Teaching Computing To Engineering Freshmen Through A High Tech Tools And Toys Laboratory

S. McKnight

G. Tadmor

E. Carr Everbach
Swarthmore College, ceverba1@swarthmore.edu

See next page for additional authors

Recommended Citation

S. McKnight, G. Tadmor, E. Carr Everbach, W. E. Cole, and M. Ruane. (2001). "Teaching Computing To Engineering Freshmen Through A High Tech Tools And Toys Laboratory". *ASEE Annual Conference And Exposition.* 6.936.1-6.936.11.
https://works.swarthmore.edu/fac-engineering/113

Authors

S. McKnight, G. Tadmor, E. Carr Everbach, W. E. Cole, and M. Ruane

Teaching Computing to Engineering Freshmen Through a "High-Tech Tools and Toys Laboratory"

S. W. McKnight, W. Cole, G. Tadmor, E. C. Everbach, and M. Ruane
Northeastern University / Swarthmore College / Boston University

Abstract

Freshman engineering courses in computing applications and programming often lack applications that are sufficiently engaging without being overwhelming. Program outputs and graphics within the reach of beginning students are often woefully primitive compared to computer graphics that are available in commercial gaming software. The students' lack of background in engineering subjects commonly leads to applications that are simplistic, mechanical (e.g., number sorting), or heavily flavored toward mathematics or physics concepts (such as fluid friction) that student have only an uneasy grasp on. We have developed a two-quarter-hour freshman computing course taught entirely in a "High Tech Tools and Toys Laboratory," equipped with HPIB-bus-linked test and measurement equipment, acoustic and ultrasonic transducers, and stepper-motor-controlled actuators. For a programming environment, we have selected MATLAB with the new Data Acquisition and Instrument Control Toolboxes. This environment will introduce fundamental computing concepts such as looping, conditional branching, and structured programming in a high-level language of continuing utility, but without the complications of variable typing and declarations. Early laboratory experiences include programming loops to cause stepper motors to move a flag, control through photocell feedback, and measuring acoustic velocity and distance by appropriately thresholding a reflected acoustic signal. As a final project, students write a program to control the movement of an ultrasonic sensor to image a metal target encased in an opaque gelatin package.

1. Introduction

Since computation is ubiquitous in engineering practice, a freshman course in computation or computer programming is a feature of most engineering curricula. These courses are often less than satisfactory for both the students and the instructor. While contact with computers is now almost universal in the home or in schools, students' sophistication with computer operations and their knowledge of computer programming vary widely. Many students have completed courses in C or C++ or designed web pages on their own. Other students arrive at the university with only rudimentary knowledge (or less) of directory structures and file manipulation. Teaching across this wide spectrum of backgrounds and abilities is a challenge for the instructor.

This problem is intensified at institutions like our own where students from the different engineering disciplines are mixed during the freshman year. Thus aspiring computer engineers

are in the same class with students intending to be civil engineers who might have no fascination with computer algorithms and program structure. The instructor has to find a delicate balance between losing the less well-prepared students and boring those who have a passion for computers.

In addition to their diversity of preparation and interests, freshmen are also hindered by their lack of a significant technical background in engineering. Real engineering applications and examples are to a large extent precluded in the freshman year, and, as a result, computation and programming are taught either with obvious but uninspiring problems, such as number sorting or character manipulation, or with examples, such as fluid friction or numerical differentiation, that draw on the students' experience in their freshman physics and mathematics courses. While in theory connections with physics and mathematics should reinforce their learning in the other courses, in practice students often flounder with such problems because they are attempting to learn the new language of computing while simultaneously solving problems that are at the edge of their recent, and often uncertain, mastery of physics and mathematics.

Finally, the results that freshmen do coax out of their computer programs are dry and unexciting compared to the dazzling action and graphics of commercial video games and web resources that they are used to. The contrast of the demanding concentration and effort required to get even a simple program debugged and running with the primitive and mundane quality of the output achieved is less than an ideal introduction to the engineering profession at a time when many students are making a decision on whether to continue with engineering in their sophomore year. For many students, freshman computer programming courses reverse Whitehead's dictum[1] that romance should come before precision in the study of science; beginning programming forces the student to submit to an unforgiving rigor to accomplish even simple sorting tasks easily mastered by a child.

