

## Expansion and Evaluation of a Step-Based Tutorial Program for Linear Circuit Analysis

**Dr. Brian J Skromme, Arizona State University**

Dr. Brian Skromme is a professor of Electrical, Computer, and Energy Engineering and 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.

**Mr. Paul Rayes, Arizona State University**

**Dr. Bing Cheng**

**Brian McNamara**

**Aaron S Gibson**

**Dr. Angela Barrus, Arizona State University**

**John M Quick**

**Prof. Robert Kenneth Atkinson, Arizona State University**

Dr. Atkinson is an Associate Professor of Computer Science and Educational Technology with a joint appointment in the School of Computing, Informatics and Decision Systems Engineering in the Ira A. Schools of Engineering and the Division of Educational Leadership and Innovation in the Mary Lou Fulton Teacher's College at Arizona State University in Tempe, Arizona USA. His current research foci include the design of personalized learning environments, educational applications of social media, mobile learning, learner analytics, and data mining of large multimodal data sets. He has obtained—both independently and collaboratively—over \$20 million dollars in grant support from a variety of sources including the NSF, Office of Naval Research and Intel Corporation.

**Dr. Yih-Fang Huang, University of Notre Dame**

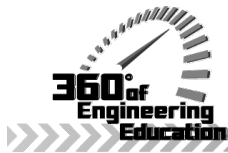
Yih-Fang Huang is Professor of Electrical Engineering and Senior Associate Dean for the College of Engineering at the University of Notre Dame. Dr. Huang received his BSEE degree from National Taiwan University, MSEE degree from University of Notre Dame and Ph.D. from Princeton University. He served as chair of the Electrical Engineering Department at the University of Notre Dame from 1998 to 2006. His research interests focus on theory and applications of statistical signal detection and estimation, and adaptive signal processing.

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.

Dr. Huang 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 College of Engineering's Teacher of the Year Award in 2013. Dr. Huang is a Fellow of the IEEE.

**Daniel H. Robinson, Colorado State University**

Dan Robinson is 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 has served as the editor of Educational Psychology Review since



2006, and as an editorial board member of nine journals. 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.

# Expansion and Evaluation of a Step-Based Tutorial Program for Linear Circuit Analysis

## Abstract

The current status of a computer-based tutorial system that uses a step-based tutoring approach to teach elementary linear circuit analysis is described. The system features automatic problem generation (varying both element values and circuit topology) and solution of those circuits by a variety of methods, including node and mesh analysis and current and voltage division. A wide variety of student inputs, such as algebraic and matrix equations, numerical answers, and re-drawn circuits are accepted. A web-based interface is being developed with the capability to pose questions of various types in a sequence that can be specified by a tutorial writer in an authoring interface. Various pedagogical features such as color coding are employed to aid in student learning, and typical student misconceptions are being addressed. Recent work includes the addition of a web-based instructor interface where instructors can create class sections and continually monitor student progress, including access to detailed logs of student activity. Graphics are now displayed directly within the program to eliminate the need for PowerPoint and facilitate wider usage. A web-based waveform sketching tool is being developed that has the potential for wide use in other courses such as calculus as well. More complete problem solutions are now available including explicit equations for the desired voltages, currents, and powers. The software has been used on a mandatory or strongly encouraged basis in 10 sections of a linear course at Arizona State University (totaling over 560 students) and by 42 students at the University of Notre Dame, and a few students at the University of Virginia and two community colleges in 2013. Student satisfaction has been very high at all sites. A controlled, randomized laboratory-based study showed that learning gains are approximately 10X higher using the software tutorials than when working conventional textbook problems for the same period of time, with a statistically significant effect size (Cohen  $d$ -value) of 1.21 standard deviations.

## 1. Introduction

Introductory linear circuit analysis is a very widely taught and important foundational class for beginning electrical engineers, and is also taught to a large number of other engineering majors as their primary introduction to electrical engineering. For example, at Arizona State University (ASU) we teach such a course (EEE 202) to over 600 students a year, typically in 11 sections. Many students struggle with this class, often due to a lack of detailed, rapid feedback and correction on their work, insufficient use of active learning strategies, and in our opinion, an insufficient systematization of and emphasis on the numerous principles that are necessary to solve a variety of problems successfully. One potentially very useful approach to address these problems is to develop step-based computer-aided tutoring systems (in the form of “games”) that systematically demonstrate and exercise the necessary skills in a way that offers student unlimited repetition until they individually achieve mastery of each topic. Such an approach allows the software the flexibility to meet each individual student’s variable needs, as opposed to the “one size fits all” philosophy inherent in conventional lecturing and textbooks. This approach is also very well suited to potential applications in massive open on-line courses

(MOOCs) and to other on-line instruction such as that in ASU's novel completely on-line bachelor's degree in electrical engineering (<http://asuonline.asu.edu/online-degree-programs/undergraduate>).

