Seattle
Making Value for Society
122nd ASEE Annual
Conference & Exposition
June 14 - 17, 2015
Seattle, WA

Paper ID #14118

Recent Progress in Step-Based Tutoring for Linear Circuit Analysis Courses

Dr. Brian J Skromme, Arizona State University

Dr. Brian Skromme is a professor in the School of Electrical, Computer, and Energy Engineering and is assistant dean of the Fulton Schools of Engineering at Arizona State University. He holds a Ph.D. in electrical engineering from the University of Illinois at Urbana-Champaign and was a member of technical staff at Bellcore from 1985 to 1989.

Paul Rayes, Arizona State University
Mr. Brian McNamara, Arizona State University
Mr. Xiaoxuan Wang, Arizona State University

Master Student in Electrical Engineering

Dr. Yih-Fang Huang, University of Notre Dame

Dr. Yih-Fang Huang is Professor of Electrical Engineering and Senior Associate Dean for Education and Undergraduate Programs in the College of Engineering. He received his B.S.E.E. degree from National Taiwan University, M.S.E.E. degree from University of Notre Dame, M.A. and Ph.D. from Princeton University. He served as chair of Notre Dame's Electrical Engineering department from 1998 to 2006. His research work employs principles in mathematical statistics to solve signal detection and estimation problems that arise in various applications that include wireless communications, distributed sensor networks and, more recently, smart electric power grid.

Dr. Huang is a Fellow of the Institute of Electrical and Electronic Engineers (IEEE) ('95). He received the Golden Jubilee Medal of the IEEE Circuits and Systems Society in 1999, served as Vice President in 1997-98 and was a Distinguished Lecturer for the same society in 2000-2001. At the University of Notre Dame, he received Presidential Award in 2003, the Electrical Engineering department's Outstanding Teacher Award in 1994 and in 2011, the Rev. Edmund P. Joyce, CSC Award for Excellence in Undergraduate Teaching in 2011, and the Engineering College's Outstanding Teacher of the Year Award in 2013.

In Spring 1993, Dr. Huang received the Toshiba Fellowship and was Toshiba Visiting Professor at Waseda University, Tokyo, Japan. From April to July 2007, he was a visiting professor at the Munich University of Technology, Germany. In Fall, 2007, Dr. Huang was awarded the Fulbright-Nokia scholarship for lectures/research at Helsinki University of Technology in Finland.

Daniel H. Robinson, Colorado State University

Dan Robinson is a Professor in the School of Education at Colorado State University. He received his Ph.D. in Educational Psychology in 1993 from the University of Nebraska where he majored in both learning/cognition and statistics/research. He has taught at Mississippi State University (1993-1997), the University of South Dakota (1997-1998), the University of Louisville (1998-1999), and the University of Texas (1999-2012).

Dr. Robinson serves as the editor of Educational Psychology Review and Associate Editor of the Journal of Educational Psychology. Dr. Robinson has served as an editorial board member of nine refereed international journals: American Educational Research Journal, Contemporary Educational Psychology, Educational Technology, Research, & Development, Journal of Behavioral Education, Journal of Educational Psychology, Journal of Experimental Education, Reading Research and Instruction, Research in the Schools, and The Open Education Journal.

He has published over 100 articles, books, and book chapters, presented over 100 papers at research conferences, and taught over 100 college courses. His research interests include educational technology innovations that may facilitate learning, team-based approaches to learning, and examining trends in articles published in various educational journals and societies. He was a Visiting Fulbright Scholar, Victoria University, Wellington, New Zealand and was named as one of the most published authors in educational psychology journals from 1991-1996, 1997-2002, 1991-2002, and 2003-2008, Contemporary Educational Psychology, 1998, 2004, 2010.

122nd ASEE Annual Conference & Exposition
June 14 - 17, 2015
Seattle, WA
Making Value for Society

Paper ID #14118

Xiang Gao, Arizona State University
Mr. Theodore Thompson

Recent Progress in Step-Based Tutoring for Linear Circuit Analysis Courses

Abstract

Recent progress is described on the development and assessment of a step-based computer-aided tutoring system to teach linear circuit analysis topics. The system automatically generates both AC and DC circuit problems whose topologies and element values are randomly varied, providing an unlimited source of novel problems of any specified difficulty level. Students input each stage of their work in a variety of forms including redrawn circuit diagrams, equations, waveform sketches, matrix equations, numerical answers, and multiple choice answers, and receive immediate feedback on the correctness of their responses. Complete, error-free solutions to the problems are also generated automatically, allowing students to view as many different fully-worked examples as needed. Here, we describe recent developments in the system, including implementation of 13 different tutorials covering topics such as identification of elements in series and parallel, including cases where different "types" of terminals are present; combination of resistors, inductors, and capacitors in series and parallel; and writing and solving both node and mesh equations for both DC and steady-state AC circuits for specified unknown circuit variables. Additional tutorials now in development include sketching the current or voltage as a function of time for an inductor or capacitor, given the opposite such quantity; the mathematical aspects of Laplace transforms; and the sketching of Bode plots from system transfer functions (and vice versa). Additional solution methods are also being developed to support all aspects of DC and AC circuit analysis. We further describe improvements to our tutorial system that now explain wrong answers given when attempting to identify elements in series and parallel, and explain the correct solutions shown in the series/parallel identification, node analysis, and mesh analysis tutorials. Initial results from a laboratory-based study showed a statistically significant 1.21 standard deviation improvement in student performance compared to normal textbook-based homework. The software has been used by over 1290 students at four different universities and some community colleges, with high levels of user satisfaction and generally favorable comments.

1. Introduction

One of the most widely taught courses in undergraduate engineering curricula is linear circuit analysis, as many majors other than just electrical engineering require their students to have at least general familiarity with electrical circuits. For example, around 19 mostly large (70-80 student) sections of this course (including 2 sections completely online) are typically taught every year at Arizona State University (ASU) to ~1200 students by many different instructors. Optimizing student success in this class is therefore of considerable importance. To this end, we have been developing a computer-aided tutoring system with the goal of replacing conventional homework assignments with a highly interactive system that provides feedback to students at every step of their work.[1-3] Such a system can adapt to the individual needs of each student, and can provide as much or as little practice as needed to achieve mastery learning of the course topics. The system generates problems from scratch based on user specifications, from very simple ones to highly complex ones if needed. At the same time, the computer generates fully worked solutions using the same algebraically-based techniques taught to students in these classes, rather than the numerical modified nodal analysis method used by programs such as PSPICE. Students therefore have the opportunity to view as many worked examples of gradually increasing difficulty on any topic as needed, and are not limited to those that are pre-printed in a textbook.

