq \frac{$n1 \pi}{$d1} \\
1 & \frac{$n1 \pi}{$d1} <  |\omega |  \leq  \frac{$n2 \pi}{$d1} \\
0 &  \frac{$n2 \pi}{$d1} <  |\omega |  \leq   \pi  \end{array} \right. \]
$PAR Determine its impulse response \(h(n)\)
$BR
\( h(n) = \)  \{ ans_rule(60) \} \{ AnswerFormatHelp("formulas") \}

$PAR Show this impulse response can be expressed as the product
```

```
of the impulse response of an ideal low-pass filter, \(h1(n)\) and
\(\cos(n $n3 \pi / (2*$d1) ).\)
$BR
 \( h1(n) \) = \{ ans_rule(60) \}  \{ AnswerFormatHelp("formulas") \}
END_TEXT
Context()->normalStrings;

##############################
#  Answer evaluation1

$showPartialCorrectAnswers = 1; # allows students partial credit

ANS( $answera->cmp() );  # answer checker for first blank
ANS( $answerb->cmp() );   # answer checker for second  blank

ENDDOCUMENT();
```

Comparison of Systems and Resources

In general, WeBWork, WebAssign and LON-CAPA have the ability to create similar problems that can be used for signal processing courses. We have extensive experience with WeBWorK, as mentioned earlier. This allows us to indicate the type of problems that can be created and are useful for our courses. We have verified that most of the features that we use in WeBWork are available in the other systems. The major features that we use include control of randomization of parameters, analytic formulas as answers, dynamic graphics, complex mathematics and easy input of complicated mathematical formulas. It is important that the instructor can interact with the students having questions and can see the exact problem and answer of the student.

There are major differences in the "look and feel" of the systems. Having extensive experience with WeBWorK and limited experience with the other two, it seems that all require much learning to use effectively. Both WebAssign and LON-CAPA have Graphical User Interfaces (GUIs) that allow the novice user to more easily create simple problems. However, using the GUIs may slow the experienced user when creating more complicated problems. We would liken the difference to people writing engineering papers in MSWord and Latex. Once you learn the language, it is more efficient to write technical papers in Latex.

The languages of the three systems are different. WeBWork is Perl based. WebAssign and LON-CAPA use HTML or XML. For these two, coding in these languages is almost required for complicated problems and associated answer checking. All systems allow the user to create answer checkers for specific problems with unique requirements.

It is possible to place restrictions on the form of the answers in all three systems. For example, in our course, we require an answer that contains a sinusoid to be entered in the form of a cosine with phase. We can disallow the use of the sine-plus-cosine form or the sum of complex exponentials form. This is relatively easy in WeBWorK by disabling a function with a simple command.

Documentation is important for all software systems. Since WebAssign and LON-CAPA are commercial systems, their documentations are very good. There is much available on the web. WeBWorK can be frustrating because of poor documentation. There are web pages for many topics and users can get additional information from the user forums. One approach to overcome the documentation problem is to use a search engine with the input: webwork topic. Since WeBWorK is open source, this works pretty well. This may not work as well with the commercial packages.

WebAssign has a large collection of texts that are keyed to its use. They cover a range of technical areas in the sciences and engineering, although engineering is represented by texts in only materials, mechanics and circuits. Mathematics has over 400 texts and thousands of problems ranging from elementary to differential equations. We know of no texts that are keyed to WeBWorK or LON-CAPA.

Since it is easy to start coding problems by using a similar problem as a template, the availability of user libraries is very helpful. WeBWorK has an open problem library that is primarily mathematics, as is expected, since the package is supported by the Mathematical Association of America. There are small problem sets in physics, electric circuits and mechanics. Many of the problem sets are keyed to specific texts like our examples of DSP problems. There is current work in developing problems for electrical and computer engineering at Louisiana Tech University and Northern Arizona University. Because of the nature of the commercial systems, access to previously coded problems is more difficult for WebAssign or LON-CAPA.

Finally, we should mention something about cost. WeBWorK is open source and freely distributed. There is, of course, a small but real cost since the institution must allocate resources to install and maintain the software on their servers. Use of the package is available for a $200-300 cost via the Mathematical Association of America. LON-CAPA is also open-source, similarly to WeBWorK. It is the driver for the commercial venture, CourseWeaver [8]. We are not familiar with the licenses for the commercial systems, WebAssign and CourseWeaver, and must refer the reader to their websites.

