Developing a New Affordable DC Motor Laboratory Kit for an Existing Undergraduate Controls Course

Rebecca M. Reck and Ramavarapu S. Sreenivas

Abstract— **The aim of this research is to expand the access of hands-on controls education at the undergraduate level with an affordable laboratory kit. A kit was assembled for around $130 while replicating the educational functionality of a typical station in an introductory controls laboratory. The kit consists of a Raspberry Pi, a DC motor, and various circuits required for the lab exercises. Details of each of these components are explained. Additionally, the laboratory experiments tested with this kit are described. This kit replaces expensive equipment with an affordable alternative that can be easily shipped anywhere in the world and used by students who do not have a formal laboratory space to use. The kit greatly enhances the accessibility of the laboratory experience to students in budget-strapped campus laboratories and those participating in distance education.**

I. INTRODUCTION

This research developed a modular, portable, and affordable laboratory kit to support the accompanying curriculum for the introductory controls course in the General Engineering program at the University of Illinois at Urbana-Champaign. The objective was to design each kit to be assembled for around $100 while replicating the educational functionality of a lab bench in a university controls laboratory. A kit will also allow older analog computers to be updated with newer technology that is more representative of what is currently used in industry [1]. By replacing expensive equipment with an affordable kit that can be shipped anywhere in the world increases the accessibility of the controls laboratory experience for students on campus and remote locations. Previous research shows that hands-on laboratory experiments help students understand and apply course material [2].

Some affordable and transportable laboratory devices for engineering education have already been developed, such as the Mobile Studio IOBoard, which is centered on a custom-built board that replicates the functionality of an oscilloscope, function generator, multimeter, and power supplies and is primarily used in introductory circuits courses [3].

The target course for the kit is GE320 (an introduction to control systems for general engineering students). This course is representative of the first course in controls for many electrical, mechanical, and aerospace engineering programs. The kit design consists of a Raspberry Pi (a fully functional ARM-based computer that is the size of a deck of cards), a DC motor, and the various circuits required to drive the motor, to measure position and speed, and perform system identification.

II. BACKGROUND

The need for laboratory experiences in control systems courses has been well established in [4], [5], [6], [7] and others, however there are challenges associated with including them. Some hurdles include: budget constraints, space limitations, class size, and limited teaching resources [5], [8], [9], [10]. Additionally, the increasing popularity of online courses has added a new consideration for laboratories [4], [11], [12].

The literature shows that the cost of equipment per station varies in cost from $180 [13] to $32,493.74 [14]. This research looks to replace the basic functionality of these laboratories with an affordable kit. The target budget of $100 for the kit was used because it is the approximate cost of a textbook, only three times the cost of an iClicker, which is another common piece of technology that students purchase for courses, and the approximate cost of other low-cost kits for other courses found in literature [3], [15]. The Arduino prototyping kit described in [15] is approximately $95 and was designed for a multidisciplinary course on perception, light, and semiconductors. The Mobile Studio IOBoard described in [3] has multiple versions ranging in price from $80–$130. The primary application of the Mobile Studio IOBoard is undergraduate circuits courses.

In addition to monetary cost, dedicated laboratory space is also limited and class sizes are increasing. These factors place restrictions on the capabilities of face-to-face labs. Additionally, not all students can attend and complete these laboratory experiments due to time, location, or physical disability [16]. An alternative to face-to-face laboratory experiences are laboratory kits.

A lab kit allows students to take home the laboratory equipment to complete the experiments on their own time [15], [13]. These kits started to become more popular as the cost of the required hardware has decreased [15]. The kits contents vary based on the objectives of the course and can be assembled by the instructor [15], [8], adapted from an existing kit [13] or purchased as a complete kit like Lego Mindstorms NXT [17], [18]. These kits have been well received by students [8], [10], [15].