The other key feature of the system is that it accepts a wide variety of inputs from students, rather than just a final numerical answer, as in more typical answer-based tutoring systems. The latter method is essentially equivalent to providing answers in the back of a textbook. While this approach does tell students whether they worked the problem correctly or not, it does not begin to explain where they went wrong if they did not. In the system described here, students write equations directly on the computer screen, receiving immediate feedback as to whether each equation is correct. They do not, therefore, have to waste hours of time solving equations that are not even correctly written. Students can also re-draw circuit diagrams using an interactive graphical editor, as they need to do when combining elements in series or parallel, carrying out source transformations, using superposition, forming Thévenin and Norton equivalent circuits, and so forth. They will be given immediate feedback on the correctness of any re-drawing.

2. Problem Generation System and its Extension to AC Circuits

Previous investigations have established that computer-based instruction can be of great help in teaching circuit analysis, using for example interactive learning modules,[4-6] intelligent tutoring systems,[7-12] multimedia-based instruction,[13] web-based materials,[14-16] virtual classrooms,[17,18] expert systems,[19] and visualization tools.[20] The most widely deployed systems are however those that accompany textbooks on publisher web sites, which provide mainly algorithmic problems where a few element values are randomly varied and the student is asked to input numerical answers. A meta-analysis of different types of tutors by VanLehn found that the typical effect size for such answer-based tutoring systems is a Cohen d-value of 0.31 standard deviations (σ). However, step-based tutors, such as the one we are developing, average d-values of 0.76, nearly as good as the average effect of expert human tutors ($d = 0.79$).[21] A controlled, randomized laboratory-based study, which compared use of our tutorial exercises to working conventional textbook problems for the same period of time, found a large and statistically significant effect size of $d = 1.21\ \sigma$.[2-3] Unique aspects of our system include the use of a variety of pedagogical features, such as color-coding of nodes and other circuit features, and the ability to create an unlimited supply of completely unique problems (both in terms of circuit topology and element values). This feature is not found in most prior systems, which rely on a limited supply of human-generated problems and solutions that might contain errors.

We previously described the basic three-step algorithm we use to generate circuit problems that are similar to typical problems found in textbooks.[1] We first generate a circuit “topology” consisting of only short and open circuits laid out on a square grid, having the desired number of meshes and not being “hinged” (where two parts can be connected by a single wire, and therefore effectively isolated). Next, the required number of generic circuit elements are substituted for some of the shorted segments, with at least two per mesh to ensure that no elements are shorted. Finally, the generic elements are changed into the actual desired ones by placing voltage sources and inductors on the twigs of a randomly selected tree, and current sources and capacitors on the links of that tree (resistors can be placed anywhere). This procedure is a sufficient but not necessary way of ensuring that there are no loops of only voltage sources and inductors or stars (elements connected to a single node) consisting only of current sources and capacitors, which might violate Kirchoff's laws for DC circuits.

For DC circuits, inductors are of course equivalent to short circuits and capacitors to open circuits, necessitating their special treatment as noted above. We recently extended the system to generate AC circuits as well for phasor analysis. In this case, inductors, capacitors, and resistors all have finite impedances, and can be placed anywhere. We generate randomly selected impedances (from which component values can be calculated at a given frequency if the problem is to be specified in terms of them) for all passive elements, and introduce complex values for sources (usually displayed in polar format, with randomly selected magnitudes and phases). The impedances are normally shown in rectangular format, in keeping with the practice most commonly used in textbooks.

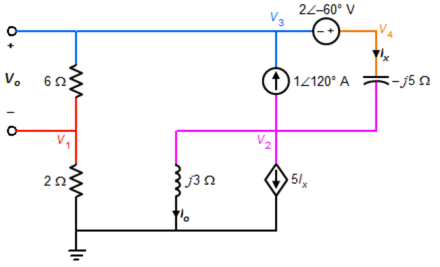
We further extended the equation writing module to cover the case with complex impedances and source values. The matrix solution and all calculations are now performed using complex numbers for all circuits (even DC ones, where the imaginary parts are identically zero and phases are all 0° or 180°). The speed penalty for doing so is negligible. An example of a randomly generated AC circuit (created directly in the phasor domain) is shown in Fig. 1, along with the equations and solution via node analysis (all generated by the computer). Mesh analysis solutions can also be performed (not shown). Note that nodes are color coded in the figure to help students understand the circuit. It is also possible to select a particular KCL equation to color code, in which case each term in the equation is color-coded to match the color of an arrow drawn on the circuit diagram to represent a current leaving a given node or supernode (as shown in Ref. 2) (nodes are then not color coded to avoid confusion). Similar color coding can be performed for the voltage drops around a selected mesh or supermesh when doing mesh analysis. These pedagogical features help students understand the origin of the terms in the equations.

Problem #1

Circuit Diagram with Node Analysis

Compute the following 2 quantities for this circuit:

V_o; I_o



Voltage constraint equations:

$V_4 - V_3 = 2\angle{-60}°\text{ V}$

KCL equations for each node or supernode:

$$\frac{V_1}{2\ \Omega} + \frac{V_1 - V_3}{6\ \Omega} = 0$$

$$\frac{V_2}{j3\ \Omega} + 5I_x + 1\angle 120°\text{ A} + \frac{V_2 - V_4}{-j5\ \Omega} = 0$$

$$\frac{V_3 - V_1}{6\ \Omega} - 1\angle 120°\text{ A} + \frac{V_4 - V_2}{-j5\ \Omega} = 0$$

Equations for control variables of dependent sources:

$$I_x = \frac{V_4 - V_2}{-j5\ \Omega}$$

Sought variable equations:

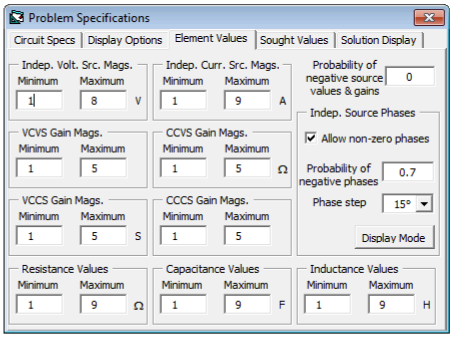
$$V_o = V_3 - V_1$$

$$I_o = \frac{V_2}{j3\ \Omega}$$

Solution:

$V_o = 6.42\angle 101°$ V; $I_o = 2.77\angle 176°$ A

$V_1 = 2.14\angle 101°$ V; $V_2 = 5.17\angle 85.9°$ V; $V_3 = 8.56\angle 101°$ V; $V_4 = 6.71\angle 94.8°$ V; $I_x = 0.357\angle{-149}°$ A


Problem Specifications

Circuit Specs | Display Options | Element Values | Sought Values | Solution Display

Indep. Volt. Src. Mags.: Minimum 1, Maximum 8 V

Indep. Curr. Src. Mags.: Minimum 1, Maximum 9 A

Probability of negative source values & gains: 0

VCVS Gain Mags.: Minimum 1, Maximum 5

CCVS Gain Mags.: Minimum 1, Maximum 5 Ω

Indep. Source Phases

☑ Allow non-zero phases

Probability of negative phases: 0.7

Phase step: 15°

VCCS Gain Mags.: Minimum 1, Maximum 5 S

CCCS Gain Mags.: Minimum 1, Maximum 5

Display Mode

Resistance Values: Minimum 1, Maximum 9 Ω

Capacitance Values: Minimum 1, Maximum 9 F

Inductance Values: Minimum 1, Maximum 9 H

Fig. 1. Automatically generated steady-state AC circuit problem (generated directly in the phasor domain), showing the computer-generated equations and solution as well as the instructor interface used to specify the problem characteristics.

