Inductive Learning Tool Improves Instrumentation Course

Prof. James Andrew Smith P.Eng., Ryerson University

Dr. Smith specializes in Biomedical Engineering at Ryerson University in Toronto, Canada. He was Biomedical Engineering Program Director in 2010/11 and is currently Biomedical Engineering Stream Coordinator. His research combines aspects of biomechanics and robotics, with active research projects in legged systems, obstetrics and surgical systems. In addition to teaching awards received at the University of Alberta and Ryerson University, he is a recipient or co-recipient of four IEEE Real World Engineering Projects awards between 2007 and 2010. He was co-recipient of second place in the 2012 Healthcare Innovation Conference’s design competition in Houston Texas.
Inductive Learning Tool Improves Instrumentation Course
Abstract

Engineering instructors are typically restricted to a narrow range of simplified models that apply to both white-board lectures and hands-on labs. This is as true in a mechanical vibrations course as it is in electrical circuits or biomedical instrumentation courses. More complex systems, which are more representative of reality, require more time or more advanced tools for derivation or understanding than is often available in an undergraduate curriculum.

In 2011 we addressed this disconnect by introducing our students to Maplesoft’s MapleSim multi-domain simulation software in an instrumentation course. MapleSim permits students to schematically represent electrical and mechanical systems and simulates their changes over time. Using Maplesoft’s Maple application, MapleSim can then produce the analytical equations which underlie the model in a form that is similar to what is shown to the students in class or in their textbooks. This process can be self-directed and can occur even if students don’t have advanced skills in the mathematics required to derive the equations by hand. This tool, used in conjunction with traditional hands-on electronics lab activities permits the students to explore the behaviour of the systems of interest in an inductive learning manner more representative of natural everyday learning. To illustrate the inductive process taken by students, a typical example undertaken by students of examining temperature effects on a resistor and diode circuit is given.

Using MapleSim and Maple in an inductive context, it is now far easier for the student to use basic concepts conveyed by the textbook or standard lectures, vary them in subtle or extreme ways and to see the effect not only on the numerical simulation output but on the underlying mathematics as well. Examples that were not practical to attempt by anyone but the brightest students are now within the reach of motivated and curious students.

Student evaluations have shown an improvement since the introduction of the inductive approach along with Maple and MapleSim. The positive response of students to the use of MapleSim as a front-end tool and Maple as a support tool has encouraged us to use it as the core of a new distance education course in embedded systems architecture.

Introduction

The ELE 604 Sensors and Measurement class at Ryerson University is presented to Electrical and Computer Engineering students in the third (junior) year of the undergraduate program. The objective is to expose students to basic instrumentation systems, including both analog and digital aspects. The laboratory component focuses on the development of a microcontroller-based instrumentation system that can provide readings to a PC and an onboard LCD from switches, accelerometers and load cells.

The course is heavily-focused on hands-on work in the laboratory. From student feedback in 2009 and 2010 it became apparent that there was a mismatch between the theoretical background provided in class and the practical work being undertaken in the laboratory. This is reflective of the general approach taken in Engineering classes. Engineering instructors are typically restricted to a narrow range of simplified models that apply to both white-board lectures and
hands-on labs. This is as true in a mechanical vibrations course as it is in an electrical circuits or biomedical instrumentation courses. More complex systems, which are more representative of reality, require more time or more advanced tools for derivation or understanding than is often available in an undergraduate curriculum. Furthermore most subject-specific simulation products do not provide a method for linking the GUI-based model’s dynamics with the fundamental equations used to model the simpler examples in class.

In 2011 we addressed this disconnect by introducing our students to Maplesoft’s MapleSim\textsuperscript{2} multi-domain simulation software in our instrumentation course. Like its competitors, MapleSim permits students to schematically represent electrical and mechanical systems and simulates their changes over time. It can also produce the analytical equations which underlie the model in a form that is similar to what is shown to the students in class or in their textbooks. Students can examine these equations to derive further insight into the dynamics of their systems. This process can be self-directed and can occur even if students don’t have advanced skills in the mathematics required to derive the equations by hand. This tool, used in conjunction with traditional hands-on electronics lab activities permits the students to explore the behavior of the systems of interest in an inductive learning manner more representative of natural everyday learning.