The concept of the "High-Tech Tools and Toys Lab" (HTT&TL) arose out of the education thrust of a proposal to the National Science Foundation for an Engineering Research Center (ERC) in Subsurface Sensing and Imaging Systems (CenSSIS) which two of us (SM and MR) were involved in authoring. The HTT&TL Lab was originally conceived to use computers interfaced with test and measurement instrumentation to benchmark, interface, and characterize state-of-the-art commercial subsystems such as Global Positioning System sensors, ground-penetrating radar, digital cameras, and hard disk arrays. The HTT&TL was intended to allow for hands-on measurement of the performance of systems assembled from top-of-the-line component subsystems donated by industry, a kind of discovery lab with expensive parts using the students' fascination with high-technology products to inspire hands-on learning.

In application to beginning engineering courses, we realized that the HTT&TL provided a solution to many of the problems of teaching computation and programming to freshmen. The combination of computers with test and measurement instrumentation is an opportunity to teach computer programming through real-time data acquisition and control. Instead of printing out

"Hello, world!", the first programming exercise could be to set a voltage from 0 to 5 volts, run the program, and see the results on a voltmeter. Instead of printing out a series of characters, the first loop could be used to cause a stepper motor to raise a flag. The advantage of this approach is that programming can be introduced in simple, easily observable sequential steps—move a stepper motor right, turn on an LED, move the motor left—without the more computer-savvy students finding the exercise boring. The experience of watching a computer instruction result in the physical movement of a motor is instantly engaging. Besides being novel and interesting, computer-controlled data acquisition and control requires no prerequisites, does not involve difficult physical or mathematical concepts, and gives immediate and concrete feedback that engages the hands as well as the mind and eyes.

We have created a series of instructional modules using the HTT&TL for application to freshman courses. At Northeastern the HTT&TL is being used in a freshman General Engineering course GE1102: Engineering Problem-Solving Using Software Applications. GE1102 is a two-quarter-hour course that, in the standard freshmen program, is taken in the winter quarter along with full four-quarter-hour courses in physics, chemistry, mathematics, and English. The goals of GE1102 are to teach computer problem-solving skills, to introduce students to some engineering concepts and problems, and to teach two software packages, MATLAB and EXCEL, that will be used as tools throughout the curricula of each engineering department. Since the students are taking four courses in addition to GE1102, the standard, non-HTT&TL, sections are arranged so that much of the course work can be completed in a computer lab during one of the two weekly meetings. This was also adopted as a constraint for the HTT&TL implementation: the course should require a minimum of work outside the single 140-minute weekly meeting in the HTT&TL At Boston University, a HTT&T Lab half-semester module is being designed for a freshman engineering project course which has a goal of introducing students to design and system engineering concepts.

2. Course Design

Our original idea was to use the HTT&TL in a standard problem-solving/programming course using C. We still believe that this would be a fruitful use of the HTT&TL concept, particularly since C is well-adapted for low-level board instructions. Instead, because of scheduling issues, we have piloted the concept in two sections of GE1102. GE1102 is offered the quarter before a full four-quarter-hour course in C, and we believe that GE1102 in the HTT&TL will make an excellent introduction to the programming concepts that are often troublesome in the beginning programming class. Because one of the goals of GE1102 is to teach MATLAB and EXCEL, we made the decision to use the new Data Acquisition and Instrument Control Toolboxes in MATLAB to implement our measurement and control functions.

In fact, we are now convinced that there are good reasons to choose MATLAB in conjunction with the HTT&TL to teach programming concepts, and we are considering this interface for use in the modules at Boston University as well. MATLAB is a widely-used engineering tool that

students can make use of throughout their careers. For many, or even most, one-time analysis tasks, MATLAB is superior to C and may be a preferred tool. MATLAB is a high-level language so it is easier to get results easily, including high-quality graphics that have considerable appeal to students. Syntax details such as type declaration, variable types, and array dimensioning are either not present or are optional. It is, however, a sequential, structured programming language environment that can be used to teach programming concepts such as variables, assignment statements, loops, branching, function calls, and decisions. The concept of arrays is fundamental to MATLAB use, and its use is somewhat more intuitive than in C. It is easier to make connection with linear algebra from mathematics because structures and operations such as row vectors, column vectors, matrices, and matrix algebra are fully supported.