We also added the ability to create AC problems in the time domain, where reactive element values are shown in farads and henries, and independent voltage and current sources are shown as cosine functions with appropriate phase angles. In this case, we first generate impedances in a desired range, so that the different elements in the circuit will have comparable impedances (otherwise, student errors in combining them might go undetected when they enter answers to a limited number of significant digits). We then select a linear time frequency whose mantissa and exponent are both randomly chosen, and convert the impedance values of the reactive elements into element values for that frequency. These values are then rounded to the nearest integers to create a reasonable looking problem. The first step in the automated or student-generated solution is then to convert those element values back to impedances at the specified frequency (which will now not be integers, because we use round numbers for f rather than for ω).

3. Student Input Modules

A. Graphical Circuit Editor

Problems such as superposition, source transformations, combination of series and parallel passive elements and sources, and formation of Thévenin and Norton equivalents all require students to modify given circuit diagrams. To do this in our step-based philosophy, where all student work is immediately checked for correctness, we created a graphical circuit editor module as shown in Fig. 2. This editor (currently implemented in forms using Visual Basic 6.0) can be used to draw new circuits from scratch, or to modify an existing (randomly generated) circuit in the course of solving a problem. All component values as well as control and sought variable subscripts are directly editable in the colored text boxes visible in the figure. ("Sought" variables are the quantities for which students are asked to solve.) Inappropriate entry values (such as negative resistances) are immediately flagged by turning them red. In selection mode, clicking on any element allows one to then transform it into any other type of element (including a short circuit), delete it (i.e., replace it by an open circuit), reverse its polarity if it has one, add a

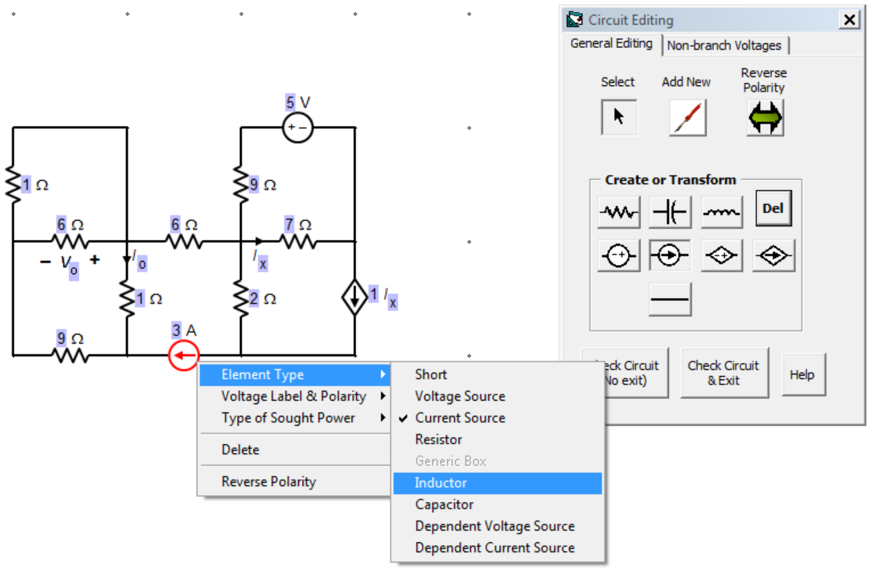


Fig. 2. Current version of the circuit editing interface. Drop-down menus are being used to change a selected element (highlighted in red) into a different type. The blue text boxes contain editable values.

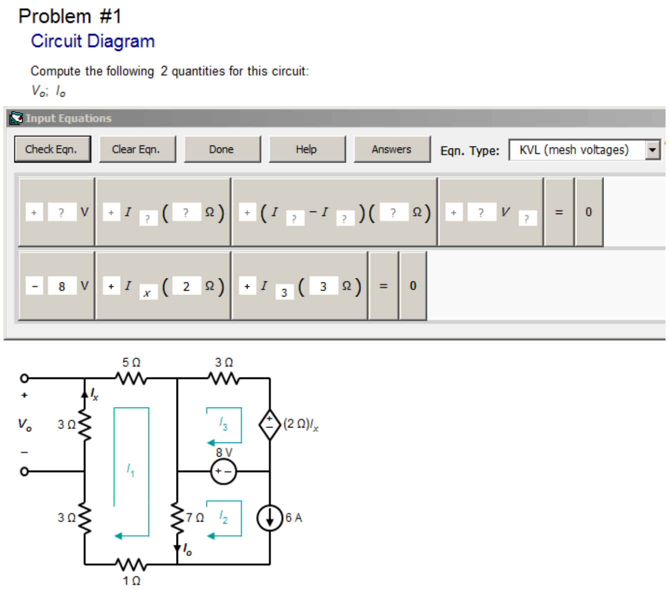
control and/or sought voltage across it or a control and/or sought current through it (or remove same), or make its absorbed or supplied power (as appropriate, depending on element type) a sought quantity. These changes are readily made using a context-sensitive drop-down menu, as visible in the figure. One can also add a ground symbol to any node, which automatically replaces any existing ground (we only allow one such symbol, as we do not currently use it to symbolize interconnected ground points).

In the “Add New” mode, one can add new circuit elements of any selected type to any open location on the grid (elements automatically snap into place on the grid as the cursor is moved). One can also designate a “non-branch voltage” (voltage drop that does not appear directly across any one circuit element) as a sought quantity (item to solve for) by placing markers for the positive and negative sides of the voltage and then clicking a button. (These markers are available on a second tab, which is not visible in Fig. 2.) Any existing non-branch voltage can also be selected and removed using a drop-down menu. This type of feature would not normally be enabled for students, but mainly for instructors who wish to construct a specific (non-randomly-generated) problem for students to work in the system. Such problems can be stored on disc and later loaded into the system during a tutorial. The second tab also allows the problem type to be changed (from seeking one or more specified sought voltages, currents, and/or powers) to finding all node voltages or all mesh currents. The equation editor has the ability to edit both DC and steady-state AC circuits (the latter in either time or phasor domain).