Student Evaluations and Comments

In the first semester that we used WeBWorK problems for ECE220, all homework was submitted online. The students were very positive and gave the system a 4.1/5 rating. The course was offered with two sections the following spring and during the summer session. While no formal questions were put on the course evaluation survey, the response via comments to the instructors was quite positive. The evaluation for Fall 2014 yielded a 4.0/5 rating. A trial run was made in ECE421 for the Fall 2014 class. Student comments were again positive. Related problems were used for BME311 (Linear Systems in Biomedical Engineering) in Fall 2014. WeBWorK was expanded to a more tutorial environment, where if students submitted a few wrong answers, they were directed to series of more elementary tutorial problems, before returning to the main problem thread. These students were overwhelmingly positive, giving 85% strongly agree or agree responses to questions on the effectiveness of the tutorial.

The students like the immediate feedback of correctness. They usually are able to determine their errors and submit a correct answer in a few attempts. We allow multiple attempts for analytic problems, since typographical errors are common, as are arithmetic errors. In the cases where the students have problems, they appreciate the easy access to the instructor via the "email" button. We have found the email question load to be quite small. With almost 200 students in ECE220 for Spring 2014, we averaged under nine emails per week. Students were able to use the problems for review for the final exam since the randomization offered practice problems.

Students did not like long problems that had only a single answer. They preferred multiple step problems, where intermediate results were required, similar to Figure 1. Even in the multiple step problems, a similar problem was that each submission counted toward the total number for the problem. This means that taking five attempts for part one reduces the number of attempts for part two by five. Our approach has been to allow proportionately more attempts for multiple step problems. WebAssign allows the instructor to set the number of attempts for each part, but this has its own problems.

While we informed students of the required accuracy of the solutions, students found using four decimal places for answers annoying. In all of the systems, the instructor can assign the type of accuracy required, relative or absolute. For signal processing problems, which have signals and functions that often have values near zero, absolute accuracy is preferred.

Link to Online Demo

We have collected problems used in both ECE220 and ECE421 classes at NCSU in a WeBWorK problem set that can be accessed with a Guest login. This will allow readers to get a better idea of the range of problems currently available, as well as ideas about how they may be modified to fit their particular needs. We have set the permissions to allow the problem code to be viewed for only a few problems. There is also an Orientation Problem set, that contains examples of how to use WeBWork, e.g., input answers, previewing answers, contact the instructor, etc. The link is

`http://webwork-jitar.math.ncsu.edu/webwork2/DemoCourse/.`

Conclusions

Our goal is to demonstrate the ability to create online problems to train engineers in the mathematics of linear systems and signal processing. Such training requires that the student see realistic problems that illustrate the skills to be learned, feedback on the performance of those skills and the ability to continue practicing those skills. We hope that readers will recognize the possibilities and join in the creation of libraries that can be easily used by the majority of signal processing educators.

Authors

H. Joel Trussell (hjt@ncsu.edu) received degrees from Georgia Tech (BS), Florida State (MS) and the University of New Mexico (PhD). He worked at the Los Alamos National Lab, 1969-1980, and is currently a professor in the Electrical and Computer Engineering Department at NC State University, Raleigh, NC.

Dror Baron (barondror@ncsu.edu) received degrees from the Technion – Israel Institute of Technology (B.Sc. and M.Sc.) and the University of Illinois at Urbana-Champaign (Ph.D.), all in electrical engineering. Dr. Baron joined the Electrical and Computer Engineering Department at NC State University in 2010, where he is an assistant professor.

References

[1] W. L. Everitt, "Engineering Education - Circa 2012 A.D.," Proc. IEEE, Vol. 50, No. 5, pp. 571-572, May 1962.

[2] WebAssign http://webassign.com

[3] LON-CAPA Hosting and Support, http://www.courseweaver.com/lc.php

[4] WeBWorK, http://WeBWorK.maa.org

[5] MasteringEngineering, http://www.pearsonmylabandmastering.com/northamerica/masteringengineering/

[6] ConnectEngineering, http://connect.customer.mcgraw-hill.com/

[7] J. G. Proakis and D. G. Manolakis, *Digital Signal Processing: Principles, Algorithms and Applications, 4th Ed.*, Pearson Education, Inc., 2007.

[8] http://www.courseweaver.com/