

# Teaching Practical Software Tools Using Screencasts While Simultaneously Reinforcing Theoretical Course Concepts

Jon Ashdown, David Doria, Michael Wozny

January 14, 2011

## Abstract

In this paper, we address a frequent complaint of students in technical fields that they are expected to simultaneously learn course material while learning software tools required for a course. Additionally, the learning of these tools is often expected to be self directed, rather than formally taught. We propose the use of screencasts to teach the software packages while at the same time reinforcing course concepts. We discuss our implementation of such a solution to this problem in a junior level electrical engineering course at an engineering focused university in the northeast. We also present survey responses from students who took the course which demonstrates that they enjoy and benefit from this method.

## 1 Introduction - Using Screencasts to Augment a Traditional Signals and Systems Course

After many years of taking and teaching engineering courses, we have realized that students are typically expected to simultaneously learn course material while learning software tools required for a course. Additionally, the learning of these tools is expected to be self directed, rather than formally taught. We first considered teaching separate courses on the tools themselves. However, modern curricula are typically overfull as it is, so introducing more required courses was not feasible. Rather than teach these tools separately, we must integrate the instruction into the existing courses. However, a similar problem to the curriculum crowding one exists inter-course as well. That is, the course calendar is already full, so where would this new material fit? We claim that rather than take time away from the course, these tools can be taught while simultaneously reinforcing course material. To accomplish this, we propose the use of screencasts to teach the software packages while at the same time following up on the lecture material presented in the course. We discuss our implementation of such a solution in a junior level electrical engineering course, Signals and Systems, at an engineering focused university in the northeast. We also present survey responses from students who took the course which demonstrates that they enjoy and benefit from this method. In the Signals and Systems course, students are encouraged to make connections to practical ideas in otherwise theory based courses. At the beginning of the

course, students had trouble making these connections. We expected students to be very proficient at learning new software and applying theory to practice to solve real world problems. We were quite surprised that this was not the case. Although Generation Z students are generally “good with computers” (Posnick-Goodwin, 2010), without the proper training and instruction on how to use the technology even the best students can have trouble and may become frustrated. Students are often expected to solve challenging problems using MATLAB with little or no formal instruction. After recognizing this deficiency in the approach, we decided to offer students some help by teaching formal lectures in how to use MATLAB to model, simulate, and solve challenging real world problems to reinforce the theory being taught in the course.

## 2 Related Work

The concept of using screencasts in a learning environment is not new. In (Peterson, 2007), screencasts were used to teach the Dewey Decimal Classification System. In (Peterson, 2008), screencasts are used to deliver detailed solutions to homework problems. In both cases students found the screencasts very helpful. In (Allison Carr, 2009), screencasts were used to demonstrate procedural processes. These screencasts simply serve as a replacement for a linear, textual “how to” explaining how to carry out a specific task. Our contribution differs from the above uses of screencasts as we are using screencasts as an enhancement to existing course material while simultaneously teaching concepts in the specific software tool we demonstrate in the screencasts.

## 3 Screencasts

### 3.1 What is a Screencast?

A screencast is a digital recording of computer screen output, also known as a video screen capture, often containing audio narration. The concept was first introduced in 2005 in (Notess, 2005). There are many software programs available to produce screencasts. We selected to use Camtasia Studio (*Camtasia Website*, 2011). Figure 1 shows a frame of a typical screencast.

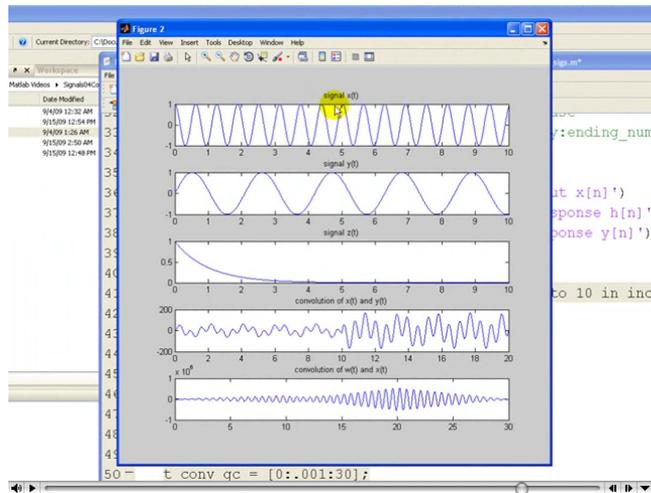


Figure 1: A frame of the “Convolution of Signals” screencast.