In addition to teaching programming concepts and the specific applications packages (MATLAB and EXCEL) GE1102 in the HTT&TL will serve a number of other worthwhile goals for freshmen engineers. Real-world, laboratory effects such as noise, sampling artifacts, and thresholding techniques are experienced in ways that are not often seen in other freshman science labs. Students will practice reconciling real data with theory and inferring simple mathematical models from measurement data. They will also be exposed to the some of the excitement of engineering and asked to make realistic engineering trade-offs between experimental accuracy and measurement time, learning the sometimes difficult lesson that "good enough" is often preferable to "the best." In the words of one engineering graduate student on having the HTT&TL concept explained to him, "You're teaching the students to be real engineers."

Boston University is developing its own HTT&TL for a first offering in fall 2001. The lab will support two freshman Introduction to Engineering modules, on image processing and on subsurface sensing and imaging. Boston University's lab will develop optical imaging systems, using various commercial sensors (visible and IR cameras) and both commercial and student-written image processing software. Each experiment will be geared to a particular problem, e.g. inspection by color, profile, orientation, or brightness, or visible/IR characterization of an object. Students will develop instrument control, data acquisition and software skills in the context of using full-featured professional equipment.

3. Laboratory Instrumentation

The standard HTT&TL instrumentation is illustrated in the photograph in Figure 1. Standard test and measurement equipment including Agilent programmable multimeter (34401A), power supply (E632A), signal generator (33120A), and digital oscilloscope (54645A) are linked over a bus to a GPIB controller card in the backplane of a standard Pentium II PC. Not visible in the Figure is the National Instruments 6024E multifunction I/O board in the computer backplane and the breakout boxes for the input/output connections. Although representing a substantial capital investment, such instrumentation is by now fairly common in university electrical engineering and engineering technology laboratories where measurements with test equipment are compared to computer simulations on the same bench. Our pilot sections at Northeastern were run in an
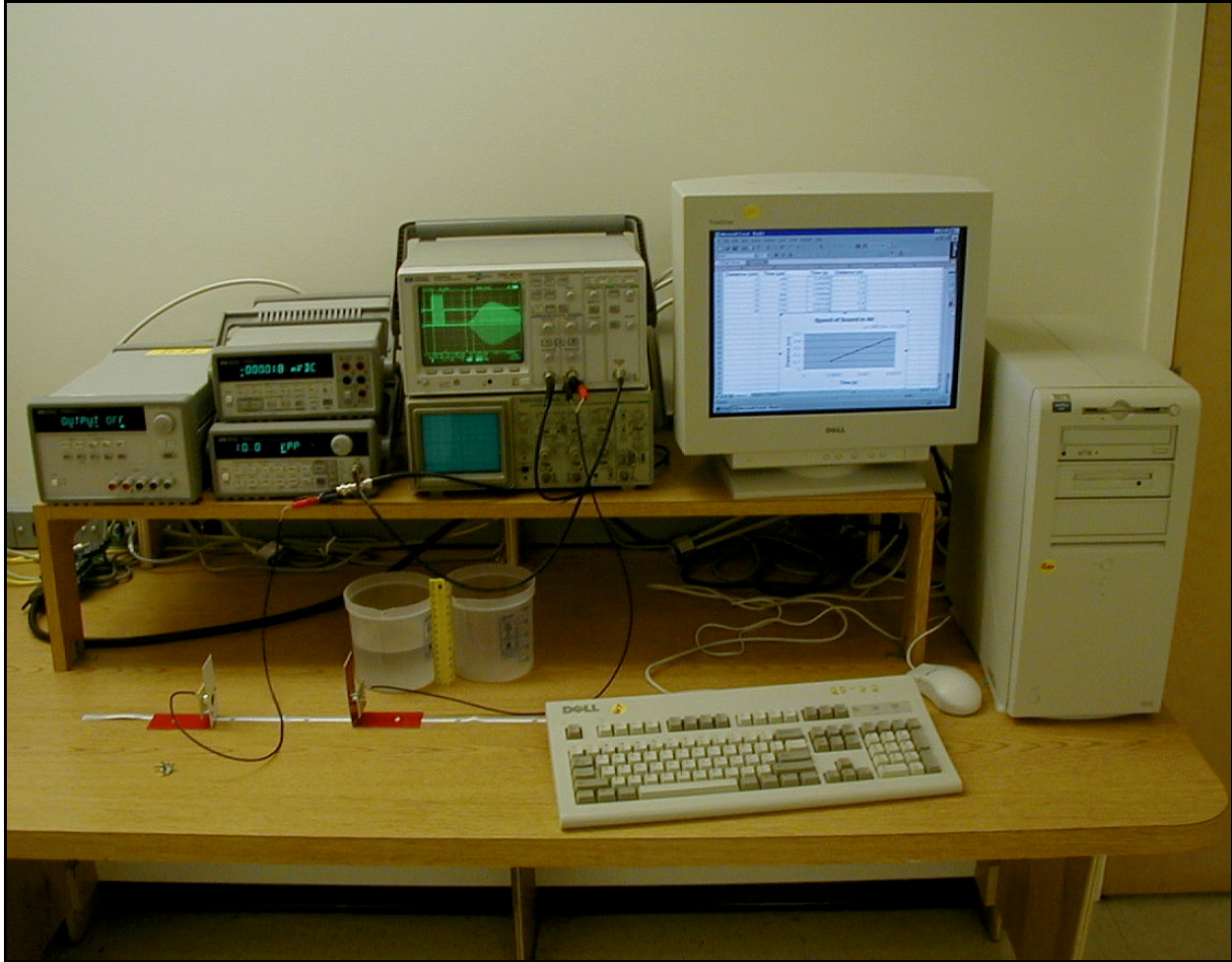
Figure 1. High-tech tools and toys instrumentation set up. The module illustrated in the measurement of the speed of sound in air using acoustic transducers, a signal generator, oscilloscope, and an EXCEL spreadsheet.

