Signal Processing Experiments with the LEGO MINDSTORMS NXT Kit for Use in Signals and Systems Courses

Bonnie Heck Ferri, Safayet Ahmed, Jennifer E. Michaels, Eric Dean, Chris Garyet, Sam Shearman

Abstract–**This paper presents a set of inexpensive signal processing experiments that can be used as projects or hands-on demos to supplement signals and systems courses. Signals and systems concepts tend to be very mathematical and abstract, and students who prefer more practical material are at a disadvantage in these courses. The experiments are performed on the LEGO MINDSTORMS NXT platform, which is portable, relatively inexpensive, and rugged enough so that students can perform the experiments at home or in the classroom. Experiments introduced in this paper highlight fundamental concepts such as sampling, aliasing, digital filtering, frequency analysis, system identification, and control design. Projects, programs, and additional hints for instructors are posted on a website for free use.**

I. INTRODUCTION

Material in most signals and systems courses can be too abstract for many students. Experiments can reinforce and clarify the concepts, but most experiments are relegated to lab courses that are separate from the theoretical or lecture-based courses. It has been shown in other applications that cheap, portable experiments that can be done in class or at home by students can enhance the learning in lecture-based courses [1-3].

The LEGO MINDSTORMS System has been used successfully for many college-level experiments because it is easy to use, versatile due to the availability of several sensors and a large variety of LEGO building pieces, and rugged enough for use by 12 year olds. Numerous educational experiments have been developed for the initial model, the LEGO MINDSTORMS kit with the RCX Brick. Most of the experiments reported in the literature are in the areas of controls, robotics, and computer science; see, for example, [4-11], which are not in the area of signal processing.

The new version of the LEGO MINDSTORMS, the NXT, was introduced in July 2006. The LEGO MINDSTORMS NXT kit has the following features: 32-bit ARM7 microcontroller; 256 Kbytes FLASH; 64 Kbytes RAM; Bluetooth wireless communication; 4 input ports and 3 output ports; touch, light, encoder, ultrasonic, and sound sensors; and DC motors. The sensors of the NXT kit have much higher resolution than those of the RCX kit and the processor is faster, using 32-bit operations as opposed to 8-bit operations. The improvements over the RCX model have inspired many new projects for the NXT, mostly for the hobbyist market [12-16]. These improvements also make the NXT well-suited for educational signal processing experiments.

The NXT processor can be programmed using a variety of options such as C and graphical programming. Two C-based software development packages are ROBOTC [18] and BricxCC, which uses Not eXactly C (NXC) [19]. ROBOTC was developed at Carnegie Mellon University, and NXC is a successor of nqc, a programming language used extensively for the LEGO RCX model. Both ROBOTC and NXC were tested during the development of these experiments, and both yielded comparable results with respect to accuracy and maximum sampling rate achievable. If students are to take the experimental platforms home to do the experiment on their own computers, then NXC is preferred due to the fact that it is free and works on a larger selection of operating systems (Windows, MAC, OSX, LINUX).

Graphical programming alternatives include the LabVIEW NXT Tookit, NXT-G, Robolab, and the Microsoft Robotics Studio (MSRS) [12,17,21]. The LabVIEW NXT Toolkit is especially interesting for signal processing labs because, as an add-on for the general-purpose LabVIEW graphical development environment, it allows users to work with the NXT hardware in several ways. For one, users can build graphical programs that interact with NXT hardware that remains connected to the PC through a USB cable. This tethered configuration enables interactive control and transfer of sensor data to the PC for immediate analysis, processing, and visualization. Users can also build graphical programs that execute autonomously on the NXT hardware allowing sensor data to be uploaded to the PC for offline processing and analysis.