Observations

The learning strategy that underlies the introduction of MapleSim is that the professor-driven deductive learning approach of memorizing rules should be complemented with a student-driven inductive approach of determining intuitive rules through experience. The inductive approach\textsuperscript{3,4} is more natural and typically leads to a deeper, more thorough understanding of completed material -- it is also in stark contrast to the artificial deductive approach we experience so often in school. A comparison of the rule-to-behavior approach used in deductive learning versus the behavior-to-rule approach used in inductive learning is shown graphically in Fig. 1.

Watch how a three year old learns how to play videos on an iPhone. No one tells her how the capacitive sensing array under the glass works -- she simply moves and swipes, sometimes succeeding, sometimes failing at making the device work the way she wants. These hands-on, trial-and-error tests induce a change in how the child uses the phone. After a few attempts of learning like this, children are often more adept at using the phone than their parents are! It’s the opinion of the author that this kind of learning should be encouraged within the standard engineering curriculum.

Here we describe an example of using MapleSim in an inductive manner during a laboratory exercise. In the three years since introducing MapleSim as an inductive learning tool the student feedback has grown more and more positive, as is reported later.
Figure 1 The traditional deductive learning approach (left) compared to the inductive approach (right). The deductive approach focuses on rule-learning first, while the inductive approach emphasizes exposure to system behaviour first.

Methods

Over a 12 week period students are tasked with developing an instrumentation system from scratch using a Freescale 9s12 microcontroller.\textsuperscript{5,6} They design and implement circuitry and software which link the 9s12 to sensors and a desktop computer. Here we look at two introductory activities which use Maple and MapleSim in the learning of fundamental instrumentation concepts.

In the fourth and fifth weeks of the class students examine methods for driving LEDs and switches on the 9s12. One of the important aspects to explore is the temperature dependence of the instrumentation system. It is impractical to do cold weather testing within the laboratory so students explore the effect that temperature will have on the LED’s light through simulation.
The students set up a simple LED schematic in MapleSim, as shown in Fig. 2a. Normally students would set the resistance and temperature values to very specific values. However, here the numeric values are replaced with parametric values ResistorValue and TemperatureValue, as shown in Fig. 2b. Also, two probes have been inserted to measure diode voltage and current. These are given variable names DiodeVoltage and probecurrent. The model will then be called from Maple, where the resistance and temperature values are set. The voltage and current values will be returned to Maple. The MapleSim model is called from Maple using the Simulate command. This is similar to how Simulink models are called from Matlab. The main command for this is

\[
\text{simData := A:-Simulate(output = datapoint, duration = 1.0, params = ['ResistorValue' = 100, 'TemperatureValue' = 300])}
\]

where the resistance has been set to 100 Ohms and the temperature has been set to 300 Kelvin. The more detailed set of commands are given in Fig. 5 in the Appendix.

The result is a set of three numbers giving the time, current value and voltage value from the simulation. While students could explicitly run a few dozen simulations like this to determine the relationship between resistance, temperature, voltage and current, it is much more efficient to have the students run a parameter sweep, instead.

The parameter sweep centers on a nested For loop and outputs a pair of surface plots that show how voltage and current vary based on changing temperature and inline resistance. These are shown in Fig. 3, below. The Maple script for executing this is shown in Fig. 6 to 9 in the
Appendix. Without knowing the fundamental mathematical models for the resistor or diode the students can visualize how diode voltage and current are highest for the lowest resistance and temperature.

Figure 3 Parameter sweep results: highest current and voltage are found when resistance and temperature are lowest.

These graphs permit the student to observe the behaviour of the system as it is subjected to variations in system parameters. The rules governing the behaviour can be induced from these graphs, without requiring the student to perform high-level mathematical analysis. This type of parameter sweep could be performed in any number of software packages, including Matlab or a PSPICE variant. What distinguishes Maple and MapleSim from these other programs is the ability to generate the underlying equations of the system. This is a critically important feature.

MapleSim contains an “attach equation” option which populates a Maple worksheet with live interactive hooks to the MapleSim file. Three equations are created by the worksheet which describe the mathematical rules governing the system:

\[
V_{batt} - OutputVolt(t) - D_\_s(t) \cdot \left\{ \begin{array}{ll}
D_{\_Goff} & D_{\_off}(t) \\
1 & \text{otherwise}
\end{array} \right\} - D_{\_Goff} \cdot D_{\_VKnee} = 0
\]

\[
D_{\_off}(t) = D_{\_s}(t) < 0
\]

\[
OutputVolt(t) = D_{\_s}(t) \cdot \left\{ \begin{array}{ll}
1 & D_{\_off}(t) = \text{true}