The step-based approach means that the system accepts input at every stage of a student's work, providing immediate feedback so that the student does not waste time solving an incorrect set of equations, or writing equations for an incorrectly drawn circuit diagram, etc. Such an approach has been found by VanLehn to be much more effective than answer-based tutors, which only provide feedback on the correctness of a final (usually numerical) answer without giving insight as to where a student went wrong.<sup>1</sup> In particular, step-based tutors showed an average effect size (Cohen *d*-value) of 0.76 standard deviations, compared to 0.79 for human tutors and 0.31 for answer-based tutors. As shown in the following, we achieve an effect size of 1.21, even larger than most step-based tutors. To implement the step-based approach, our system can accept re-drawn (edited) circuit diagrams using a newly re-vamped and much improved editing system, algebraic equations using a novel template-based entry system that provides guidance on the correct form of the equations, waveform sketches as a function of time using a newly-developed web-based graphical input sketching module, systems of simplified algebraic equations and corresponding matrix equations, numerical answers (in a tabular format), and other question types such as multiple choice, true-false, etc.

To provide an unlimited source of error-free worked examples and exercises of variable difficulty for students, our system employs automatic problem and solution generation as described previously, where both element values and circuit topology are randomly generated from scratch for each student.<sup>2-4</sup> With this approach, a student unable to find the correct answer at any step can simply be given the complete answer, and is then given a new problem of the same type and level of difficulty to solve. An example of such a randomly generated problem and its solution using nodal analysis is shown in Fig. 1. This figure illustrates automatically generated node equations, simplification of those equations by collecting like terms, placement of the simplified equations into a matrix equation, explicit equations for the variable the student is asked to calculate (the "sought quantities,") and finally the numerically calculated values of the node voltages and sought quantities.

## **2. New Software Features**

### ***2.1. Instructor Interface***

In order for the program to be widely used, we need to make it convenient for instructors to set up and administer their course sections. We therefore created a web site (at [www.circuittutor.com/web](http://www.circuittutor.com/web)) where instructors can log in, create a new class section, add students to their section using a pre-existing class roster (in Microsoft Excel format), automatically generate and distribute registration codes to enable those students to register on the same site, monitor student progress, and download completion data when students have finished the tutorials. Instructors can also download a trial copy of the software themselves from the site to decide if they want to use it. Administrators can perform additional functions, such as entering new schools in the system, adding instructors for each school (who are then manually sent their own registration codes to gain access), and the other functions that instructors can do. For example, an administrator can pre-create a course section if desired for a given instructor (though

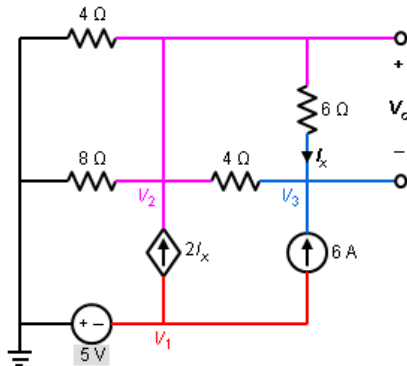
## Problem #1

### Circuit Diagram with Node Analysis

Compute the following 3 quantities for this circuit:

$$V_o; I_x; P_{abs}(5V)$$

Here,  $P_{diss}(1\ \Omega)$  denotes the power dissipated in the gray-labeled  $1\ \Omega$  resistor,  $P_{abs}(1\ V)$  is the power absorbed by the gray  $1\ V$  source, and  $P_{suppl}[(2\ S)V_x]$  is the power supplied by the gray  $(2\ S)V_x$  source.



Voltage constraint equations:

$$V_1 = -5\ V$$

KCL equations for each node or supernode:

$$-2I_x + \frac{V_2}{8\ \Omega} + \frac{V_2 - V_3}{4\ \Omega} + \frac{V_2 - V_3}{6\ \Omega} + \frac{V_2}{4\ \Omega} = 0$$

$$-6\ A + \frac{V_3 - V_2}{4\ \Omega} + \frac{V_3 - V_2}{6\ \Omega} = 0$$

Equations for control variables of dependent sources:

$$I_x = \frac{V_2 - V_3}{6\ \Omega}$$

Simplified node equations:

$$\begin{array}{cccc} V_1 & +0\ V_2 & +0\ V_3 & +0\ I_x = -5 \\ 0\ V_1 & +0.792\ V_2 & -0.417\ V_3 & -2\ I_x = 0 \\ 0\ V_1 & -0.417\ V_2 & +0.417\ V_3 & +0\ I_x = 6 \\ 0\ V_1 & -0.167\ V_2 & +0.167\ V_3 & +I_x = 0 \end{array}$$

Matrix form of node equations:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0.792 & -0.417 & -2 \\ 0 & -0.417 & 0.417 & 0 \\ 0 & -0.167 & 0.167 & 1 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ I_x \end{bmatrix} = \begin{bmatrix} -5 \\ 0 \\ 6 \\ 0 \end{bmatrix}$$