Once a circuit has been edited, the user clicks on a button to check the circuit for validity and (optionally) exit the editor. The normal validity check ensures that that the circuit does not have dangling or shorted elements, that it is not “hinged” (i.e., it does not consist of separate parts that could be connected by only one wire, and are therefore usually isolated from each other), that all required blanks have been filled in with valid values, that all dependent sources have unique control variables that are properly defined somewhere in the circuit, that there are no duplicate sought variable names, and that the circuit is in fact soluble by both node and mesh analysis (i.e., not inconsistent). When this editor has been incorporated into tutorial exercises, as we plan to do in the near future, the validity check will be extended to ensure that the student has made a circuit transformation that is in fact correct. For example, we would check that the student has correctly combined a series or parallel set of elements if they are being asked to simplify the circuit in that manner. They will be required to perform only one such simplification at a time, to make this check easier to carry out. They system could also check that they have performed a valid source transformation, that they have properly turned “off” all but one sources in a superposition problem, and so forth. These checks will be easy to implement using our existing data structures and methods.

B. Equation Entry System

A unique feature of our system is that students enter equations they have been asked to write (such as node or mesh equations) using a template-based interface, as illustrated in Fig. 3. This interface both makes it possible to enter the required terms very quickly, and at the same time gives students guidance on the proper form of their equations as a kind of scaffolding. For a given analysis method (such as node analysis, mesh analysis, voltage division, current division, etc.), the student first selects the type of equation to be entered (if more than one such type exists) from a drop-down menu. A palette of terms is then provided, which normally includes


Problem #1

Circuit Diagram

Compute the following 2 quantities for this circuit:

V_o; I_o

Eqn. Type: KVL (mesh voltages)

$+ \; ? \; V \quad + \; I_{?} \; (\; ? \; \Omega) \quad + \; (I_{?} - I_{?})(\; ? \; \Omega) \quad + \; ? \; V_{?} \quad = \quad 0$

$- \; 8 \; V \quad + \; I_{x} \; (\; 2 \; \Omega) \quad + \; I_{3} \; (\; 3 \; \Omega) \quad = \quad 0$

Fig. 3. Equation entry interface.

only those that would be of the correct form and type needed to write the specified type of equation. Students drag the desired term types into the editing area, then fill in blanks as appropriate (such as the numerical values of resistances, fixed voltage or current values from sources, appropriate control subscripts and gains for dependent sources, etc.) Terms can be re-arranged or deleted from the entry area with the mouse, making changes or corrections easy. Once the student feels the equation is complete and correct, they click a button to check the equation. The computer first validates the syntax of the equation (checking for one and only one equals sign, for example), then checks it against all solution equations of the specified type that have not already been entered. Duplicate entries are flagged. We allow for placing terms on either side of the equation and any sign changes that are consistent (algebraic equivalence). Students generally adapt quickly to using this system, and appear to benefit from the scaffolding it provides.

We also have form-based interfaces to enter "simplified" forms of the systems of equations involved in node or mesh analysis (where all terms involving the same variable have been combined in standard form), and the corresponding matrix equations.[2] These systems again check user entries against the correct ones, allowing for algebraic equivalence (except that equations may not be combined or added to each other at this stage).

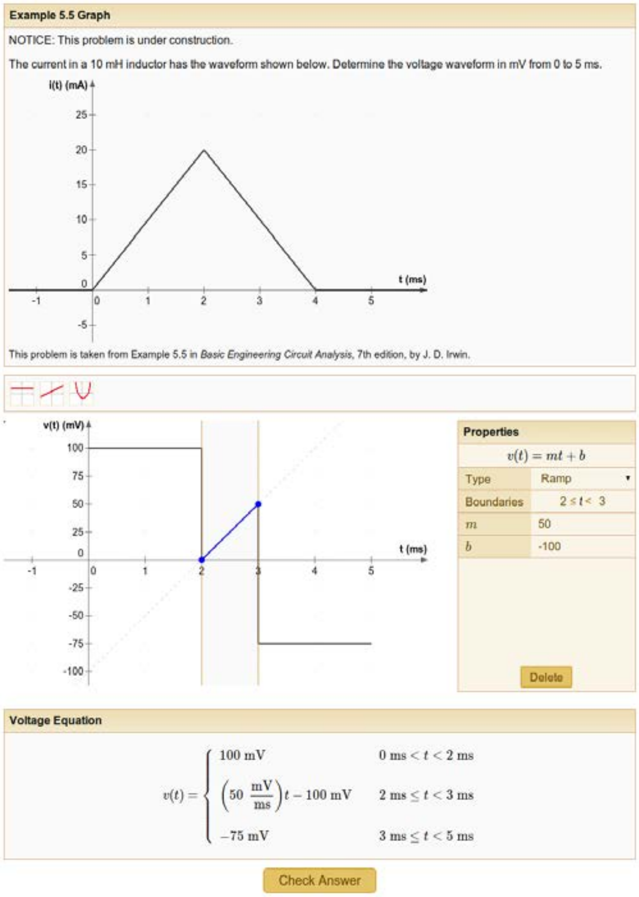
C. Graphical Waveform Interface

A number of problems typically assigned from textbooks give a time-varying signal, often a current or voltage of a circuit element (capacitor or inductor). The student is asked to draw a different time-varying signal, which often involves integration or differentiation. For example, the problem might give the voltage across a given capacitor, and ask the student to sketch the current through that capacitor as a function of time.

We are developing a new system to produce randomly generated problems of the above types, which will present a graph to the user and request that the user sketch one or more different graphs as the answer (these are not multiple choice questions, the graphs must be created from scratch). As our ultimate goal is a purely web-based application, and this type of exercise is somewhat separate from the rest of our tutorial system, we are implementing this function as a web application. The server side is written in PHP, and the client side is written in HTML5 with JavaScript, CSS3, and Canvas. No plug-ins, such as Flash or Java, are used, and the interface will work in all modern browsers.

A screenshot of this interface being used to solve a problem is shown in Fig. 4. In the screenshot, a pre-defined problem is shown to the student with a graph of the current through an inductor, and the voltage across that inductor is requested as the answer. The user has already drawn a waveform (although incorrect), and the equation corresponding to their sketch (with the proper units) is automatically displayed underneath the graph.

The ramp segment is currently selected in the graph, as indicated by a change in color of the line and background. All properties of the segment may be adjusted either by dragging the round handles on the graph with the mouse cursor (or a finger on a touchscreen), or by changing the numbers in the dialog box to the right of the graph. The two endpoints may be moved to adjust the endpoints of the segment. Other functions (such as constants, parabolas, exponentials, sinusoids, exponentially-damped sinusoids, etc.) display handles appropriate to their type. The function types made available change depending on the problem type and difficulty to guide the student in their selections.