The logistics of using LEGO kits for an experiment in a lecture-based course has been discussed in [8]. That experiment consisted of a feedback control module based on the LEGO RCX kit programmed with NQC. It was found that 3 to 4 kits could suffice for a class of 50 students when used on a time-shared basis and assuming 3-person groups. Students were able to download NQC for free and write the code on their own computers. To ease the programming burden for students unfamiliar with C, a skeletal program was provided so that they could concentrate on the algorithms rather than on the low level details such as storing data. Groups would borrow a kit for overnight or for

B. Ferri, S. Ahmed, J. Michaels are with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA; E. Dean, C. Garyet, S. Shearman are with National Instruments, Austin, TX, 78759. Corresponding author email is bonnie@ece.gatech.edu.

This work was supported by NSF CCLI Program, Russell Pimmel Project Director, project number 0618645.

half-day periods of time to complete their work at home. The units were small enough and robust enough that an experimental setup was sent to a remote campus location for students to complete the experiments via a distance learning environment. The success of the LEGO RCX control experiment with the NQC programming language inspired the current use of the LEGO NXT with the NXC programming language for signal processing experiments. In addition, work is underway to implement exercises using the LabVIEW NXT toolkit.

II. EXPERIMENTAL MODULES

The experiments are summarized in Table 1. These experiments have the following features:

- Cost of $250 for LEGO NXT kit
- Sensors: touch, light, ultrasonic, sound (1 each)
- Students work in 2 to 3 person groups, recording data to be uploaded to a PC
- Mode of delivery: done in class, in a lab, or at home
- Skeletal code provided to students

The additional specific features for the different experiments are listed below along with the fundamental concepts demonstrated by the experiment and the incremental cost over that of purchasing the basic kit. Since the resolution on the smallest timer in NXC is 1ms, a base sampling period of 6 ms is typically utilized in these modules to maintain a relatively constant sampling period.

Table 1

Experiment	**Concepts**
Light sensor	Aliasing, Frequency Analysis, Digital Filtering
Shaded Disk Experiment	Periodic Signals, Chirp Signals, Aliasing, Highpass and Lowpass Filters
Sound Sensor	Frequency Analysis, Modulation, System Identification
Motor Control Demo	Effect of Feedback Control, Root Locus
Motor Control Project	System Identification, Root Locus, Control Design and Implementation

Supporting documents for these experiments are available on the website:

users.ece.gatech.edu/~bonnie/book3/LEGO_Experiments.html.

The website includes projects, code, and notes for instructors on best practices for implementing the projects and frequently asked student questions.

A. Light Sensor Experiment

Consider an experiment where one light sensor is put into a passive mode with the raw sensor data recorded. The features specific to this experiment include

- Hardware: LEGO NXT processor, 1 or 2 LEGO light sensors, miscellaneous LEGO parts for mounting the sensors together
- Fundamental Concepts: aliasing, noise characterization, covariance, frequency analysis, digital filtering
- Incremental Cost: $17 for extra light sensor (optional)

This experiment can be done in class by the instructor or done at home by the students. The light sensor data is taken while the lights in the room are turned on and off repeatedly. The light intensity recorded by the sensor correspondingly goes up and down as shown in the sample waveform of Figure 1.

While the noise appears random in Figure 1, an FFT performed on the data file indicates that there is frequency component at 47 Hz. Students are asked to determine the cause of the noise. Note that the 47 Hz component is found when the experiment is done under fluorescent lights (which are very common since students usually do the experiment at school when they are working with partners). Fluorescent lights flicker at twice the line frequency, 120 Hz. With a sampling period of 0.006 ms, the sampling frequency is 166.6 Hz, causing aliasing of the 120 Hz to appear as a frequency component of 47 Hz. Most students were initially stumped by this phenomenon, but it is a good example of aliasing when measuring real time signals.