The science and engineering active learning (SEAL) system created a take-home kit for students to develop a cart with an inverted pendulum attachment [8]. It was designed to be used in controls courses. The cost of the kit is

*This work is supported by the Academy for Excellence in Engineering Education at the University of Illinois at Urbana-Champaign

Rebecca M. Reck and Ramavarapu S. Sreenivas are with the Industrial and Enterprise Systems Engineering Department, University of Illinois at Urbana-Champaign, 117 S. Mathews, Urbana, Illinois 61801, USA reck2@illinois.edu; rsree@illinois.edu

approximately $100 plus $179 for a myDAQ from National Instruments [8], [19]. The MESAbox was also designed for controls and mechatronics courses, it uses an Arduino and costs approximately $180 [13]. The MESABox kit includes multiple motors and sensors and is based on an off-the-shelf kit from Sparkfun, however this includes more components than required for the targeted course. The laboratory experiments designed for the MESABox cover a variety of controls topics including using the Arduino programming language and wiring all of the circuits.

The DC Motor control equipment detailed in [6] includes $80 of hardware and a motor, gearbox, and encoder. The cost of the latter three components are not included, however the motor manufacturer's website indicates these components are more than $100 each [20]. The total cost for each station with this equipment is approximately $400 and it is not designed to be portable.

All undergraduate laboratory experiences still need to meet the course goals and objectives as well as ABET accreditation requirements [7], [12]. There are several goals that can be applied to laboratory experiences based on the outcomes in the ABET Criterion; a student should have the ability to conduct experiments, analyze and interpret data, use modern engineering tools, design experiments, solve engineering problems and function in teams [21]. Specific goals for controls labs based on these goals have followed. In general, the controls laboratory experience should prepare students for a career in control systems [1] by performing the following steps: building the system [5], modeling and analyzing the system, developing a controller to meet performance requirements, simulating the controller and system, observing the physical system, collecting the data, and using the data to improve the system model or control tuning [4], [10]. Experiments based on DC motors [6], [22] have been identified to meet these goals for controls laboratory experiences. An advantage to using a DC motor for a control system experiment is proportional-integral-derivative (PID) control of the motors position is a popular example [22]. Additionally, a DC motor setup can be expanded to create more complex setups like the inverted pendulum [6].

III. MOTIVATION

There were four primary considerations driving the development of this kit: achieve the same educational objectives as the current laboratory equipment, cost and accessibility of parts, portability of the complete kit, and student interface. The first consideration was to have a seamless transition in the laboratory without changing the lecture part of the course. It also enables a quasi-experiment in the fall semester of 2014 to compare the effectiveness of each type of equipment.

A budget of approximately $100 and the desire to be able to quickly obtain replacement parts if something breaks are what drove the consideration of cost and availability of parts. All of the parts in the kit are available at major online retailers or 3D printed. Cost and accessibility is also closely tied with the consideration of portability. Portability is a long-term goal of the project, so that the students can take the kits home or the kits can be shipped to students taking online courses.

The last consideration, student interface, placed the most restrictions on the current design of the kit. In the lecture portion of the course and some of the existing laboratory experiments use MATLAB and Simulink as a simulation and development platform. Therefore the new kit also needed to use MATLAB and Simulink as well. At the start of the development of the kit only two small, low-cost, hardware platforms had Simulink support: Arduino and Raspberry Pi. The latter was chosen for its flexibility and potential to expand into other controls courses with more complex algorithms and possibly tracking via video.

IV. LABORATORY KIT DEVELOPMENT

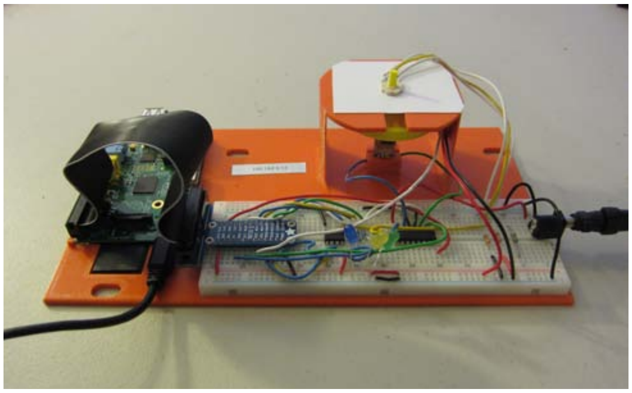

Fig. 1. Complete Kit

From the considerations in the previous section, a goal was set to develop a kit that would replicate the educational objectives of the existing lab for around $100. Off-the-shelf parts were selected when possible to make replacement parts easy to obtain. Parts that could not be purchased were 3D printed. The simplicity of the design and availability of the parts will make future expansions with the kit feasible. Fig. 1 shows the assembled kit. Table I details the supplies needed to build a complete kit, including the supplier for each part.