Sought variable equations:

$$V_o = V_2 - V_3$$

$$I_x = (V_2 - V_3)/(6\ \Omega)$$

$$P_{abs}(5\ V) = [5\ V] [ +2I_x + 6\ A]$$

Solution:

$$V_1 = -5.00\ V; V_2 = 3.20\ V; V_3 = 17.6\ V; I_x = -2.40\ A$$

$$V_o = -14.4\ V; I_x = -2.40\ A; P_{abs}(5\ V) = 6.00\ W$$

Fig. 1. Sample randomly generated circuit problem and a fully worked solution using nodal analysis.

this is very easy). An example of an instructor view of their class section is shown in Fig. 2. Note that students are identified only by codes, to ensure anonymity to everyone except their actual instructor. I.e., administrators and researchers do not have access to any information that would identify individual students, and that information is not stored on the server to comply with the requirements of the Family Educational Rights and Privacy Act (FERPA).

To register students in their section, instructors run a program on their own computers that is downloaded from our web site. This program automatically inserts identifying codes into the instructor's roster spreadsheet, transmits the codes and corresponding demographic information from the roster to the server, and can automatically e-mail the codes to students with instructions on how they can register on our web site. A downloaded program is used to ensure that protected information never leaves the instructor's personal computer. The instructors retain the codes to identify students for grading and monitoring individual progress, but no one else can do so. The entire process is automated and takes only a few minutes. Students register on our site

## Students

Export as CSV New Student(s)

ID	Registered	Usercode	Course	Unit Games score	Ser/Par	Node	Mesh	Actions
71	No	STTN5M-NP3E-22RL	EEE 202: Circuits I	72600 / 100k	3 / 4	5 / 5	5 / 5	View Log
88	No	STTN5M-NP3E-267R	EEE 202: Circuits I	37450 / 100k	4 / 4	5 / 5	0 / 5	View Log
82	No	STTN5M-NP3E-2AN8	EEE 202: Circuits I	0 / 100k	0 / 4	0 / 5	0 / 5	View Log
65	No	STTN5M-NP3E-2DCJ	EEE 202: Circuits I	89800 / 100k	4 / 4	5 / 5	5 / 5	View Log
87	No	STTN5M-NP3E-2N2J	EEE 202: Circuits I	129300 / 100k	4 / 4	5 / 5	5 / 5	View Log
79	No	STTN5M-NP3E-42K9	EEE 202: Circuits I	100400 / 100k	4 / 4	5 / 5	5 / 5	View Log
29	No	STTN5M-NP3E-4LVS	EEE 202: Circuits I	183550 / 100k	4 / 4	5 / 5	5 / 5	View Log
52	No	STTN5M-NP3E-4MFN	EEE 202: Circuits I	51500 / 100k	4 / 4	5 / 5	5 / 5	View Log
50	No	STTN5M-NP3E-5E4K	EEE 202: Circuits I	20600 / 100k	4 / 4	5 / 5	5 / 5	View Log
90	No	STTN5M-NP3E-5TU5	EEE 202: Circuits I	650 / 100k	4 / 4	5 / 5	5 / 5	View Log
44	No	STTN5M-NP3E-68G8	EEE 202: Circuits I	52000 / 100k	4 / 4	5 / 5	5 / 5	View Log

Fig. 2. Instructor view of the CircuitTutor web site, showing students registered in his section and their progress. Completed tutorials are shown in green, partially completed in yellow, and unstarted ones in red. Note the absence of identifying information (students are identified only by codes on the site). (The “Registered” field incorrectly shows Red/“No” for all students, this is an artifact of the semester having been completed.)

using an e-mail address of their choosing (for password resets) and select their own password. They can then download the software directly from the web site, which is installed with a few clicks on their machine. The software automatically updates itself from the server each time a new version is released with new features or bug fixes. All student progress is transmitted to the server, and they are warned any time the server connection is broken. Also, an error trapping system automatically reports any fatal errors that students encounter to our server for later analysis by the developers, and asks students to e-mail us a screen shot to aid in de-bugging.

In Fig. 2, note that instructors can click a “View Log” button for each student and see the student’s detailed log file if they wish, which currently notes each time a student enters a correct or incorrect answer in an exercise; views an example or a section of a tutorial; starts, gives up on, or completes a game; or finishes a level or an entire tutorial. Student time spent on each task is also recorded. The entire log file database can be downloaded and studied by the researchers and developers working on the system. This information has already proven very useful in assessing the effectiveness of the software, as discussed below. In the future, we hope to develop a more detailed dashboard for instructors and administrators, to give them an even clearer picture of which types of problems or issues are causing the most difficulty for students. Such a system would potentially enable real-time adjustment of instructional strategies in the classroom.