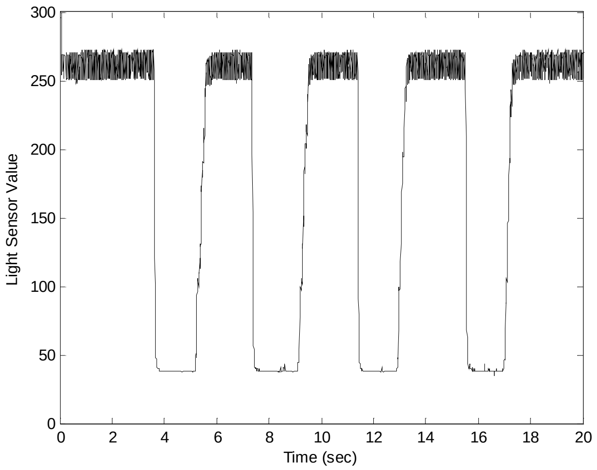

Figure 1: Results of light sensor to the light in the room being turned on and off repeatedly.

Two additional experiments are done with this set-up. The first is to repeat the experiment with two side-by-side light sensors. The sensor readings have slight but noticeable differences in offsets and gains. Students calibrate the sensors and perform covariance analysis on the results.

The other experiment requires the students to determine the frequencies of the low frequency signal and design an appropriate lowpass digital filter to remove the high frequency component while passing the low frequency component with negligible distortion. Digital filters can be implemented directly on the NXT platform. Implementing the difference equations directly in C is obviously different than running a simulation in MATLAB, and is a basic skill that many students lack. Filtering is a case where the lack of floating operations becomes apparent. Many embedded processors do not have floating point functional units to save cost, space, and energy; hence, experience with implementing difference equations using integer math, whether for digital filters or digital controllers, is a useful exercise. Students need to scale their coefficients, say by 1000, before implementing the filter. They must also be careful of the order of operations to avoid excessive truncation or overflow errors. In this experiment, students generally need several iterations to find a good bandwidth that satisfies these constraints.

B. Shaded Disk Experiment

Consider the experimental setup shown in Figure 2. A disk with graduated black and white sectors is mounted on a LEGO motor, and a light sensor is positioned near the surface of the disk. As the disk turns, the sensor measures a signal that appears sinusoidal as seen in Figure 3. Since a constant spacing between the disk and the sensor cannot be maintained, the sinusoidal signal contains lower amplitude subharmonics if more than one black and white sector is used. The more severe the wobble, the more prominent the subharmonics. Having control over the motor speed and the sampling period allows students to experiment with aliasing.

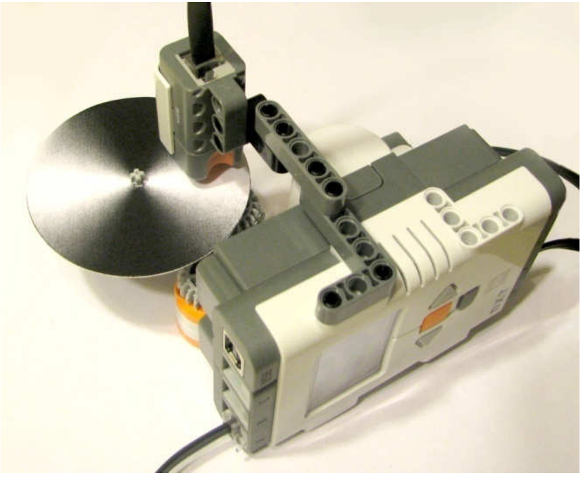

Figure 2: Experimental platform consisting of a motor spinning a shaded disk and a light sensor to measure the intensity off the disk. Related projects measure sinusoids, analyze aliasing, and implement digital filters.

Furthermore, a sinusoidal signal with subharmonics provides ample opportunity for designing and implementing digital filters.

- Hardware: LEGO NXT processor, light sensor, black and white graduated disk, computing platform is PC or LEGO NXT processor
- Fundamental Concepts: sampling, aliasing, filtering
- Incremental Costs: none

A chirp signal can be generated by increasing the speed of the motor, with the resulting waveform shown in Figure 4 (a) for a sampling period of T = 6ms. This particular waveform was obtained from a disk that had one black and one white sector. Reducing the sampling rate results in aliasing as shown in Figure 4 (b).