Example 5.5 Graph

NOTICE: This problem is under construction.

The current in a 10 mH inductor has the waveform shown below. Determine the voltage waveform in mV from 0 to 5 ms.

This problem is taken from Example 5.5 in *Basic Engineering Circuit Analysis*, 7th edition, by J. D. Irwin.

Properties

$v(t) = mt + b$

Type	Ramp ▾
Boundaries	$2 \le t < 3$
m	50
b	-100

Voltage Equation

$$v(t) = \begin{cases} 100\ \text{mV} & 0\ \text{ms} < t < 2\ \text{ms} \\ \left(50\ \frac{\text{mV}}{\text{ms}}\right)t - 100\ \text{mV} & 2\ \text{ms} \le t < 3\ \text{ms} \\ -75\ \text{mV} & 3\ \text{ms} \le t < 5\ \text{ms} \end{cases}$$

Fig. 4. Screen shot of the web-based waveform input module. The blue ramp section has been selected, so its parameters are the ones shown in the dialog boxes at right. The problem statement is at the top; in this case, it is a pre-defined problem.

The user clicks the "Check Answer" button to send their work to the server, which compares each segment with the correct answer and determines whether the user input is correct. The attempt is logged, and appropriate feedback is given to the user. The user may also give up and be shown the answer (that option is not visible in the screenshot).

We are currently developing the ability to generate problems for this interface. A computer-generated waveform will be displayed to the student, and the system will request an answer waveform for a quantity like charge, current, voltage, power, or stored energy. The system will automatically calculate the correct answer to check the user's work or to display if the user gives up. This same functionality will be used to offer an unlimited number of examples to the user.

4. Classroom Usage and Analysis of Log File Data

A. Usage Statistics and Student Satisfaction

Prior to Fall 2014, our system included three basic tutorials, covering identification of elements in series and parallel, writing node equations (including voltage constraint equations, supernodes, and dependent sources), and writing mesh equations (including current constraint equations, supermeshes, and dependent sources). Starting in Fall 2014, the system was expanded to include 10 new tutorials (for a total of 13). One involves the identification of series and parallel elements in the presence of terminals (which can be alternatively used to measure voltage, view an input impedance, or attach an arbitrary subcircuit), which prepares students to combine elements to find an input impedance as viewed from a set of terminals. Additional tutorials involve the series and parallel combination of resistors, inductors, capacitors, and general AC impedances to find the net input resistance, inductance, capacitance, or impedance, respectively (including complicated multi-step simplifications). Others include steady-state AC (phasor) versions of the node and mesh equation tutorials; and tutorials in which DC or AC circuits are fully solved for one or more desired voltages, currents, and/or powers, including formulation of the problem as simplified equations in standard form and then as a matrix equation. (A tutorial that asks students to draw the current or voltage for a capacitor or inductor, given the opposite quantity, using a web-based sketching interface, is expected to be released in Spring 2015. A number of other tutorials are currently in development.)

The first three basic tutorials were used as required homework exercises in four sections of EEE 202 at ASU, and two sections at local community colleges in Spring 2013 (as optional exercises in the latter cases), in one section at ASU in Summer 2013, in four sections of EEE 202 and in one section of EE 20224 (Introduction to Electrical Engineering) at the University of Notre Dame in Fall 2013, in six sections of EEE 202 at ASU in Spring 2014, and in one section of EEE 202 at ASU in Summer 2014. The three original tutorials were similar in all of the sections during 2013, but were revised and enhanced significantly in Spring 2014. In Fall 2014, the expanded set of tutorials was used to varying degrees in six sections of EEE 202 at ASU and in one section of EE 20224 at Notre Dame.

Through Fall 2014, the tutorials have been used by over 1290 students in a total of 18 class sections (defined as completing one or more of the tutorials). Through Spring 2014, 80% of the students using the system completed all three of the tutorials available at that time.

To assess student satisfaction with the software, we administer a brief two-question survey at the completion of each tutorial within the software itself. First we ask if the tutorial was “very useful,” “somewhat useful,” “not very useful,” or “a waste of time” in learning the relevant topic. Combining data from Spring 2013 through Fall 2014, the percentages of ratings in each category were about 70%, 25%, 2%, and 2%, respectively, so that 96% of students found them to be very or somewhat useful. The second question in this survey asked students for open-ended comments, a sampling of which is shown in Table I for Spring 2014. In general, students liked the pedagogical devices such as node coloring and found them helpful, and the most frequent request was to extend the system to other topics (which we are now doing). Students generally appreciated that they are not penalized for wrong answers, but always have the chance to simply work a new problem of the same type and difficulty, after being shown the solution to a problem they could not solve.

Subsequently, a 12-question anonymous survey was administered to those who used the software (in Fall 2013 and in Spring and Fall 2014). It addressed three main issues, whether the tutorials were useful and well designed, if the difficulty and coverage was appropriate, and whether students prefer them over conventional homework exercises. The average favorable ratings in these categories were 84%, 82%, and 80%, respectively. Ratings by students at Notre Dame were uniformly higher than those at ASU (92%, 92%, and 88% in the above categories). Ratings by students at Morgan State University (88%, 70%, and 88%) were also generally good, although more of those students apparently tended to feel that the difficulty and length of the exercises were too high. Overall, the survey results suggest that the software can be employed usefully at institutions of different types.

B. Analysis of Log File Data

Student actions are recorded to a central server while they play the "games" (tutorials). In Spring 2013, we recorded only completion of circuit problems and tutorials, but in Fall 2013 we began recording details about the numbers of right and wrong answers at each level of each tutorial, to get a better picture of student progress. Through Fall 2014, we have collected over 402,000 log entries while students attempted to analyze or solve over 35,000 distinct circuit diagrams using our system. The analysis of data from Fall 2013 showed that even though students were completing the

Table I. Sample Verbatim Student Comments on Software from Spring 2014.

Series-Parallel Tutorial

- Wish Unit Games were available. Also wish that there were versions of circuit tutor for superposition, and other concepts. This Program taught me these concepts quickly and well.
- I understand the difference between series and parallel so much better now! Thanks!
- This game was brilliant and really helped me in understanding how to spot parallel and series circuit elements. I think there needs to be more resources like this.
- I really liked this game and the tutorial before the game. I think the tutorial explained things really well and I was able to get through the game quite easily. I learned a lot of series and parallel circuits that I didn't fully understand before doing this exercise. Very helpful! We should do more of these!
- If there were video tutorials, that'd be great instead of reading through textual information.
- Coloring the nodes was EXTREMELY helpful. I honestly would have never thought about doing that. Thanks for the help

Nodal Analysis

- More chances to write out an equation, as sometimes just making a simple sign error was the issue and could have been resolved on the following guess.
- I thought I had a good understanding of Nodal Analysis but this exercise really helped
- Could have used this to study before exam.
- A tutorial at the beginning, like with the first module, would have been helpful.
- I liked how there was an option to view new circuits without penalty.