A. Raspberry Pi

The Raspberry Pi model B brought a lot of functionality to the kit, however it also provided some challenges. The Raspberry Pi includes several useful drivers in the WiringPi library [23] which is included in the Raspberian Linux distribution. These libraries are very helpful in setting up built in functions. The processing power and peripheral interface provide resources for potential future functionality. There are also several predefined functions in MATLAB and Simulink that make it straight forward for students to develop the code. The cost and availability of the Raspberry Pi and accessories like the Pi Cobbler breakout board from Sparkfun are also advantages. The Raspberry Pi also has 17 general purpose input/output (GPIO) pins available for use, however all of these pins can only be used for digital signals and this kit requires an analog position to be used.

An additional limitation is that the Raspberry Pi requires the communication with another computer to be through Ethernet. Plus, it also requires a separate USB micro power source. There are other alternatives to the Raspberry Pi that do not have the limitations of Ethernet or an additional power supply, including the Arduino and BeagleBone Black. The rest of the kit could be adapted to work with these platforms as well.

B. Circuits and Sensors

The full circuit diagram is included in Fig. 3. The sections below describe the major components in the circuit diagram.

1) H-Bridge: The digital GPIO pins do not supply enough voltage or current to drive the DC motor. An H-Bridge was selected to create the interface between the Raspberry Pi and the motor. L293D H-Bridge has four channels; two of them are used in the kit so that the direction can also be controlled.

2) Speed Sensor: To measure the speed of the motor, a 3D printed encoder wheel is attached to the shaft of the DC motor. The encoder wheel has 20 evenly spaced holes. The wheel passes through a photo interrupter to create a digital signal that is connected to a GPIO pin on the Raspberry Pi. See 3D model in Fig. 2.

3) Position Sensor: To measure the position of the motor, a rotary potentiometer was used. The potentiometer measures 340 of the 360 degrees of travel. It is mounted on a table shaped surface that is part of the 3D printed base. A motor shaft extension was printed as one part with the encoder wheel, see Fig. 2. However, it was found that the accuracy of 3D printing can cause inconsistencies between the kits. For example, there is a hole in the bottom of the part that was designed to fit tightly over the shaft of the motor. In some cases the part fit and did not wobble, however others the hole was too big and would not stay attached to the motor. Being loose or wobbling caused the part to hit the photo interrupter. An aluminum hub with set screw was added to some kits for stability.

TABLE I

COMPONENTS OF THE KIT

Item	**Supplier**		**Cost**
Raspberry Pi Model B	Adafruit	$	39.95
12V DC motor	Sparkfun	$	12.95
3D printed stand	in house	$	5.00
Bread board	Adafruit	$	5.95
H bridge (L293D)	Adafruit	$	2.50
ADC (MCP3002)	Sparkfun	$	2.30
Power supply (for RPi)	Adafruit	$	9.90
Power supply (for DC Motor)	Adafruit	$	14.95
Rotary Position Sensor (3382)	DigiKey	$	2.60
Photo Interrupter	Sparkfun	$	3.45
Pi T-cobbler breakout & cable	Sparkfun	$	6.95
Wires	Adafruit	$	1.60
Resistors	ECE Store	$	0.15
LEDs	Sparkfun	$	0.59
SD Card	Amazon	$	17.09
Total		*$*	*125.93*

4) Analog-to-Digital Converter: There is not an analog input on the Raspberry Pi so an analog-to-digital converter (ADC) was selected to convert the potentiometer voltage to a digital signal. As a substitute, the Raspberry Pi has a built in serial peripheral interface (SPI) and inter-integrated circuit (I2C) interfaces built that make interfacing to ADC straight forward. The MCP3002 and MCP3008 ADC were both tested with the kit. These ADCs use the SPI interface and have the added benefit of having predefined functions included in the WiringPi library that is available for the Raspberry Pi. Ultimately, the MCP3002 was selected for cost and availability. It has two available input channels instead of eight, which is sufficient for this course.

C. Simulink

The primary software package that that is used in the course is MATLAB and Simulink. Therefore it was important to use it in the laboratory experiments as well. There were a two non-standard blocks that needed to be developed. All of the models are run in Simulink's external mode. This mode allows for the data collected by the Raspberry Pi to be viewed while the simulation is running on the Raspberry Pi [24]. The models were built and tested in MATLAB Release 2014a.