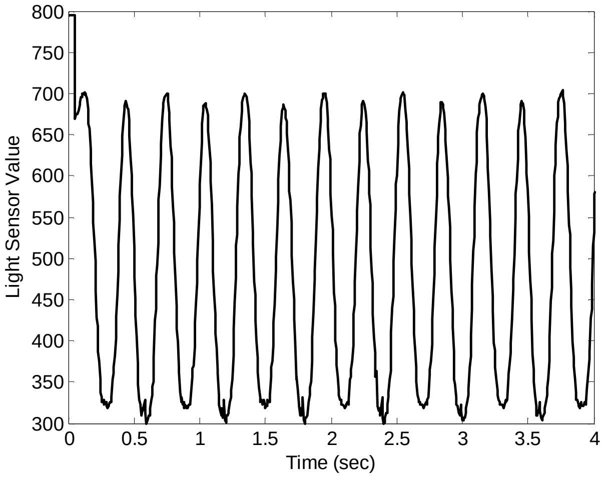

Figure 3: Sinusoid measured from the shaded disk experiment shown in Figure 2.

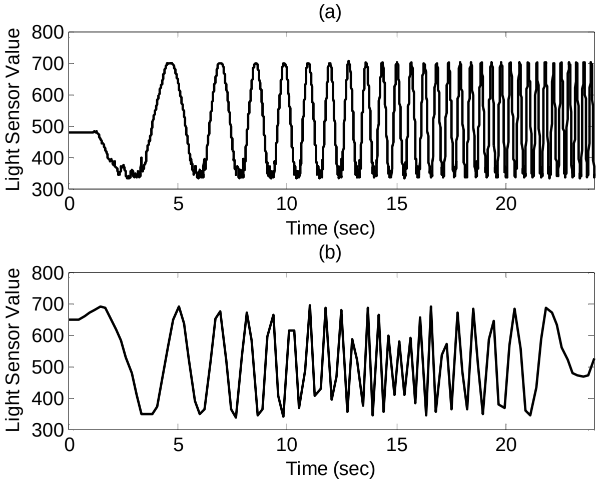

Figure 4: Chirp signal recorded with (a) sampling rate of 6 ms and (b) 240 ms, showing aliasing.

Figure 5 shows the result of the chirp signal sent through a 50th order FIR lowpass filter implemented on the NXT. For the sake of comparison, the unfiltered data shown in Figure 5(a) was imported into MATLAB and the same digital filter implemented. The resulting filtered data generated by MATLAB was virtually identical to that shown in Figure 5(b). As part of the experiment, students are asked to use MATLAB to double-check the accuracy of their implementation. Other types of digital filters can be used as well.

A modification to this experiment can be done by adding an additional light sensor in a quadrature configuration to build an optical encoder. Details of such an experiment are given on the website.

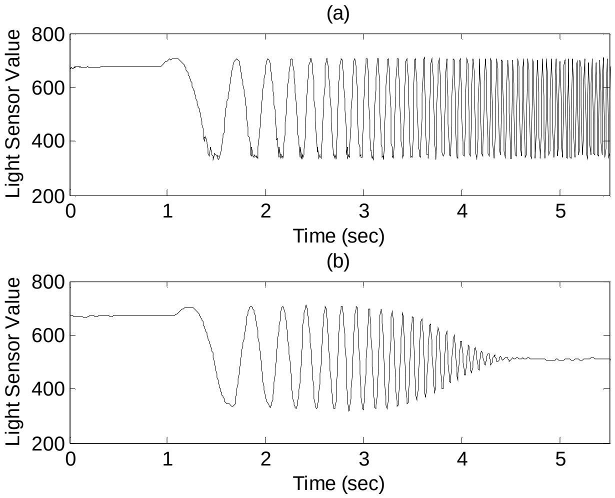

Figure 5: (a) Measured data of a chirp signal and (b) output of a low-pass filter of the chirp signal, implemented on the NXT processor.