Mesh Analysis

- Cleared my confusion on Mesh Analysis. Thanks again! I hope you do other exercises on other circuit's topics (Thevenin .. etc)
- I like these games! They help drill the concepts in too :)
- More enjoyable than a standard HW.
- These are awesome!
- This was also very helpful in understanding this topic. Lots of great practice.

tutorials “successfully,” they were still entering many wrong answers, even up to the end of the tutorial. For example, 51% of the sets of elements they identified as being in series or parallel were incorrect. They were nonetheless able to complete the exercise because there was (at that time) no limit on the number of wrong answers (possibly guesses) they could enter about a particular circuit diagram, and no penalty for wrong answers. Further examination showed a different result when averaging the fraction of incorrect answers over users, where the average percent wrong was only 32%, with a median percent wrong of 21%, indicating a skewed distribution. Examination revealed that students having more difficulty (and getting more wrong answers) played the game much longer than those doing well. Moreover, the average percentage correct tended to stagnate or even decrease as students moved from the easier levels into harder levels, which does not reflect the improvement we would like to see as they gain experience (see Fig. 5). If the students were truly mastering the ideas, we would hope to see a reduction in wrong answers as students gain experience and go the higher levels, or at least fewer overall. (The laboratory trial did show very substantial learning gains from doing this exercise,[2,3] but that does not mean that all students mastered the ideas.)

For the node and mesh equation tutorials, the fractions of entered equations that were correct were 51% and 64%, respectively (the node tutorial usually being completed first). Students had the greatest difficulty with control variable equations for dependent sources, even though they are generally very simple to write (only 33% and 35% of these were correct in node and mesh

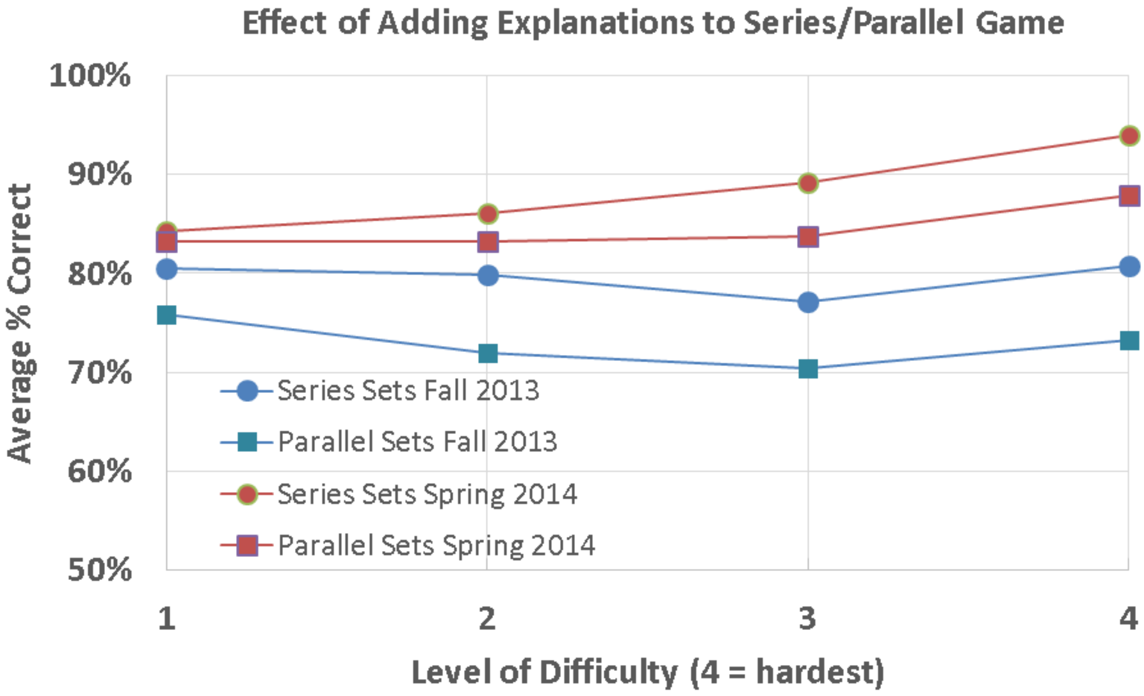


Fig. 5. Average percentage of correct answers submitted in the series-parallel identification game as a function of difficulty level, for two different versions of the software in Fall 2013 and Spring 2014. In the latter case, detailed explanations of right and wrong answers were included, leading to substantial improvements in the fraction of correct answers and in the trend as a function of difficulty level and experience.

analysis, respectively). This observation suggests that students are just not understanding the concept of a control variable, and perhaps confuse it with something else. The KCL and KVL equations were the next most difficult, with correct answer rates of 49% and 66%, respectively. It is not yet clear why students do so much better with KVL than with KCL, and that question needs to be studied further. The voltage and current constraint equations were found to be the easiest, with 65% and 69% correct answers, respectively. Yet further improvements in the tutorials would certainly be desirable to foster more complete and rapid learning.

5. Iterative Refinement of Tutorials

Based on the above observations, we undertook significant revisions of the tutorials in early 2014, in attempt to see if we can achieve "perfect learning" of the material. First, we limited the numbers of wrong answers, so that students are forced to "give up" on a problem if they enter too many for a given type of entry. This change should encourage them to think more deeply about what is correct, rather than just trying many different answers to find the right one. When they do give up (voluntarily or involuntarily), they are now offered much more detailed explanations of the correct solutions (previously, we just showed them the correct answers without any discussion). The explanations are tailored to the specific circuit and refer to its color-coded nodes (for series-parallel identification and node analysis tutorials). In the series-parallel tutorial, they are also given a detailed explanation of why any incorrect (as opposed to incomplete) set of elements they entered is not in fact in series or parallel as they stated. We also improved the introductory tutorial explaining the ideas of series and parallel connections. The effect of these changes is shown as the red curves in Fig. 5. The percentage of correct answers (averaged by student) now increases with increasing difficulty and experience, rising from 83% and 86% for parallel and series sets, respectively, to 88% and 94%. This trend reflects the type of desired learning behavior.

A similar tutorial introduction has not yet been created for the node and mesh analysis cases but is planned, though of course students can also rely on their textbook and lectures for this purpose. We do not yet offer explanations of why equations are incorrect for mesh and node analysis, as that will require revision of our equation entry interface to ask students to specify which mesh or node (or supermesh or supernode) they are writing the KVL or KCL equation for, or for which source they are writing a constraint equation, or for which dependent source they are writing a control variable equation (if there is more than one, which is not normally the case).