1) Reading the potentiometer voltage through ADC and SPI interface: In order to take advantage of the libraries provided with WiringPi, an S-Function builder block was used to generate the output to the ADC and read the input from the ADC. The block outputs the value from the potentiometer at a rate of 100 Hz, which is set on the Initialization tab of the S-Function Builder. In the Libraries tab the following WiringPi libraries are included:

- wiringPi.h
- wiringPi.c
- mcp3002.h
- mcp3002.c
- piHiPri.c
- wiringPiSPI.c

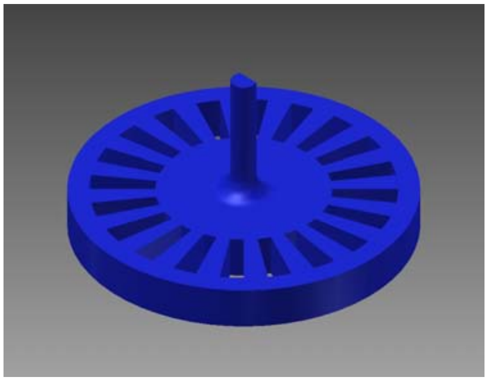

Fig. 2. Encoder Wheel and Motor Shaft Extension

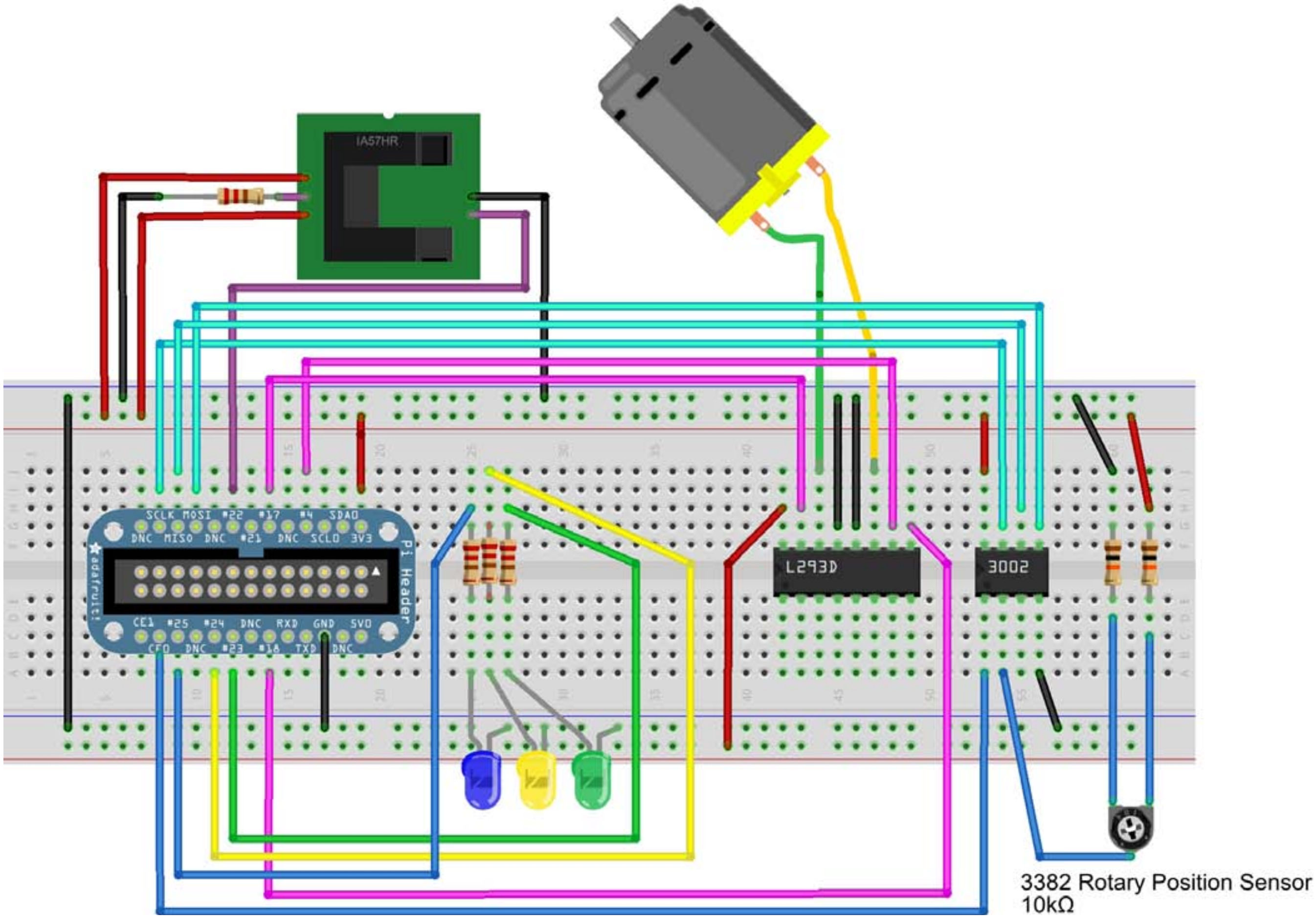
IA57HR
SCLK MOSI #22 #17 #4 SDA0
DNC MISO DNC #21 DNC SCL0 3V3
CE1 #25 #24 DNC RXD GND 5V0
CE0 DNC #23 #18 TXD DNC
adafruit
Pi Header
L293D
3002
3382 Rotary Position Sensor
10kΩ