existing engineering technology laboratory, and at Boston University a nearly identical setup exists in an electrical engineering laboratory.

The MATLAB Data Acquisition and Instrument Control Toolboxes are new additions to the MATLAB suite of special-purpose software libraries. The Data Acquisition Toolbox consists of a set of MATLAB "m-files" to control input/output from some commercial A/D-D/A boards and software tools to create MATLAB drivers for boards. At present only a few manufacturer's boards are supported, and not all functions on the supported boards can be accessed. This situation will undoubtedly improve in the future but does require some caution in acquiring hardware; we had to replace existing A/D boards with the supported National Instruments cards to be able to run the lab. The Instrument Control Toolbox allows arbitrary instrument control text strings to be sent out on the GPIB bus through the controller card. Again some caution is required because the Instrument Control Toolbox became generally available only with the

release of MATLAB 6.0 (R12) on December 1, 2000, and we found that there were some differences in functioning between National Instrument GPIB-bus cards and Agilent HPIB-bus cards. The overriding advantage of these new toolboxes is that it allows instrument control and data acquisition functions to be carried out in the same environment and with the same syntax as the powerful MATLAB signal processing, computation, graphics, and visualization tools.

There are many alternatives for a "high-tech" systems to be used in the HTT&TL to teach computing. For the Northeastern course, we have selected ultrasonic transducers and positioners, while Boston University is developing their HTT&TL around optical instrumentation. This choice is primarily dictated by available resources and expertise. We were also motivated by a desire to create an experience with subsurface imaging content to develop an awareness among our undergraduates of the CenSSIS theme and mission. Ultrasonic instrumentation is used in the societally important subsurface application of medical imaging. The final project in the GE1102 HTT&TL, imaging the shape of an object inside an opaque gelatin package, will create an immediate connection for the students with medical imaging technology. The connection of engineering practice to technologies that solve important human problems is important in motivating many students to persist in the difficult technical training to become an engineer.

In addition, subsurface sensing with ultrasonic transducers tends to be considerably less expensive than similar experiments with, for example, ground-penetrating radar. The mathematics of ultrasound imaging can be made very simple by neglecting refraction, interference, and diffraction effects. The only mathematical equation we depend on is distance = rate × time. Stepper motors are introduced, first on their own, and then as a part of an x-y positioning system for the ultrasonic transducers. Stepper motors are a good way to introduce concepts of control and feedback (though the use of optical encoders, for example). They are also a satisfying source of immediate gratification: a program step or sequence causes an immediate physical movement. The initial excitement of this manifestation of computer code in gross physical movement needs to be seen to be appreciated.

4. HTT&TL Modules

We have developed seven lab modules to be presented in the winter quarter of 2001 to two pilot sections of GE1102 at Northeastern. Since the quarter has 10 weeks of instruction, Lab 3, Lab 6, and Lab 7 are allowed two weeks each. Each pilot section was limited to 20 students, two students at each experimental station. The activities of each lab are listed below.