## 2.2. Platform Improvements

The original software was developed entirely in Microsoft PowerPoint using Visual Basic for Applications (VBA).<sup>2</sup> We later transitioned to a hybrid approach using a standalone program written in Visual Basic 6.0 (VB6, chosen for its close similarity to VBA), which communicated with a separate PowerPoint VBA program on the same computer to draw the graphics. This approach also allowed us to log progress to a central server, rather than storing progress locally as in the original system. While this approach facilitated very rapid development, the PowerPoint platform is not well suited to interactive circuit editing or other input functions, since graphic objects on a PowerPoint slide cannot generate events and cannot be locked against unintentional changes. We have now completed a complete transition to VB6, where editable graphics are displayed on a form but appear virtually identical to those formerly displayed in PowerPoint. The new graphical circuit editor is much faster, more robust, and more user friendly than the previously described version,<sup>3</sup> though it uses much of the same code. A screen shot of the VB6-based circuit editor is shown in Fig. 3.

The program no longer requires that students have a copy of PowerPoint to use it, but is currently still limited to Windows PC's, or other platforms running Windows emulators. Students can easily access such machines on campus computing sites if they do not own one. A future web-based version is planned, but will require considerable additional development to implement.

## 2.3. Web-Based Waveform Sketching Input Module

A number of typical problem types in textbooks involve showing the student a waveform for the current or voltage of a capacitor or inductor as a function of time, and asking them to sketch the other quantity (by integration or differentiation). Similar problems are used for the relationship between charge passing a point and current, or for the relationship between absorbed (or supplied) power and stored or dissipated energy. We are therefore constructing a graphical interface to accept student "sketches" of such piecewise functions (which can alternatively be

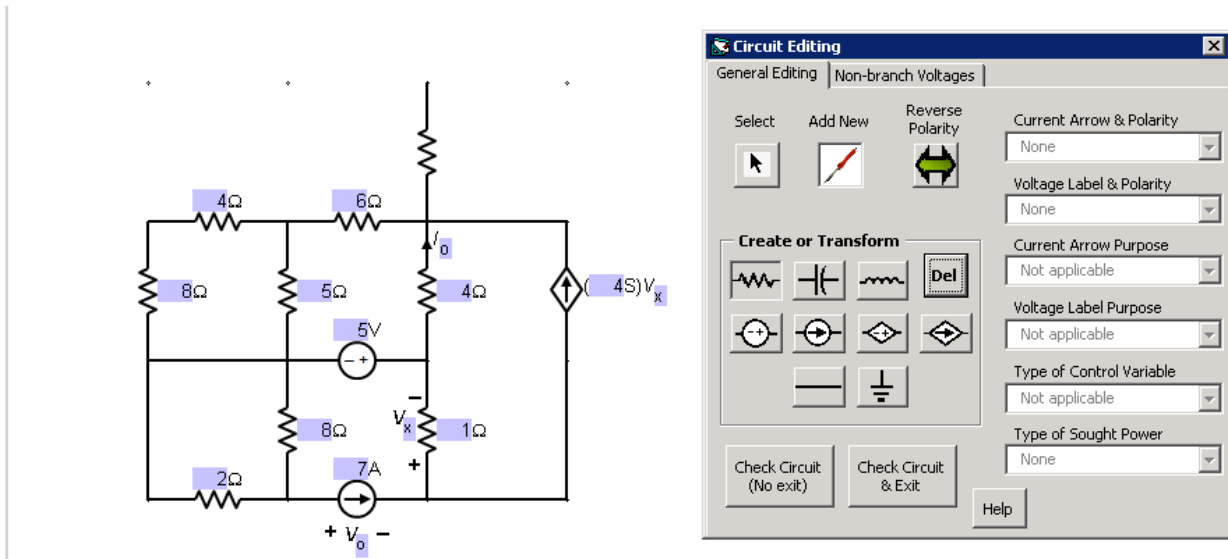


Fig. 3. Revamped form-based circuit editing interface. A resistor is about to be placed on the diagram by clicking. The blue boxes show editable values.

specified by typing in their parameters). As our ultimate goal is a purely web-based application, we have constructed this input module using a web-based platform. The server side application is written in PHP with a MySQL database. The client side uses HTML5, JavaScript, CSS3, and Canvas, which are supported in all modern browsers. No extra plug-ins, such as Flash or Java, are used, meaning the system can be used on any relatively new device, including personal computers, smart phones, and tablets, regardless of which operating system they are running. So far, it has been tested on Internet Explorer 10/11, Google Chrome v. 31, Firefox v. 26, and the Android 4 browsers.