## 3.2 Selected Screencast Details

To test the concept of using screencasts to teach software tools while augmenting the course material, we selected a junior level electrical engineering course, Signals and Systems. We created a series of screencasts which taught students the fundamentals of MATLAB by demonstrating how to solve typical problems that would be encountered in the course. These screencasts simultaneously addressed two frequent concerns of students. First, students often express frustration over the fact that they are expected to learn these large software packages by themselves. We have found that this generally results in a less than complete understanding of the software. This often leads to the implementation of ineffective and inefficient solutions. Second, students often request less abstract and more “real world” type problems to practice the theoretical concepts presented in class. Our series of screencasts presented the basics of the software using exactly this real world example, so both objectives were achieved simultaneously.

We created a total of 10 screencasts, which are available here (*MATLAB Screencasts*, 2011). In this paper, we will focus on a small selection of them which highlight different benefits.

### 3.2.1 “Representing and Plotting Basic Signals”

As this is this first screencast in our series, we use it to introduce several fundamental MATLAB concepts. For instance, we provide instruction in how to create and manipulate arrays, create and label plots using the *stem* and *plot* commands, and how to use independent variable indices to increase resolution. These important MATLAB topics are introduced while at the same time reinforcing the concepts of the representation of discrete and continuous time signals, which is a fundamental concept in the Signals and Systems course.

### 3.2.2 “Convolution of Signals”

In this screencast, we introduce the tools needed to perform the operation that is essential to a Signals and Systems course - convolution. At the heart of this manipulation is the *conv* command. This screencast also provides further instruction on selecting appropriate axes scales using the *axis* command. These MATLAB concepts are taught while simultaneously reinforcing concepts from the course including using convolution to find the output of a Linear Time-Invariant (LTI) System given the input signal and impulse response, and properties of convolution (commutative, distributive, and associative). Furthermore, several real world examples are given as practice to further reinforce the learning.

### 3.2.3 “Even/Odd Signal Decomposition”

In this screencast, we teach students how to manipulate arrays using the *fliplr* (flip left/right) and *flipud* (flip up/down) commands. We cover the difference between row and column vectors, and show students how to take the transpose of arrays. It also brings the students further in formatting plots and covers the *hold* command which allows for the overlaying of multiple plots on the same figure. These MATLAB concepts are covered while concurrently reinforcing essential course concepts which include flipping, shifting, and scaling of signals, as well as decomposing a signal into its even and odd components.

### 3.2.4 “Programming With Applications to Signal Processing”

This screencast delves into some of the fundamental programming tools that are essential for successful programming in MATLAB. Some MATLAB concepts that are covered include commenting, for loops, while loops, conditional statements, and logical operators. Several real world signal processing examples were covered to help reinforce concepts in the course while teaching basic MATLAB programming. For instance, one of the simple examples was to read in and sample an audio signal, loop through the contents of the audio vector and add a slightly delayed version of the original signal back to itself. When the students played the resulting audio signal back in MATLAB, they noticed an echo effect. This concept was then expanded to develop various reverb effects that resembled those used in professional music recordings.

## 3.3 Detailed Example of a Real World Problem: Separating Bird Chirping from Wind Using Filtering

The use of software tools inherently allows the complexity of the problems to increase while keeping them solvable in a reasonable amount of time for a homework assignment. One example of a MATLAB demonstration performed in the classroom that sparked interest among students involved filters and Bode plots. The course theory teaches high-pass, low-pass, band-pass, and notch filters. Traditionally, the instructor draws them on the chalk-board and teaches the students about gain, cutoff frequency, Bode magnitude plots, Bode phase plots, and Fourier transforms. Time is then spent teaching Bode approximations, which make filters easier to sketch by hand. These details are important, but we wanted to present them in a way that would make filters come alive for the

students. The goal was to explain these things in such a way that the students would be excited to share what they have learned with their friends.

To do this, we recorded some birds chirping on a windy day using a condenser microphone. The high-pitched bird chirps could be heard when we listened to the recording. However, the recording had been compromised by the comparatively low frequency howl of the wind. We demonstrated the design of a high-pass filter in MATLAB with the appropriate cutoff frequency to remove the low frequency wind noise while retaining the beautiful bird chirps. We then created a filter design assignment to give the students practice in removing unwanted noise sources from audio files. This assignment was inspirational for many students who enjoyed seeing the theory applied to practice to solve problems that were relevant to real life. The students walk away with a concrete example of how powerful and useful these techniques can be.

## 4 Student Survey Results

We performed a short survey of the students in the course for which we created the MATLAB screencasts. Our goal was two fold. First, we wanted to verify that the screencasts were helpful to students. Second, we wanted to get an overall picture of the use of technology in the classroom. Specifically, we wanted to understand the students view of the current level of instruction as well as their need for additional instruction in various software tools. Thirty students responded to the survey.