Fig. 3. Breadboard Schematic

The following code was added in the Outputs tab:

```
if(xD[0] == 1)
{
    #ifndef MATLAB_MEX_FILE
        y0[0] = analogRead(100);
    #endif
}
```

Additionally the following code was added to the Discrete Update tab:

```
if(xD[0] != 1){
# ifndef MATLAB_MEX_FILE
  wiringPiSetup() ;
  mcp3002Setup(100, 0) ;
#endif
//done with initialization
    xD[0] = 1;
}
```

In order to use the SPI interface on the Raspberry Pi it needs to be enabled. It can be done in the MATLAB workspace by entering the following commands:

```
mypi=raspi('ipaddress','pi','raspberry')
mypi.enableSPI
```

where ipaddress in the above command refers to the actual IP address of the Raspberry Pi.

2) Encoder to Tachometer Reading: The digital signal created by the photo interrupter can be read directly by the Raspberry Pi. Once the signal is in Simulink in needs to be converted from a digital signal to a speed in radians per second. A counter block is used to count the number of pulses in a tenth of a second and then the count is converted to a speed.

3) Interface between a computer and the Raspberry Pi: Based on the type of experiment students create inputs for the motor or closed-loop control laws in Simulink. Then using Simulink's Embedded Coder the software is built and deployed on the Raspberry Pi via the network connection. When the Simulink model is run in External mode, data can be collected and viewed as the code runs via scopes or other sinks provided in Simulink. This mode requires a high volume of traffic over the network while the model is running, therefore it is recommended that a separate network be created between the computer and the Raspberry Pi. The kit was tested using separate network within the laboratory.

When setting up the Linux distribution with MATLAB, the manual network settings option was used. The manual setting allowed each Raspberry Pi to be assigned a known IP Address consistent with local network set up in the laboratory and unique for each kit. The IP address was printed on a label and affixed to the base for easy access during laboratory experiments.

There are other alternatives that can be used when the code is run as a standalone application on the Raspberry Pi. Data can be collected in a file or sent back to the host computer via User Datagram Protocol (UDP) Send and Receive blocks provided in Simulink. When the data is saved to a file on the Raspberry Pi a File Transfer Protocol (FTP) connection can be established to transfer the file to the host computer for analysis.

V. LABORATORY EXPERIMENTS

When the laboratory experiments were designed for the first semester, they were kept as close as possible to the existing experiments. This allowed the researchers to setup a quasi-experiment to test the effectiveness of both types of equipment throughout the semester. Once this test is complete, the experiments below will evolve with to align with the functionality of the kit and its portability.

The following laboratory experiments have been verified to work with the laboratory kit described in the previous section. Since this is the first laboratory in control systems, the students are led through each step of the laboratory experiments with cookbook style instructions.

A. Lab 1: Introduction to Sensors and DC Motor

The objective of the first laboratory experiment is to introduce the students to the DC motor and the equipment they will be using all semester. First the students identify the voltage and current dead zones of the motor, then they calibrate the sensors for position and speed.