A screen shot of this interface being used to construct a waveform is shown in Fig. 4. The user has entered several segments of the piece-wise defined graph using the graphical interface, using constant and ramp functions in this case. The corresponding equation for the graph is automatically generated and displayed at the bottom. Other functions such as parabolas, exponentials, and sinusoids are also available (note the icons for those). Each “segment” of the piecewise graph has moveable boundaries, which become visible brown lines when the corresponding segment is selected (the constant segment extending from 3 to 5 s is selected in the figure). The parameters of the function are adjusted by grabbing handles that move the end points of a line segment; the value of a constant function; the baseline, amplitude, period/frequency, and phase of a sinusoid; and the baseline, initial value, and time constant of an exponential, for example.

This module will be able to generate problems of user-specified types randomly. The tutorial author will specify how many segments to include, the types of functions to use, whether the graph should be continuous, and so forth. The computer-generated waveform will be displayed to the student and tell them what they are to calculate. The system will then find the “answer” waveform for a quantity like current, voltage, power, or stored energy automatically, and check

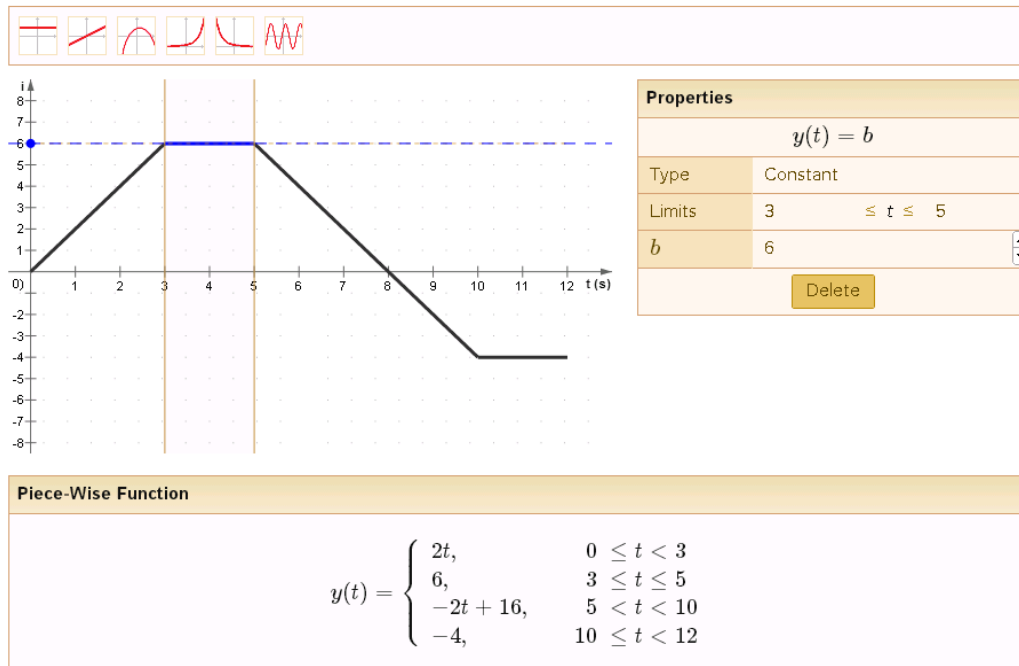


Fig. 4. Screen shot of the web-based waveform input module. The blue constant section has been selected, so its parameters are the ones shown in the dialog boxes at right.



the user-inputted waveform against that answer. As usual in our system, the student will be able to “give up” and see the answer (or possibly obtain hints), and then be given another problem of the same type to work themselves.

This system could also be generalized to be used in many other engineering, mathematics, and science courses. For example, calculus students could learn to sketch integrals and derivatives this way to develop graphical understanding, and the system could be applied to problems relating quantities such as position, velocity, and acceleration in mechanics; or charge density, electric fields, and electrostatic potentials in electrostatics and semiconductor device problems.

We also plan to adapt this system to enter and plot asymptotic forms of Bode plots, where it could automatically convert from such plots to frequency response or system response functions, or vice versa.

#### ***2.4. Web-based Tutorial Authoring and Delivery***

We have begun developing a system to allow authoring of tutorial sequences that employ our software through a web-based platform into JavaScript Object Notation (JSON)-formatted data, rather than having to hard-code all our tutorials as we do now. Thus far, we have the framework to create sequences of questions that can be multiple-choice, true-false, numerical answer, short answer, or essay types, which are then presented to students on a similar web-based platform. We will initially use this system to administer the DIRECT concept inventory<sup>5</sup> as a pre- and post-test to students in our circuits class, which we currently administer using BlackBoard. In future work, we will interface this system to our circuit generation, solution generation, and graphics generation software in VB6 (running on a web server) so that it can fully control the structure of tutorial sequences.

### **3. Classroom Usage and Survey Results**

A major recent focus of our project has been expanding usage of our software to a large number of students in a variety of institutions of different types. The use of the software is generally made mandatory (e.g., as a required homework assignment whose completion forms part of their grade), as busy students do not tend to use it very often if it is purely optional for them. The web-based interface described above has greatly facilitated broader trials and adoption. As we increase usage of the software, we simultaneously collect a rich set of data including the log files described above, a brief two-question survey at the completion of each tutorial, and an expanded survey after the student finishes using the software for the semester. Even students who did not use it in a section where it is offered are surveyed to find out why they did not do so.