### 4.1 General Technology

Given the choice of learning material from a textbook, a screencast, or a lecture, only 10% opted for textbooks, while 43% of students preferred screencasts, and 47% preferred in class lectures. Of the respondents, 90% percent of students responded “Seldom” or “Never” to the availability of video lectures or screencasts in their courses. The high number of students who prefer in-class lectures is perhaps not surprising due to the rich nature of interfacing face to face with an experienced professor. However, it is notably significant that such a large percentage of students preferred screencasts when learning course material.

### 4.2 General Theoretical/Practical Balance

When asked if instructors should demonstrate practical applications of the theoretical concepts taught in class, 91% said yes. Ninety-five percent of students indicated that they would like to see less time spent on theoretical homework problems which they viewed as “busy work” and more time spent on practical, “real world” problems. On the survey, we asked a free response question: “If you could change anything to improve the quality of your education, what would it be?”. A majority of the responses indicated that more “hands-on”, “practical” homework problems would enhance the learning experience.

### 4.3 Software Tool Instruction and Usage

There are several software packages/tools that students are required to use throughout their engineering education. However, adequate training/instruction

in how to use these tools is often overlooked which can lead to a suboptimal education for students. In a typical electrical engineering curriculum, software packages that students are required to know how to use include MATLAB, Simulink, PSPICE, Cadence, VHDL, and several others. Our survey results indicate that students were often required to use software packages but seldom given any training in how to use them. Among the most common software packages that students are required to use include C++, C, PSPICE, Cadence, MATLAB, Maple, and Simulink. Of these, a majority of student survey respondents reported receiving little or no instruction in the use of these important tools. The exceptions were C++ and MATLAB. Seventy-seven percent of students reported receiving C++ instruction often or very often. This is because there are computer science courses that are required in the curriculum which dedicate a full semester to programming instruction in C++. Sixty percent of students reported receiving instruction in MATLAB often or very often which is a result reflecting our use of screencasts in the curriculum. For the other commonly used software packages, a majority of students reported seldom or never receiving instruction in the use of these tools.

It is very important to effectively and correctly match the amount of required use of a software tool and the amount of instruction given in that tool. In one case, there are situations in which students are given an abundance of training in a software tool that they will seldom be required to use. For instance, most students are required to take a computer modeling (CAD) course as part of their core curriculum. Depending on the type of engineering in which they decide to specialize, they may or may not use the skills acquired in this course to any significant extent. Among the electrical engineering students we surveyed, many report having a plethora of instruction in the use of CAD but were seldom or never required to use it throughout the rest of their tenure at the institute. A much more ideal case is the scenario in which students receive significant instruction in a tool that they will be required to use frequently throughout their education and careers. A prime example is C++. The majority of surveyed students reported having adequate instruction in C++ and also reported frequently being required to use these skills throughout their education. Finally, there is the situation about which we are most concerned: when students are frequently required to use a tool but are not given adequate instruction in how to use it. Our survey reflects that MATLAB fell into this category. Although the screencasts improved the situation dramatically, similar efforts must be made in adjacent areas to fix this common problem.

#### **4.4 Our Screencasts**

The survey results indicate that students enjoy and prefer this modality of learning. Over 70% of students ranked the degree of helpfulness as 6 or above (out of 10).

#### **4.5 Summary**

The feedback from students was overwhelmingly positive. They were not only thankful to receive formal, supplemental instruction in using MATLAB, but were also inspired to learn the course theory and material since now that they

could see its relevance in the real world. Many students who have since gone on to successful engineering careers in communication systems, signal and image processing, or control systems have reported how helpful learning MATLAB in the classroom has been in helping them to excel in their career.

## 5 Conclusion

We have hypothesized and verified that students will benefit and enjoy learning software tools more formally via screencasts rather than learning in an undirected fashion as they progress through a course which uses such tools. A survey of students who had access to the screencasts during their course indicated that a strong majority agreed that they were beneficial to their learning experience. We believe that the materials produced by this course have left a lasting impact on the course. However, there is no reason for its impact to stop at the institute's border. Ideally, faculty in STEM courses around the world can either use these videos directly, or at least take them as inspiration to produce similar, modern educational materials.

## References

- Allison Carr, P. L. (2009). More than words: screencasting as a reference tool. *Reference Services Review*, 37.
- Camtasia website*. (2011). Available from <http://www.techsmith.com/camtasia/>
- Matlab screencasts*. (2011). Available from <http://www.rpi.edu/~ashdoj/MatlabScreencasts.html>
- Notess, G. R. (2005, November). Casting the net: podcasting and screencasting. *"Online" Magazine*.
- Peterson, E. (2007, November). Incorporating screencasts in online teaching. *The International Review of Research in Open and Distance Learning*, 8.
- Peterson, E. (2008, December). *Frontiers in Education Conference*.
- Posnick-Goodwin, S. (2010, February). Meet generation z. *California Teachers Association*.