Lab 1: Measurement of Speed of Sound in Air
In Lab 1 the students use only the manual controls on the oscilloscope and a pair of 25kHz or 40kHz ultrasonic transducers to determine the speed of sound in air from an EXCEL spreadsheet. The setup is illustrated in Figure 1, which shows the transmitter and receiver on a centimeter scale. The time between the transmitted pulse and the received pulse is determined from the

scope using the manual controls to change the cursor. The resultant data is recorded and plotted in EXCEL, and a straight line fit is made with the EXCEL Trendline. The slope is identified with the speed of sound in air and the y-intercept is related to the offset of the transducer elements with the front of the case. The transducers are then used in a reflection configuration to measure the height of water in a beaker, modeling a real ultrasonic tank monitor. The intensity of the signal can also be measured as a function of distance, but the expected 1/r decrease needs to be distinguished from interference effects. Many real acoustic effects including interference, reflection, diffraction, and antenna patterns can be observed qualitatively, so the lab is amenable to many extensions and optional activities.

Lab 2: Control of Stepper Motor
In Lab 2 students are introduced to MATLAB and create simple MATLAB programs to move a flag attached to a stepper motor through a sequence of steps. Our approach throughout the course has been to shelter the students from the details of the board control. Thus we have created m-files to implement primary controls for the stepper motor: STEP.M, CW.M, and CC.M cause the motor to take one step and change the direction of the movement from clockwise to counter-clockwise. From these primitive commands the students set up loops to move the motor over a number of steps, use a protractor to measure the angle, and calculate the angle per step. By a simple examples, they are lead to estimate the error in their determination and conclude that their measurements are more precise if they use a larger number of steps. The concept of feedback is introduced by having the flag pass over two photosensors monitored by the multimeter. Their final program causes the flag to move from an arbitrary initial position to the first photocell, stop, turn on a light-emitting diode, then reverse directions and go to the second photocell position. The program thus includes sequential programming steps, conditional looping (WHILE statements), and a simple threshold test on the photocells.

Lab 3: Automating an Experiment to Measure the Speed of Sound in Air
Lab 3 repeats the material of Lab 1 except that the students now use MATLAB to record the receiver signal, measure the time delay, and do the linear regression to find the velocity of sound in air. Since this lab requires familiarity with MATLAB arrays and plotting functions, it is carried out in two sessions. The first week the students complete exercises in MATLAB and create a simulated ultrasound detector signal. In the second week they write a MATLAB program to sample and record the received signal from the detector, take the absolute value of the signal, determine if it lies above a selected threshold, and, if so, calculate the time delay of the received signal from the transmitted pulse. Sampling is done by using another primitive program ECHOREAD.M which is written for the students. ECHOREAD includes calls to the Instrument Control Toolbox which read in data from the digital oscilloscope and sample the waveform at a supplied frequency. The use of the Instrument Control Toolbox with the 100 MHz scope allows a clean signal to be obtained of the waveform of the 40 kHz transducers, even if down-sampled by a significant factor. The trade-off speed and accuracy for different sampling rates will be observed. After some simple signal preconditioning (adjusting the average signal to zero and taking the absolute value) the student will select a threshold test criteria using MATLAB

conditional logic (IF or FIND statements). The results at a number of source-receive separations will be fit to a straight line using MATLAB's POLYFIT command, and the slope extracted as the velocity of sound in air.

Lab 4. Control of X-Y Positioner
In Lab 4, we move to a different instrumentation, a 10cm x 10cm travel x-y positioner (Velmex, Inc) mounted on an aquarium tank to control the position of a 1 MHz ultrasonic transducer in water. In this lab, the positioner will be used to control the position of a felt-tip pen to create patterns on a piece of paper. The positioner is run by two stepper motors using the same primitive MATLAB m-files as previously. Students will be asked to create a pattern (the letter F and, perhaps, their initials) using a program of their own design. They will also write a program to create a raster scan, in preparation for the imaging labs to follow.

Lab 5. Speed of Sound in Water
Lab 5 recreates the measurements of Lab 3 using a 1 MHz ultrasound transducer in a water instead of the 40 kHz transducers in air. A single transmitter/receiver is used in a reflection mode. The time delay in the signal reflected off the side wall of the aquarium is measured as a function of the x-position of the transducer controlled by the positioner. A linear regression fit determines the speed of sound in water. Experiments with partial transmission through films and objects will prepare for the subsurface imaging experiment in Lab 7.