In Spring, Summer, and Fall semesters at ASU, a total of 561 students in 11 sections of our course EEE 202 (Circuits I) used our software (defined as completing one or more of the three available tutorials). Over 80% of these students completed all three tutorials. In addition, 42 students used our software at the University of Notre Dame in EE 20224 (Introduction to Electrical Engineering), which has a typical class size of 50-60 students, and also a few at the University of Virginia, at Chandler-Gilbert Community College, and at South Mountain Community College. During this period, we recorded over 101,000 log entries on our server while students analyzed over 7,000 different circuits, which provides a wealth of data to analyze.

---

Table I. Sample Verbatim Student Comments on Software.

---

**Series-Parallel Tutorial:**

- This game helped out a lot with my understanding of parallel and series circuits. The colors of the nodes in the tutorial helped explain a lot also.
- Awesome! a little confusing at first but once you got the hang of it...
- At first i thought was really stupid and hard to use but then it ended up turning out pretty helpful to identify the series/parallel connections
- Great explanations, I look at elements in parallel completely different than I did before.
- The most important concept I was able to grasp was how to properly think of these circuits in nodes, thanks to the coloration of the easier levels. This was a huge leap, as now I can easily identify the circuit and think through problem
- It was fun and very interactive. I liked doing it a lot.
- I needed this repetitive practice to master the application of the concepts in class.
- Explain why some of the series connections work. I didn't understand why some of the less obvious ones were correct in the beginning.

**Node Equation Tutorial:**

- very informative and the examples shown are a great help in figuring out how the equations are calculated. One suggestion would be to have a short exercise explaining the process of how the equation is gotten
- I'm really glad that I was able to work on supernodes, as that's one of my bigger weaknesses about this. I also like the fact that dependent sources were worked on as well.
- I feel much better about nodal analysis and super nodes!
- This game was difficult. I enjoyed it though!

**Mesh Equation Tutorial:**

- I really enjoyed these exercises and wish that we had this tool for all of our lessons. This gave me the clearest explanations of what to do. Could have been slightly more detailed for the loops with dependant sources.
  - As always, very helpful! A tutorial on what constitutes a supermesh and what has to be included would be nice.
  - I like to have immediate feedback and the ability to go through the examples. this is a good product
- 

A simple two-question survey is administered at the end of each tutorial. Students are asked if the tutorial was “very useful,” “somewhat useful,” “not very useful,” or “a waste of time” for improving their understanding of the topic. The results this year are 71% very useful, 27% somewhat useful, 2% not very useful, and 1% a waste of time. Thus, 97% rate them as very or somewhat useful. The ratings are similar among the three tutorials, where students are asked to identify elements that are in series and parallel with each other, to write node equations for a given circuit, or to write mesh equations for a given circuit. The equation writing games have somewhat higher “very useful” ratings than the series-parallel exercise.

Open-ended student comments and suggestions are also solicited in the second question. Some typical comments are shown in Table I. Most were quite positive, though some asked for more help on the user interface (for which we plan to add videos as illustration), for Mac compatibility (which is only possible using an eventual web-based approach), or for additional explanatory feedback on wrong answers.

In the Fall semester, a more detailed survey was sent to all students who used the software and to those who did not do so in sections where it was available. A total of 148 responses were received from students who used the software (107 at ASU and 31 at Notre Dame), and from 97 who did not (81 at ASU and 16 at Notre Dame). The survey for non-users asked

why they did not, and the top reason was having too much other work to do (50%). When asked what would have motivated them to do so more, 60% said they would have done so if they had earned more points towards their grade for doing so, and 50% said they would have done so if their instructor placed more emphasis on their value. Usage rates of 92-95% were achieved in at least six sections whose instructors required and encouraged use of the software, indicating that it has the potential to be quite high.

In the 12-question anonymous survey administered to software users, we attempted to determine if the tutorials were useful and well designed (4 questions), if the coverage and difficulty was appropriate (4 questions), and if they are preferred over conventional exercises and homework (4 questions). A five-point Likert scale was employed with a random mixture of questions where agreement indicated either a favorable or unfavorable view of the software. On average, 87% of student agreed or strongly agreed with favorable views of the tutorials being useful and well designed (with a higher 92% favorable rating from Notre Dame students). A question about being user-friendly received about 80% favorable. On average, 85% of the students agreed or strongly agreed with favorable views of the coverage and difficulty (somewhat more thought they were too easy as opposed to too difficult). An average of 78% of students agreed or strongly agreed with favorable views of the tutorials compared to conventional homework. (Notre Dame students were more favorable in all cases). The lowest favorable response was to the question “If extended to cover all aspects of the material, doing these types of exercises would be more helpful than conventional homework (as a replacement for conventional homework),” where 68% agreed or strongly agreed. We feel that this view will improve when we extend the exercises to cover more of the solution steps in a problem, rather than just setting up equations as is required now. In general, these results suggest high levels of student satisfaction, though further improvements can always be pursued. The high satisfaction at a second, private institution in addition to that at a large public university supports a view that this software can be employed very usefully at a variety of institutions of different types. Further expansion to other institutions including the University of Virginia and the University of the Pacific is planned in the coming year to further test this hypothesis.