C. Sound Sensor Experiment

A sound sensor can be used to measure ambient noise, sound of speech or clapping, and computer-generated sound.

- Hardware: LEGO NXT processor, 1 sound sensor, computer
- Fundamental Concepts: frequency response, system identification, spectrum, modulation
- Incremental Costs: none

A chirp signal can be generated in MATLAB and modulated as shown in Figure 6(a). This signal is then broadcast over the PC speakers. The LEGO sound sensor picks up the envelope of the sound signal (i.e., the original chirp signal) as shown in Figure 6(b). Note that the signals shown in the figures are not synchronized. The roll-off of the sound sensor data shows that the sound signal has gone through a system with low-pass characteristics. The sound system includes the speakers, the air channel, and the microphone. Figure 7 shows the Fourier Transforms of the original chirp signal and the signal recorded from the sound sensor, where the Fast Fourier Transform was used to approximate the Fourier Transform as described in [22]. This experiment can be used to characterize the frequency response of the sound system by the students taking the ratio of the Fourier Transforms of the signals.

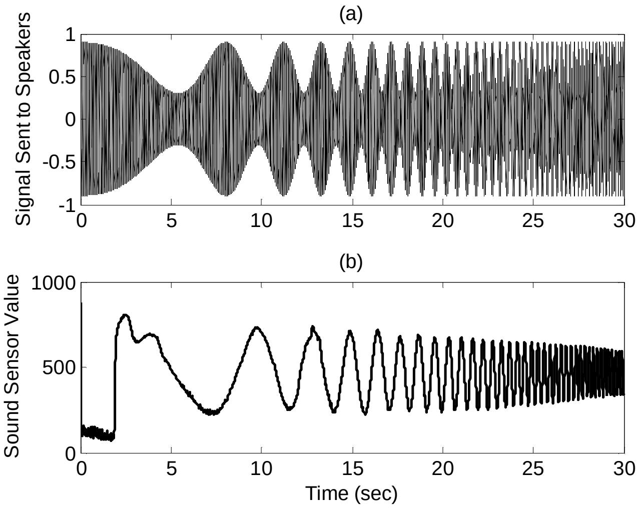

Figure 6: (a) modulated chirp signal generated using a PC and broadcast through its speakers and (b) signal recorded from the LEGO sound sensor.

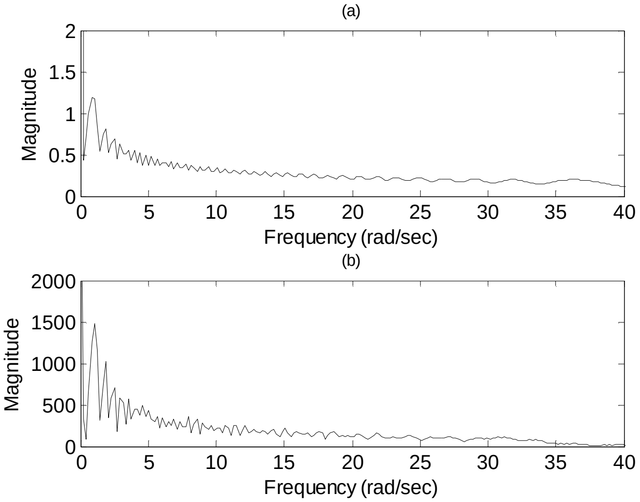

Figure 7: Fourier Transforms of the (a) original chirp signal and (b) the signal recorded from the LEGO sound sensor.

D. Feedback Control Demo

An updated version of the simple motor control experiment discussed in [8] can be used as a demonstration of the effect of feedback control. The experiment requires only a motor and the NXT processor.