We made similar improvements to the node and mesh equation games in the same time frame. These games require students to enter node or mesh equations for randomly generated circuits using a pre-defined palette of terms to provide scaffolding. First, we re-arranged the terms in the equation palette to more logically reflect how the different types of equations are formed, and limited the number of wrong answers prior to giving up. Next, we added detailed explanations of the correct Kirchhoff current law and Kirchhoff voltage law equations, and generic explanations of voltage & current constraint equations and control variable equations for dependent sources. We also required completion of two problems rather than one at the hardest level, for extra practice. With these improvements, the percent correct generally improved by about 10% (with the exception of difficulty level #3 out of 5, where they decreased for an unknown reason), as illustrated in Figs. 6 and 7. The trend of correct percentage with difficulty was not monotonic, but showed improved behavior compared to the earlier versions.

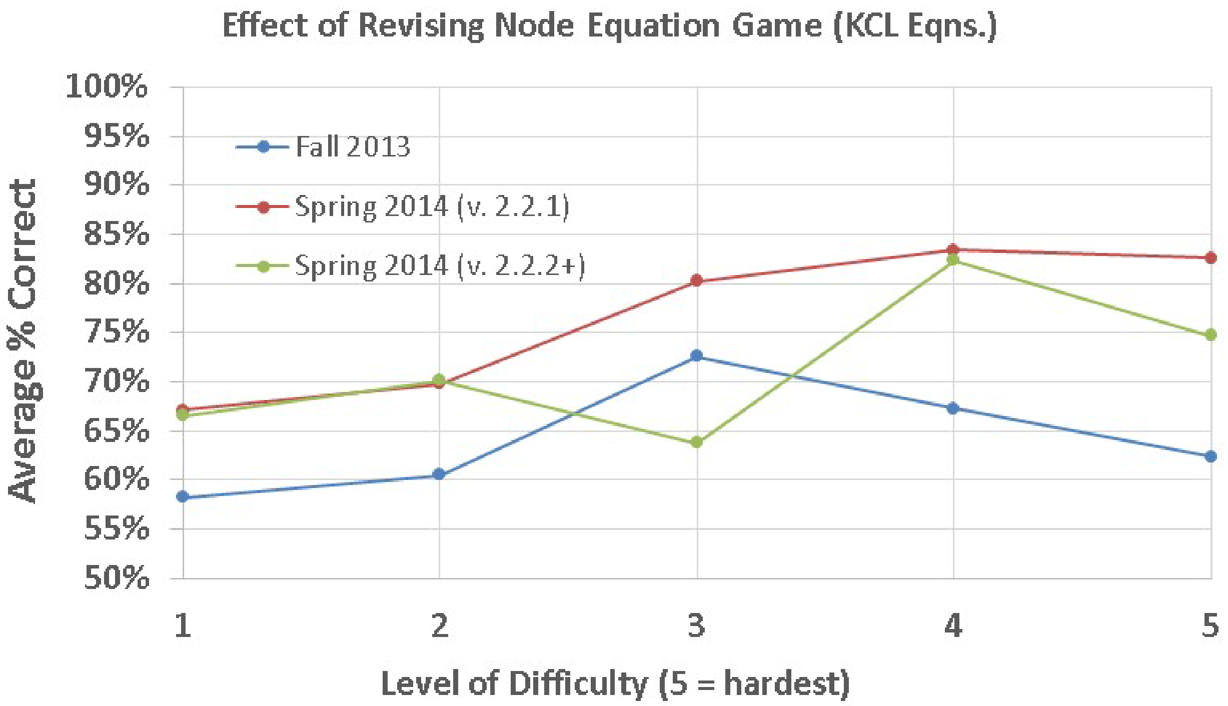


Fig. 6. Average percentage of correct KCL equations entered in the node equation game as a function of the level of difficulty for three software versions, in Fall 2013 and in Spring 2014. Version 2.2.1 revamped the layout of the equation entry interface; and version 2.2.2 added explanations of correct KCL and KVL equations along with generic explanations of control variable and constraint equations, limited the number of wrong answers before giving up, and required completion of two problems rather than one at level 5. The users of version 2.2.1 were students who chose to use it earlier in the semester, and may have been more motivated than those using version 2.2.2.

6. Conclusions

Our computer-based interactive tutorial system has been extended to incorporate new features, including an improved, highly capable graphical circuit editor, the ability to generate and solve AC circuits using phasor analysis in addition to DC circuits, and a fully developed web-based waveform sketching tool that can assess the correctness of student drawings. The number of available tutorial exercises has been considerably expanded from 3 to 13. At the same time, usage of the system has been expanded to many more class sections both at ASU, at Notre Dame, and at Morgan State University, and will shortly be extended to use at other institutions including the University of the Pacific as well. High levels of student satisfaction have been achieved. A large quantity of log file data is being analyzed and used to iteratively refine the existing tutorials, with the eventual objective of achieving “perfect learning” of the covered topics by all students. New tutorials are also under development.