Student grades and performance in the above classes have not yet been impacted in a statistically significant way, but this observation is not surprising given the numerous other factors that affect the consistency of instruction and assessment from section to section, and the fact that the tutorials cover a relatively small fraction of the course topics at present. Work is in progress to greatly expand the scope of the tutorials and to optimize student learning in the existing ones, as discussed below. Future assessment will likely be based on experiments using the software itself to do pre- and post-testing, to allow a consistent measurement of student learning under different experimental conditions within the software environment.

#### **4. Formative Assessment Using Log File Data**

As mentioned above, the large quantity of data from our log files can prove very useful to understand how the tutorials are working. This is particularly true for the data from Fall 2013, where we began logging each individual student input as either correct or incorrect. For the exercise in which students are asked to identify sets of circuit elements that either in series or in parallel in randomly-generated circuits of four progressively increasing levels of complexity, we found to our surprise that 51% of sets checked were incorrect. This percentage increased as the difficulty increased, from 39% to 47% to 53% to 57% incorrect. On delving into log files for individual students, we discovered that a substantial number of students did very well, entering very few wrong answers. Other students, however, continued to enter a large number of wrong answers throughout the exercise and clearly were not learning the ideas well. They managed to complete the exercise “successfully” by getting the correct answers eventually, but entered far too many wrong answers in the process as we are not currently penalizing them for doing so. In

a few cases they appeared to be “gaming” the system by entering many responses rapidly, but this did not usually appear to be the case.

As a result of this observation, we learned that we need to tighten up the passing criteria to prevent successful completion of an exercise if too many wrong answers are entered (though students can always do additional problems at the same level of difficulty to show mastery). More importantly, however, we learned that just telling students their answers are wrong, and showing them the correct answers when they give up, is not sufficient as we had initially thought it would be. We need to add a specific explanation about why their answers are incorrect when they enter a set that is not in series or parallel, and perhaps explain why the correct answers are correct. Some students had specifically requested such features in the surveys. These objectives can be achieved with only a small amount of additional programming work. Once we have done so, we will compare the log files to see if the performance of the weaker students improves. If not, we may do some qualitative research where we observe volunteers using the program and experiment to see what kinds of additional explanations are needed. These observations also highlight the fact that high levels of student satisfaction do not always indicate that effective learning is being achieved for all students.

For the node equation writing tutorial, we found again that only 51% of equations being entered were correct. The Kirchhoff’s current law (KCL) equations that are required (the most common type) were 49% correct, the much simpler voltage constraint equations involving voltage sources were 65% correct, and the equations for control variables of dependent sources were only 33% correct. Students only have to enter the latter at the highest of the five levels of difficulty, where dependent sources are introduced for the first time. Clearly, students have special problems understanding what a control variable is, because forming the equations themselves should be straightforward. We again see that we need to provide more explanation of wrong and right answers, beyond showing students the correct answers and graphical aids to understand them, as described previously, and beyond showing an unlimited number of fully worked examples. To do so will require that we modify our equation entry interface to ask students for which node or supernode they are writing a KCL equation, or for which voltage source they are writing a constraint equation, or which control variable they are expressing in an equation. Such an extension was planned in any event. This knowledge is necessary to have a better idea of what a student-entered equation should look like, so that a better explanation and hints can be given. Again this formative assessment approach should help greatly improve the software.

In the mesh equation case, the correct equation percentage was higher at 64%. It is unclear at this point if this is due to greater experience on the part of the students (they now do node analysis prior to mesh analysis), or due to some intrinsic difference in difficulty. This question could possibly be explored by inverting the order in which these topics are treated. The percentages of correct Kirchhoff’s voltage law (KVL) equations, current constraint equations for current sources, and equations for control variables of dependent sources were 66%, 69%, and 35%, respectively. The KVL equations show the biggest difference from the corresponding KCL equations of node analysis, whereas students struggle a great deal with control variable equations in both cases. This information can suggest new directions for additional research into the learning difficulties students are having. Improvements in our feedback on wrong and right answers can be done in this case, just as we plan to do for nodal analysis.