- Hardware: LEGO NXT processor, motor
- Fundamental Concepts: feedback control, effect of gain on steady-state error and disturbance rejection, root locus, time domain performance criteria
- Incremental Costs: none

The motor assembly includes an optical encoder to measure the output shaft angle. The demo, consisting of the NXT processor and an attached motor, is built to be passed around during class time. Each student runs a program on the NXT processor that implements a proportional position controller with a step reference input. The program also allows students to input different gains to the controller and view the resulting responses. A sample step response is shown in Figure 8.

This experiment is particularly useful when first introducing the concept of feedback control as well as being used in conjunction with the study of root locus. The effect of control is felt by students when they try to turn the motor angle away from its set point. The higher the gain, the harder it is to move the motor from the set point. The system is modeled with two poles, with one being at the origin. Students can correlate the responses seen with different gains (underdamped, overdamped) to the root locus plot. Additional practical considerations are present in the experiment, such as saturation of the control and a dead zone about zero in the motor control signal.

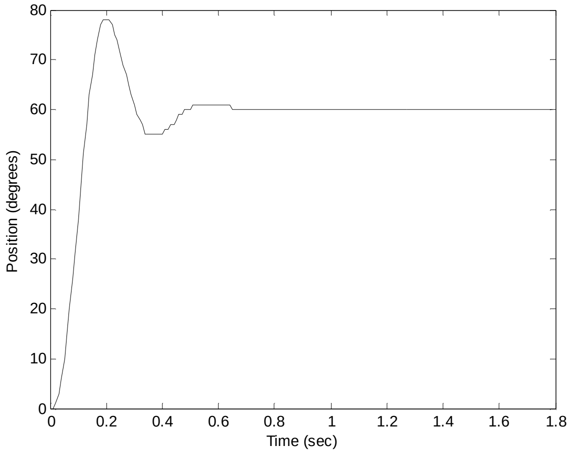

Figure 8: Proportional control of a LEGO motor with a gain of 3 and a set point of 60°.

E. Feedback Control Project

The project for a motor control experiment requires students to perform system identification to find a model of the system, design either a suitable PID control or lead compensator to satisfy design criteria, implement the control, and measure the results resulting from step and sinusoidal inputs.

- Hardware: LEGO NXT processor, motor
- Fundamental Concepts: feedback control design, model identification, robustness, time domain and frequency domain performance criteria, sinusoidal tracking
- Incremental Costs: none

A cart is built from the motor and processor as shown in Figure 9. This setup is a little more elaborate than needed for the demo described in Section II.D for two reasons: to test robustness and to make it more interesting for students. The robustness is tested by holding the cart in the air and running the experiment versus running the experiment with the cart rolling on the ground. The added resistance to the wheels does make a difference in the response.

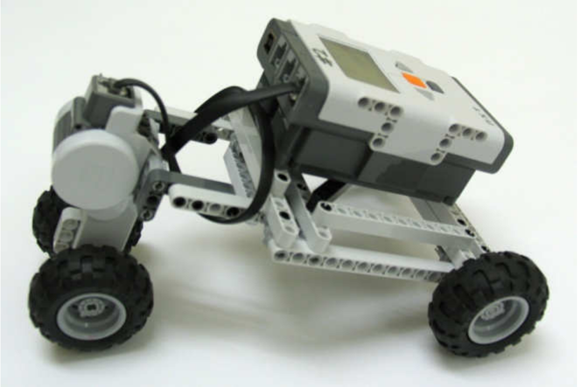

Figure 9: Cart experiment for motor control.

The components of the project are 1) model identification, 2) control design to meet time domain specifications on settling time, rise time, and/or percent overshoot, 3) discretizing the controller and implementing it on the cart, 4) the iteration on the design, 6) identifying the controller bandwidth and finding the experimental response of the system to a sinusoidal input at half the bandwidth and at double the bandwidth, and 7) determining a strategy to overcome a nonlinear phenomena, that of the dead zone.