Lab 6. Imaging an Object by Reflection
In Lab 6, the ultrasonic transducer is raster-scanned with the x-y positioner and the time delay of a pulse reflected from an object on the aquarium wall is plotted as a distance vs. position. The experimental configuration is illustrated in the photograph in Figure 2, and the MATLAB surface plot of the imaged object (in this case a rubber doorstop) is shown in Figure 3.

Lab 7. Subsurface Ultrasonic Imaging
In Lab 7, the students are provided with an opaque gelatin package with a metal shape inside it. They are to submerge the package into the aquarium tank and image it with the ultrasonic transducer as in Lab 6. In this case, by careful thresholding or time-gating, the students can image the subsurface metal rather than the front surface of the package.

5. Assessment of HTT&TL Experience

A full assessment of the student experience in the HTT&TL pilot sections of GE1102 will be presented later. It should be noted that several features of the course make an objective comparison of student learning between the HTT&TL experience and the normal GE1102 sections difficult. While the goals of the HTT&TL and regular sections of GE1102 are similar, the experiences are so different that any objective test of skills will inevitably be biased toward one or the other experience. In addition there are other uncontrolled variables that may make the
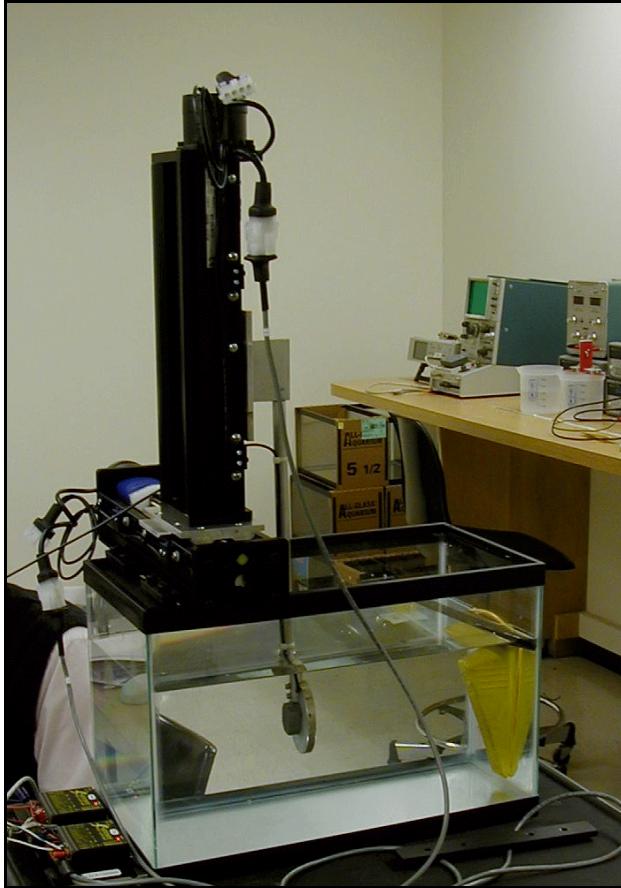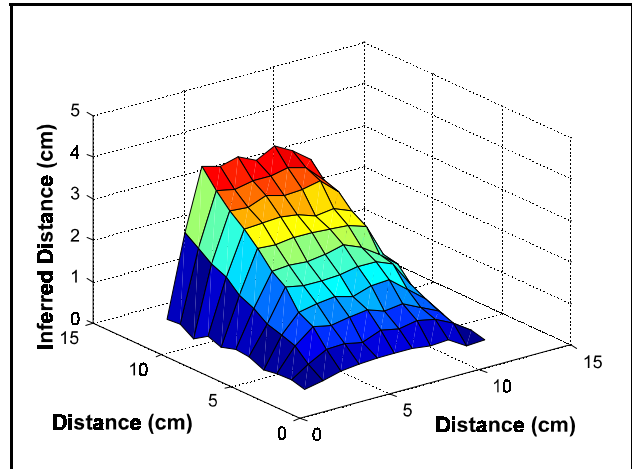
Figure 3. 10cm x 10cm surface plot of test object (doorstop) created by MATLAB control and imaging program from experimental configuration of Figure 2.