Table II. Learning Gains in Randomized, Controlled Laboratory-Based Study

	Exptl. Condition	Pre-Test Score	Post-Test Score	Gain
Average	Textbook*	58.6	61.6	2.9
Std. Dev.	Textbook	25.3	28.0	14.1
Average	Software**	57.8	86.4	28.6
Std. Dev.	Software	22.1	11.5	14.9
Std. Dev.	Pooled	23.0	20.5	14.1

\*16 users. \*\*17 users.

Even more insight might be obtained by logging the actual circuits students are solving and their actual answers, though this would greatly increase the amount of data and would require development of a “replay engine” to view their work. Such an approach could be pursued if it proves necessary.

## 5. Controlled Laboratory Trial

Assessing the impact of our software on student learning is challenging in a classroom environment, because of the many other factors that come into play, such as variations among instructors, different difficulties of examinations and assignments between semesters, fluctuations in student characteristics, and the fact that the software currently only covers a relatively small fraction of course topics. It is also difficult to construct a controlled experiment and still be fair to all students. We therefore elected to carry out a laboratory-based study using paid student volunteers, all of whom had completed our course in linear circuit analysis during the past year or were currently enrolled in it (but who had not previously used the software). Students were randomly assigned either to use the series-parallel and node equation tutorials for one hour (specified periods for each), or to work textbook problems of similar types for the same period. A pre-test and post-test was administered in each case to assess learning gains (randomly assigning two different forms of the test to average out any difference in difficulty). The Instructional Materials Motivation Survey (IMMS) of Keller<sup>6</sup> was administered to all of the students after they completed the post-test.

The results are summarized in Table II. The learning gains were dramatically (about 10X) larger for the software users compared to the textbook users (~29 vs. 2.9 points). The results were statistically significant with 95% confidence,  $t(19.7) = 3.303$ ,  $p < 0.05$ . The effect size is a Cohen  $d$ -value of 1.21 pooled standard deviations, which is generally considered very large (and larger than that of most intelligent tutoring systems or expert human tutors as noted earlier). The total score on the IMMS survey was 3.54 out of 5.0 (higher being better) for the software users and 3.01 for the textbook users, showing a statistically significant effect size (Cohen  $d$ -value) of 0.91. It therefore appears clear that students are more satisfied and motivated by the tutorials than they are by traditional textbook-based materials, which is in keeping with the results of the surveys administered to classroom users of the materials. We expect that results should get even better if we make the improvements to our software discussed in Section 4.

## 6. Conclusion

We have expanded the usage of our software tutorials for the teaching of linear circuits classes to a total of 613 students in over 12 different course sections over the last year (2013) at several institutions, primarily Arizona State University and the University of Notre Dame. This expansion has been supported by developing an instructor web site to register students and monitor their progress in detail, revamping the software platform to be more convenient, and

logging all student activity to a central server in a FERPA-compliant manner. We are also developing web-based modules for students to draw waveforms and for instructors and others to easily author tutorial sequences. Student satisfaction has been high, with over 97% of students rating the tutorials as “very useful” or “somewhat useful” for learning the topics (74% said “very useful”). Students appreciate the opportunity for unlimited repetitive practice with no penalties for wrong answers, as long as they eventually learn the material. They also appreciate the pedagogical devices we use to promote learning.

Results have been consistent at two very different institutions, suggesting that the materials should be broadly applicable in many settings. A controlled, randomized laboratory-based trial showed approximately 10X learning gains and higher satisfaction levels for the software when compared to conventional textbook-based exercises. Detailed examination of the log files of use in actual courses showed, however, that all students are not learning well from merely being told which answers are correct and by being shown worked examples, even when expository material on the underlying concepts is supplied. Thus, we plan to revise the tutorials to provide better explanations of right and wrong answers, hopefully resulting in even higher learning gains and satisfaction. We feel that our general approach of automatic problem generation and solution could be applied to a variety of other engineering, science, and math courses where many problems of similar types are typically solved.

## 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 M. Ardakani, R. Gorur, G. Karady, B. Matar, C. Tepedelenlioglu, T. Thornton, Chao Wang, Hongbin Yu, and Hongyu Yu for using our software in their EEE 202 sections at ASU and A. Holmes for using it in his course ECE 2630 at the University of Virginia, and for their interest in the project. We thank Daniel Sayre of John Wiley & Sons, Inc. for providing the textbook copies used in our laboratory experiment and for helpful discussions.

## References

- <sup>1</sup>K. VanLehn, “The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems,” *Educ. Psychologist* **46**, 197 (2011).
- <sup>2</sup>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), Session M356.
- <sup>3</sup>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), paper 7940.
- <sup>4</sup>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.
- <sup>5</sup>P. V. Engelhardt and R. J. Beichner, “Students' understanding of direct current resistive electrical circuits,” *Am. J. Phys.* **72**, 98 (2004).
- <sup>6</sup>J. M. Keller, *Motivational Design for Learning and Performance: The ARCS Model Approach* (Springer, New York, 2010).