The model identification is done by identifying the parameters in a simple DC motor model:

$$G(s) = \frac{A}{(s+b)s}$$

These parameters can be determined from the closed loop step response with proportional control and a known gain [8]. The step response for the cart resembles that of the motor alone, shown in Figure 8.

Students are asked to find the closed loop frequency response of their model with the designed control and to predict the magnitude and phase lag of the response to a sinusoidal tracking signal. Then they are asked to implement the sinusoidal tracking in the actual system and compare the responses to those predicted from the frequency response plot. The responses of the actual system to two sine waves are shown in Figure 10.

The project can either be broken into several small pieces that are done in class in small groups, or it can be given as a two-week take home project. The benefit of doing the project in class is that the pieces of the project can be distributed throughout the semester and performed in conjunction with the material being covered in the course. This mode also resembles that if a traditional lab course where a lab instructor is present to give a short overview and is available during the class time to answer questions and render aid. However, this mode takes more class time. A take-home project puts all the individual pieces together at one time so that students receive a cohesive experience in designing and implementing a control system.

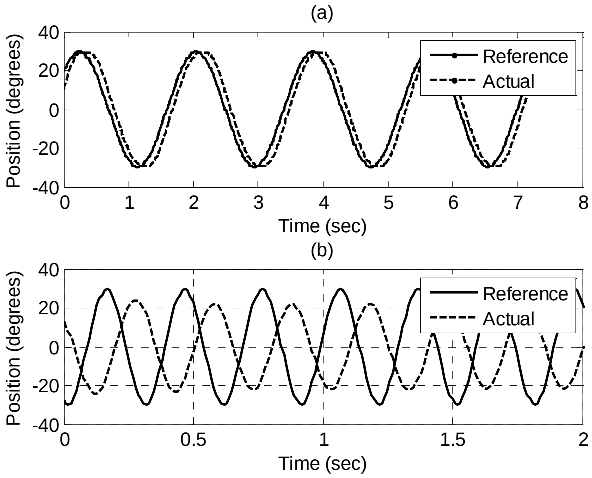

Figure 10: Steady-state sinusoidal tracking results of the cart experiment showing the reference signal and the corresponding actual response of the cart for (a) a sinusoidal input with frequency within the controller bandwidth and (b) a sinusoidal input with frequency above the controller bandwidth.

III. SUMMARY

The experiments described in this paper are relatively inexpensive, rugged, and portable enough to be used in a classroom or taken home by students. The experiments are also sophisticated enough to demonstrate several fundamental concepts taught in standard signals and systems courses. The concepts taught in these experiments include sampling, aliasing, digital filtering, measurement noise. frequency analysis, system identification, control design, and quadrature computations. Most students who take signals and systems courses would find the material very abstract and mathematical without experiments such as the ones described here. Experience with these experiments at Georgia Tech, both in electrical and mechanical engineering courses, has been very positive. The logistics and practicality of using 3 to 4 kits for a class of 50 students are reasonable. The signal processing experiments are part of a larger program to develop many portable and inexpensive experiments and distribute them throughout the ECE curriculum to incorporate "hands-on" experience into a wide selection of lecture-based courses [20].

ACKNOWLEDGMENT

The authors appreciate the suggestions of Jennifer Garcia, Joel Jackson, and Doug Williams in designing the experiments.

REFERENCES

[1] Dua, R., Seiffertt, J.E., Blaha, B., Gupta, K., Satagopan, V., Stanley, J.R., Beetner, D., and Wunsch, D.C., 2005, "Hands-On Projects and Exercises to Strengthen Understanding of Basic Computer Engineering Concepts," *Proceedings of the American Society of Engineering Education Annual Conference & Exposition*, Portland.