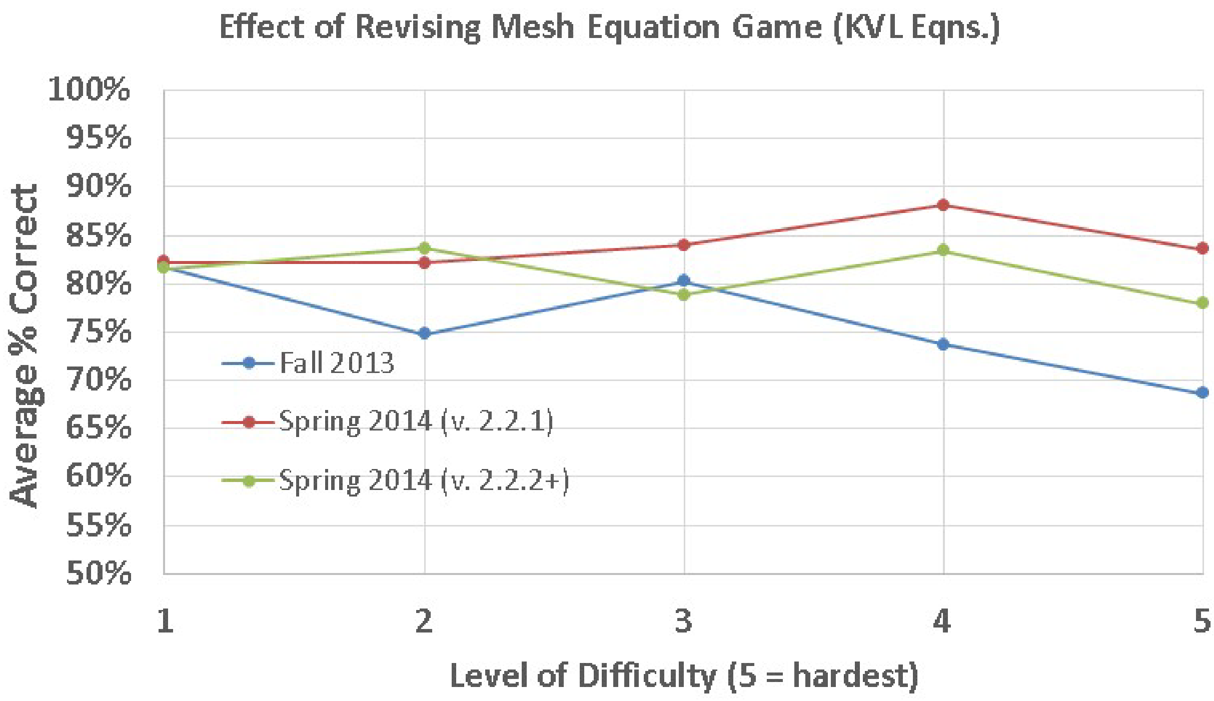


Fig. 7. As Fig. 6, but for correct KVL equations in the mesh equation game.

7. Acknowledgments

This work was supported by the National Science Foundation through the Transforming Undergraduate Education in Science, Technology, Engineering and Mathematics Program under Grant Nos. DUE-1044497 and DUE-1323773. We thank Drs. J. Aberle, M. Ardakani, R. Ferzli, S. Goodnick, R. Gorur, G. Karady, Hongwei Mao, B. Matar, L. Sankar, Donghoon Shin, Meng Tao, C. Tepedelenlioglu, T. Thornton, D. Vasileska, Chao Wang, Hongbin Yu, and Hongyu Yu for using our software in their sections of EEE 202 at ASU; A. Holmes for using it in his course ECE 2630 at the University of Virginia; and Y. Astatke for using it in his course EEGR 202 at Morgan State University. We thank Daniel Sayre of John Wiley & Sons, Inc. for providing the textbook copies used in our laboratory experiment and for his support of the project.

References

[1] C. D. Whitlatch, Q. Wang, and B. J. Skromme, "Automated problem and solution generation software for computer-aided instruction in elementary linear circuit analysis," in *Proceedings of the 2012 American Society for Engineering Education Annual Conference & Exposition* (Amer. Soc. Engrg. Educat., Washington, D.C., 2012), p. Paper 4437.

[2] B. J. Skromme, C. D. Whitlatch, Q. Wang, P. M. Rayes, A. Barrus, J. M. Quick, R. K. Atkinson, and T. Frank, "Teaching linear circuit analysis techniques with computers," in *Proceedings of the 2013 American Society for Engineering Education Annual Conference & Exposition* (Amer. Soc. Engrg. Educat., Washington, D.C., 2013), p. 7940.

[3]B. J. Skromme, P. J. Rayes, C. D. Whitlatch, Q. Wang, A. Barrus, J. M. Quick, R. K. Atkinson, and T. S. Frank, "Computer-aided instruction for introductory linear circuit analysis," in *Proceedings of the 2013 IEEE Frontiers in Education Conference* (Inst. Electrical & Electronics Engrs., Piscataway, NJ, 2013), p. 314.
[4]D. L. Millard, "Interactive learning modules for electrical engineering education," in *Electron. Compon. and Technol. Conf.* (2000), p. 1042.
[5]D. Millard and G. Burnham, "Increasing interactivity in electrical engineering," in *33rd ASEE/IEEE Frontiers in Educat.* (IEEE, Boulder, CO, 2003), p. F3F8.
[6]J. R. Jones and D. A. Conner, "The development of interactive tutorials for introductory circuits," in *Proc. IEEE 1st Internat. Conf. on Multi-media Engineering Education* (1994), p. 108.
[7]M. Nahvi, "A computer based intelligent synthetic tutor-tester for electrical engineering," in *Proc. IEEE Internat. Conf. on Systems, Man, & Cybernetics* (1990), p. 742.
[8]B. P. Butz, M. Duarte, and S. M. Miller, "An intelligent tutoring system for circuit analysis," IEEE Trans. Educ. **49**, 216 (2006).
[9]B. P. Butz and S. M. Miller, Evaluation of IMITS for the National Science Foundation, http://www.temple.edu/imits/shock/Evaluation.pdf.
[10]A. Yoshikawa, M. Shintani, and Y. Ohba, "Intelligent tutoring system for electric circuit exercising," IEEE Trans. Educ. **35**, 222 (1992).
[11]P. D. Cristea, R. I. Tuduce, and A. R. Tuduce, "Knowledge assessment in intelligent e-learning environments," in *Internat. Conf. on Signals & Electronic Systems* (Poznan, Poland, 2004), p. 573.
[12]S. Watanabe, J. Miyamichi, and I. R. Katz, "Teaching circuit analysis: A mixed-initiative intelligent tutoring system and its evaluation," in *Proc. IFIP TC3 Internat. Conf. on Advanced Research on Computers in Education (ARCE'90),* edited by R. Lewis and S. Otsuki (Tokyo, 1991), p. 19.
[13]G. F. Shannon, "Multi-media computer based teaching--A case study," in *Proc. IEEE 1st Internat. Conf. on Multi-Media Engineering Education* (1994), p. 398.
[14]B. Oakley II, "Use of the Internet in an introductory circuit analysis course," in *Proc. 1993 Frontiers in Education Conf.* (1993), p. 602.
[15]E. C. Shaffer and F. J. Mabry, "A student designed, Web-based learning program for circuit analysis," in *Proc. 2000 Frontiers in Education Conf.* (2000), p. T2D.
[16]L. Weyten, P. Rombouts, and J. De Maeyer, "Web-based trainer for electrical circuit analysis," IEEE Trans. Educ. **52**, 185 (2009).
[17]B. Oakley II and R. E. Roper, "Implementation of a virtual classroom for an introductory circuit analysis course," in *Proc. 1994 Frontiers in Education Conf.* (1994), p. 279.
[18]B. Oakley II, "A virtual classroom approach to teaching circuit analysis," IEEE Trans. Educ. **39**, 287 (1996).
[19]H. E. Hanrahan and S. S. Caetano, "A knowledge-based aid for dc circuit analysis," IEEE Trans. Educ. **32**, 448 (1989).
[20]E. R. Doering, "*CircuitViz*: A new method for visualizing the dynamic behavior of electric circuits," IEEE Trans. Educ. **39**, 297 (1996).
[21]K. VanLehn, "The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems," Educat. Psychologist **46**, 197 (2011).