D_{\_Ron} & \text{otherwise}
\end{array} \right\} + D_{\_VKnee}
\]

where \( V_{batt} \) is the input battery voltage, \( OutputVolt(t) \) is the voltage of the output voltage probe, \( D_{\_VKnee} \) is the knee voltage of the Diode. Other characteristics of the Diode include the conductance in the off state, \( D_{\_Goff} \), the resistance in the on state, \( D_{\_Ron} \), and the saturation current, \( D_{\_s}(t) \). The Diode’s on/off state is recorded by \( D_{\_off}(t) \) is true when the diode is off and false when it is on.
The traditional deductive approach to teaching this material would require the student to have a relatively extensive foundation of mathematical and electrical engineering knowledge before being able to determine these equations by him or herself. The student’s exposure to the classroom professor and teaching assistant is limited and the relevant examples in the textbook or on the Internet tend to be simplified and repetitive.

The MapleSim approach provides a far more flexible and accessible resource to the student than is possible in the classroom or in the standard reference material. Can the student simply hit the “attach equations” button and submit the result for homework? No. The equations typically require some understanding of the underlying variables and structure. The student must still have some basic understanding of what many of the components are, and will benefit from being able to discuss questions and concerns with a teacher or teaching assistant. However, it is now far easier for the student to use basic concepts conveyed by the textbook or standard lectures, vary them in subtle or extreme ways and to see the effect not only on the numerical simulation output but on the underlying mathematics as well. Examples that were not practical to attempt by anyone but the brightest students are now within the reach of motivated and curious students.

Results

Inductive learning is very effective and the author started using new software in class that is designed specifically to work with this approach. Using MapleSim students start by drawing the system schematic, and then simulate it. Then they extract the underlying equations in the software, explore them using different scenarios, and analyze the equations to derive conclusions. The best part of this for students is that they can match it with what they are seeing in their textbooks, as the simulation process they go through is the same as in the textbook. Students in ELE 604 value the inductive approach. Since 2010 I have made the course more inductive and less deductive. The students' answers to the two pertinent questions on the Faculty Course Survey, Q1: “Concepts are clearly explained with appropriate use of examples” and Q2: “The way this course is taught helps me to learn”, shown in Fig. 4 provide good evidence that the approach is working.
Figure 4 Student evaluations have shown notable improvement. In particular Question 1: “Concepts are clearly explained with appropriate use of examples” and Question 2: “The way this course is taught helps me to learn” illustrate that since inductive tools were introduced and emphasized in class the students have given more positive ratings.

Perhaps the most significant feedback was the following comment from a student on the student evaluations in 2013, illustrating that students get this inductive approach:

“The professor for this course was very good. His approach in teaching the material was very helpful. He decided to teach us the material in a way where we learned the material through the labs and from the labs we would then learn the theory. This was an effective method of teaching, since the labs made me learn the material before the lectures and then understand the ‘why’ of the material I learned in the labs through the theory presented in the lectures. ... Overall, the professor was friendly and really helpful with the students -- he really wanted us to learn the material and see the actual application of it in the real world.”

As seen in Fig. 4 it took two years to see a real change in student response to the introduction of MapleSim as an inductive learning tool. The main reasons were that it initially was used only in the laboratories and without accessing the equation extraction feature. Over time it became more integrated in classroom exercises, the students began using the equation extraction feature and the ability to use components outside of the electrical engineering domain (from the mechanical toolboxes, primarily). In 2014 more Maple-only exercises (without MapleSim) were introduced during classroom discussions to illustrate methods for solving mathematical equations related to instrumentation problems. Students have reported that they are now self-selecting Maple in their Control Engineering class rather than the instructor’s preferred tool, Matlab. This could be due due to the Maple interface more closely resembling their textbooks and lecture material.

Finally, in 2014 MapleSim will be introduced in an online distance education course at Ryerson University, CKRE 130 Embedded Systems Hardware Architecture and Implementation. The possibility for self-directed learning using MapleSim is particularly attractive in the context of this course and the lessons learned from teaching with it in ELE 604 will be applied.

Conclusions

End-of-semester student evaluations have shown a quantifiable improvement in the course since the introduction of the inductive approach. As MapleSim and Maple has become more integrated into the course question categories such as “The way this course is taught helps me to learn” and “Concepts are clearly explained with appropriate use of examples” have seen quantifiable improvements over the past three years. The positive response of students to the use of MapleSim and Maple has encouraged us to use it as the core of a new distance education course in embedded systems architecture. Students are now electing to use Maple rather than Matlab in other engineering courses, without instructor intervention, likely due to Maple’s interface more closely resembling lecture and text material. The inductive approach is
complementary to the traditional deductive approach and should at least be considered for portions of courses where student self-exploration could result in positive educational outcomes. With insightful and intuitive tools like MapleSim and Maple it is far easier to find an appropriate mixture of both of these styles within your own courses.