The direct replication of this experiment could not be completed within the cost of the kit. A power supply that varies both current and voltage is needed to perform the dead zone experiment the same way as before.

B. Lab 2: Introduction to Simulink and the Raspberry Pi

The objective of the second laboratory experiment is to make sure students are familiar with Simulink and the Raspberry Pi. First they build three different models in Simulink to explore the basic components of a control algorithm: gain, derivative, and integral. In the first model with gains, slider gain blocks are used to illustrate changing gain values. They are also asked to explore the difference in long vs. short data types. In the derivative model, students input a triangle wave to the standard derivative block. Then in the integral model, a square wave is input into the standard integral block. In the latter two models students view the outputs via a standard scope block.

The final section of the experiment, students are asked to build a model that send three different duty cycles of square waves to specific pins of the Raspberry Pi. Each of these pins are hooked up to LEDs. When the LEDs begin to blink, the students are provided instant feedback that they have set up and run code successfully on the Raspberry Pi.

C. Lab 3: First Principles Systems Identification

The objective of third laboratory experiment is to build a system model with first principles. Students measure most of the electrical characteristics of a DC motor such as armature resistance and inductance. The only parameter that is not measured directly in the experiment is viscous friction.

Students are provided the following equations:

The voltage equation:

$$e_a(t) = i_a(t)R_a + L_a\frac{di_a(t)}{dt} + e_b(t) \tag{1}$$

Angular velocity:

$$e_b(t) = K_b\frac{d\theta(t)}{dt} \tag{2}$$

Torque equations:

$$T_m = K_T i_a(t) \tag{3}$$

$$T_m = J\frac{d^2\theta(t)}{dt^2} + B\frac{d\theta(t)}{dt} \tag{4}$$

Once all of the parameters have been measured, the students are asked to find the transfer function of input voltage to output speed using these equations and the measurements.

D. Lab 4: Step and Frequency Response System Identification

In the fourth experiment it is explained to the students that the system model of the DC motor can be estimated by a first order transfer function. For the first part of the experiment, student measure the parameters of the motors response to a square wave input. From collected data, a first order model is derived. Then several different frequencies of sine waves are input to the motor so that the students can estimate the Bode plot of the motor. From the Bode plot, they also derive a first order model.

At the end of this experiment students are provided a starter Simulink model, which contains a block for the students to enter their derived transfer function. The model allows them to design and test the control algorithms they will be asked to design in the next experiment.

E. Lab 5: Control of a DC Motor

The objective of the fifth laboratory is to design three different controllers: proportional, proportional plus derivative, and proportional plus speed feedback. In this experiment, each student designs each controller with analytical methods discussed in lecture then simulate the response of the controller with the motor model provided at the end of the previous experiment. The simulation includes an estimate of the friction in the motor, which means that the analytical gains are too small. Once they have adjusted the gains in the simulation, they deploy the algorithm to the Raspberry Pi and measure the actual response.

VI. CONCLUSIONS AND FUTURE WORK

A kit was developed for around $130 and replicates most of the experiments of the existing laboratory used in the course. The kit and experiments have been tested and are being used in half of the laboratory sections of the course in the fall semester of 2014. Data will be collected in the trial run in order to analyze the effectiveness of each type of equipment. Once the kit has been proven effective other experiments and pedagogies that take advantage of the kit's portability will be explored.

ACKNOWLEDGMENT

The authors would like to thank Dan Block, Controls Lab Specialist, Jim Leake, Director of Engineering Graphics, and Dr. Michael Loui, Professor of Electrical and Computer Engineering, for their support of this research.

REFERENCES

[1] A. Leva, "A hands-on experimental laboratory for undergraduate courses in automatic control," *IEEE Trans. Edu.*, vol. 46, no. 2, pp. 263–272, 2003.

[2] K. A. Connor, B. Ferri, and K. Meehan, "Models of mobile hands-on STEM education," in *120th ASEE Ann. Conf. and Expo.*, 23 June - 26 June 2013.

[3] D. Millard, M. Chouikha, and F. Berry, "Improving student intuition via Rensselaer's new mobile studio pedagogy," in *114th Ann. ASEE Conf. and Expo.*, 24 June - 27 June 2007.