[2] Durfee, W., Li, P., and Waletzko, D., 2005, "At-Home System and Controls Laboratories," *Proceedings of the American Society of Engineering Education Annual Conference & Exposition*, Portland.

[3] Litwhiler, D.H., Lovell, T.D., 2005, "Acoustic Measurements Using Common Computer Accessories: Do Try This at Home," *Proceedings of the American Society of Engineering Education Annual Conference & Exposition*, Portland.

[4] Baum, D., Gasperi, M., Hempel, R., and Villa, L., 2000, Extreme MINDSTORMS: An Advanced Guide to LEGO MINDSTORMS, Apress.

[5] Erwin, B. Cyr, M.; Rogers, C., 2000, "LEGO engineer and RoboLab: teaching engineering with LabVIEW from kindergarten to graduate school," *Int. Journal of Engineering Education*, v 16, n 3, 2000, pp. 181-192.

[6] Gawthrop, P.J., and McGookin, E., 2004, "A LEGO-Based Control Experiment," *IEEE Control Systems Magazine*, Oct. 2004, pp. 43-56.

[7] Hatfield, J.M., and Tester, J.T., "LEGO Plus," *Proceedings of the American Society of Engineering Education Annual Conference & Exposition*, Portland.

[8] Heck, B.S, N. Scott Clements, and Ferri, A.A., 2004, "A LEGO Experiment for Embedded Control System Design," *IEEE Control Systems Magazine,* Oct. 2004, pp. 61-64.

[9] Klassner, F. and Anderson, S., 2003, "Lego Mindstorms: Not just for K-12 anymore," *IEEE Robotics & Automation Magazine*, v. 10, issue 2, June 2003, p. 12-18.

[10] Wang, E., 2004, Engineering with LEGO Bricks and RoboLab, Paperback College House Enterprises.

[11] Williams, A.B., 2003, "The qualitative impact of using LEGO MINDSTORMS robots to teach computer engineering," *IEEE Transactions on Education,* v 46, n 1, Feb. 2003, p206.

[12] Kim, S.H. and Jeon, J.W., "Programming LEGO mindstorms NXT with visual programming," International Conference on Control, Automation and Systems, October 2007, pp. 2468-2472.

[13] Bagnall, B. Maximum LEGO® NXT: Building Robots with Java Brains Variant Press. 2007.

[14] Sharad, S., "Introducing Embedded Design Concepts to Freshmen and Sophomore Engineering Students with LEGO MINDSTORMS NXT," IEEE International Conference on Microelectronic Systems Education, 2007, 3-4 June 2007 pp. 119 – 120.

[15] M. Gasperi, P. E. Hurbain, I.L. Hurbain, Extreme NXT: Extending the LEGO MINDSTORMS NXT to the Next Level, Springer-Verlag, NY. 2007.

[16] M. Ferrari, G. Ferrari, and D. Astolfo, Building Robots with LEGO Mindstorms NXT, Elsevier Publishing, MA, 2007.

[17] D. Swan, "Programming Solutions for the LEGO Mindstorms NXT," Robots Magazine, online articles, 10-31-07, downloaded 3-12-08. http://www.botmag.com/articles/10-31-07_NXT.shtml

[18] RobotC homepage, http://www.robotc.net/

[19] BricxCC Command Center 3.3, SourceForge.net http://bricxcc.sourceforge.net/.

[20] Ferri, B., Auerbach, J., Jackson, J., Michaels, J.E. and Williams, D. "A Program for Distributed Laboratories in the ECE Curriculum," ASEE Annual Conference and Exposition, Pittsburgh, June 2008.

[21] M. Gasperi, LabVIEW for LEGO MINDSTORMS NXT, http://www.ntspress.com/publications/labview-for-lego-mindstorms-nxt/ , 2008.

[22] E. Kamen and B. Heck, Fundamentals of Signals and Systems Using The Web and MATLAB, 3rd Ed., Prentice Hall, page 200-203.