Bibliographic Information


Appendix: Maple script codes to run parametric searches in a MapleSim model.

--- Part 1: Setup Maple & Locate MapleSim Model File -----------------------------
restart;
gc();
with(ExcelTools);
with(LinearAlgebra);
with(plots);

# Define the directory with the MapleSim model file
switchdir := "C:/Users/jasmith/research/Maple/MapleSim/diode_resistor": # Modify this.
currentdir(switchdir):

# Link the MapleSim model file
A := MapleSim:-LinkModel(filename="dioderesistor3_jan2013.msim") : # Modify this.

#(put your cursor here and hit ENTER to execute this part of the script)

--- Part 2: Run the MapleSim Model. ---------------------------------------------
# Call the MapleSim Model once. Explicitly set the resistance and temperature values here.
sinData := A:-Simulate(output = dataset, duration = 1.0, params = [ ResistorValue = 100, TemperatureValue = 300 ]): # Modify this.

# Examine the three output columns in sinData.
# Choose the 200th row in each column to give system time to settle.
sinData[200,1]: # Examine the 1st column of output data, time. (units: seconds)
sinData[200,2]: # Examine the 2nd column of output data, current. (units: amperes)
sinData[200,3]: # Examine the 3rd column of output data, voltage. (units: volts)

#(put your cursor here and hit ENTER to execute this part of the script)

Figure 5 Simple example of calling the MapleSim model from Maple a single time.

--- Part 1: Setup Maple & Locate MapleSim Model File -----------------------------
restart;
gc();
with(ExcelTools);
with(LinearAlgebra);
with(plots);

# Define the directory with the MapleSim model file
switchdir := "C:/Users/jasmith/research/Maple/MapleSim/diode_resistor": # Modify this.
currentdir(switchdir):

# Link the MapleSim model file
A := MapleSim:-LinkModel(filename="dioderesistor3_jan2013.msim") : # Modify this.

#(put your cursor here and hit ENTER to execute this part of the script)

Figure 6 Parameter Sweep, Part 1: Setting up Maple and MapleSim for the parameter sweep. tell Maple where and which files to simulate. Then run the simulation specifying which parameters to modify.
Figure 7 Parameter Sweep, Part 2: define parameters. When parameter sweeping you need to define the boundaries. Here's how you do it. The first parameter can be resistance, and the second temperature.

```plaintext
# Part 2: Define Parameter Boundaries and Create Sequences for the Parameter Sweep

> # Here you define the boundaries of the two parameters.
> # Minimum values for the first and second parameters are set by FirstParameterMin and SecondParameterMin.
> # Maximum values for the first and second parameters are set by FirstParameterMax and SecondParameterMax. Use
> # FirstParameterMaxIndex and SecondParameterMaxIndex to define the number of discrete intervals between minimum and maximum
> # values.
> # First Parameter: Resistance inline with the diode
> # Second Parameter: Temperature of the system

FirstParameterMin := 50; FirstParameterMax := 1000; FirstParameterMaxIndex := 10; # Modify this (resistance values)
SecondParameterMin := 250; SecondParameterMax := 400; SecondParameterMaxIndex := 8; # Modify this (temperature values)

# Create a sequence of values that you will sweep through. First, do it for the first parameter...
FirstParameterValues := evalf( seq(FirstParameterMin .. FirstParameterMax, (FirstParameterMax - FirstParameterMin) / (FirstParameterMaxIndex - 1)) [1 .. FirstParameterMaxIndex] );

# ... next do it for the second parameter.
SecondParameterValues := evalf( seq(SecondParameterMin .. SecondParameterMax, (SecondParameterMax - SecondParameterMin) / (SecondParameterMaxIndex - 1)) [1 .. SecondParameterMaxIndex] );

#( put your cursor here and hit ENTER to execute this part of the script )
```

Figure 8 Parameter Sweep, Part 3: run the parameter sweep. Vary each of the parameters and accumulate the resulting outputs from the MapleSim simulation in two columns.
Figure 9 Parameter Sweep, Part 4: plotting a pair of surface plots to visualize the result of the parameter sweep.