Figure 2. Ultrasonic tank with transducer mounted on x-y positioner. The tank is set up to image the doorstop on the tank wall at right.

comparison difficult. The students were selected for the pilot HTT&TL sections by asking for volunteers, rather than by random assignment, and as a result they may tend to be more "bought in" to the course than would otherwise be the case. On the other hand, these pilot sections were the initial run of the course and the course should improve when the bugs are worked out. In addition, this course represented the first teaching experience with freshmen for the classroom instructor (GT). Again, we would expect that the instruction experience should improve in the next run of the course.

We have had the benefit of qualitative comments from two groups of students. Student volunteers from the CenSSIS Student Leadership Council (SLC) helped us to debug the labs before the course was offered. These included undergraduate students who had taken GE1102 in the standard configuration and one graduate student who had served as a TA for a standard GE1102 class. The reaction from these SLC students to the lab was uniformly positive. Qualitative comments from undergraduates in the pilot sections was also very positive.

Quantitative data is being collected now and will be reported later. We have decided to use as quantitative assessment instruments student perceptions of learning, both at the time of the course and during the following quarter when the students will be taking the C-programming course. Students will be asked how much they learned, how they liked the course, and, during the following course, how they felt that GE1102 prepared them for their C-programming experience. A control group of students who took a standard GE1102 course will be given the same surveys and results will be compared.

6. Conclusion

We have reported on the use of a "High-Tech Tools and Toys" lab to teach computation and programming skills through data acquisition and measurement controlled by the new MATLAB Data Acquisition and Instrument Control Toolboxes. Pilot sections of a two-quarter-hour freshman problem-solving with software applications course have been presented in this format at Northeastern with good student and instructor feedback. A similar module at Boston University is planned for the fall semester of 2001. Real-time control and measurement can provide a format to teach programming skills that is captivating for students of all levels of computer skills, in addition to teaching other valuable engineering skills.

7. Acknowledgements

Bibliography
1. Whitehead, A. N., *The Aims of Education and Other Essays*, (Macmillan, NY) 1959.

STEPHEN W. MCKNIGHT
Stephen McKnight received a BA in Physics from Oberlin College in 1969 and a Ph. D. in Solid State Physics from the University of Maryland in 1977. He is an Associate Professor of Electrical and Computer Engineering and chairs the department Undergraduate Study Committee. He is also the Education Thrust Leader for the recently awarded National Science Foundation Engineering Research Center on Subsurface Sensing and Imaging Systems (CenSSIS).

WILLIAM E. COLE
Dr. Cole received his Bachelors of Mechanical Engineering Degree from Stevens Institute of Technology and his Doctor of Philosophy from the Pennsylvania State University. He has over twenty years of industrial experience developing industrial process equipment at The United Technologies Research Center and Thermo Electron Corp. Dr. Cole is now using this industrial experience to help educate the next generation of engineers.

GILEAD TADMOR
Gilead Tadmor received his B.Sc., 1977, from Tel Aviv University, and his M.Sc., 1979, and Ph.D., 1984, from the Weizmann Institute, Israel, all in mathematics. He joined the Department of Electrical & Computer Engineering at

Northeastern University in 1989, and is active in research & industrial consulting in control systems, with applications to power engineering, fluid dynamics and electro optics. Dr. Tadmor is a Senior Member of IEEE and currently serves on the Control System Society Conference Editorial Board.

E. CARR EVERBACH
E. Carr Everbach is currently an Associate Professor of Engineering at Swarthmore College. He received his B.S. in Applied Mechanics/Acoustics from Harvard College in 1982 and M.S. and Ph.D. in Mechanical Engineering from Yale University in 1986 and 1989. Carr chairs the Biomedical Ultrasound/ Bioresponse to Vibration technical committee of the ASA, the Engineering Division of the Council on Undergraduate Research, and teaches mechanical engineering, environmental studies, women's studies, and other courses at Swarthmore College (PA).

MICHAEL RUANE
Michael Ruane is Associate Professor of Electrical & Computer Engineering at Boston University. He received the B.E.E. from Villanova University in 1969, his S.M.E.E. from MIT in 1973, and the Ph.D. in Systems Engineering from MIT in 1980. He was a Peace Corps volunteer in Sierra Leone, and a staff member of the MIT Energy Laboratory and is a registered professional engineer (electrical). He joined Boston University in 1980 and is a member of the Boston University Photonics Center.