[4] B. Aktan, C. A. Bohus, L. A. Crowl, and M. H. Shor, "Distance learning applied to control engineering laboratories," *IEEE Trans. Edu.*, vol. 39, no. 3, pp. 320–326, 1996.

[5] W. E. Dixon, D. M. Dawson, B. T. Costic, and M. S. D. Queiroz, "A MATLAB-based control systems laboratory experience for undergraduate students: Toward standardization and shared resources," *IEEE Trans. Edu.*, vol. 45, no. 3, pp. 218–226, 2002.

[6] M. Gunasekaran and R. Potluri, "Low-cost undergraduate control systems experiments using microcontroller-based control of a DC motor," *IEEE Trans. Edu.*, vol. 55, no. 4, pp. 508–516, 2012.

[7] L. D. Feisel and A. J. Rosa, "The role of the laboratory in undergraduate engineering education," *J. Eng. Educ.*, vol. 94, no. 1, pp. 121–130, 2005.

[8] P. H. Borgstrom, W. J. Kaiser, G. Chung, Z. Nelson, M. Paul, S. M. Stoytchev, and J. T. K. Ding, "Science and engineering active learning (SEAL) system: A novel approach to controls laboratories," in *119th ASEE Ann. Conf. and Expo.*, 10 June - 13 June 2012.

[9] F. Khan, N. Birchfield, and K. V. Singh, "Revitalizing the engineering curriculum through studio based instruction," in *ASME 2012 Int. Mech. Eng. Congr. and Expo., IMECE*, 9 November - 15 November 2012.

[10] C. M. Ionescu, E. Fabregas, S. M. Cristescu, S. Dormido, and R. D. Keyser, "A remote laboratory as an innovative educational tool for practicing control engineering concepts," *IEEE Trans. Edu.*, vol. 56, no. 4, pp. 436–442, 2013.

[11] O. Boubaker, "The inverted pendulum: A fundamental benchmark in control theory and robotics," in *2012 Int. Conf. on Edu. and e-Learning Innovations (ICEELI)*, 2012, pp. 1–6.

[12] A. C. Hyder, J. L. Thames, and D. Schaefer, "Enhancing mechanical engineering distance education through IT-enabled remote laboratories," in *ASME 2009 Int. Design Eng. Tech. Conf. and Comput. and Inform. in Eng. Conf. (IDETC/CIE)*, 30 August - 2 September 2009.

[13] B. Stark, Z. Li, B. Smith, and Y. Chen, "Take-home mechatronics control labs: A low-cost personal solution and educational assessment," in *ASME 2013 Int. Design Eng. Tech. Conf. and Comp. and Inform. in Eng. Conf.* ASME, 4 August - 7 August 2013.

[14] R. Egbert, "New electrical engineering laboratory facility combines traditional laboratory experiments, computer-based lab exercises, and labs taught via distance," in *2009 ASEE Ann. Conf. and Expo.*, 14 June - 17 June 2009.

[15] J. Sarik and I. Kymissis, "Lab kits using the Arduino prototyping platform," in *2010 Frontiers in Edu. Conf. (FIE)*, 2010, pp. T3C–1–T3C–5.

[16] J. Sanchez, S. Dormido, R. Pastor, and F. Morilla, "A java/Matlab-based environment for remote control system laboratories: Illustrated with an inverted pendulum," *IEEE Trans. Edu.*, vol. 47, no. 3, pp. 321–329, 2004.

[17] A. Cruz-Martin, J. Fernandez-Madrigal, C. Galindo, J. Gonzalez-Jimenez, C. Stockmans-Daou, and J. Blanco-Claraco, "A LEGO Mindstorms NXT approach for teaching at data acquisition, control systems engineering and real-time systems undergraduate courses," *Comp. & Edu.*, vol. 59, no. 3, pp. 974–988, 11/01 2012.

[18] Y. Kim, "Control systems lab using a LEGO Mindstorms NXT motor system," *IEEE Trans. Edu.*, vol. 54, no. 3, pp. 452–461, 2011.

[19] Studica. (2014) National instruments for students and faculty. Retrieved from http://www.studica.com/mydaq. Accessed: September 26, 2014.

[20] maxon motor. (2015) maxon motor online shop. Retrieved from http://www.maxonmotorusa.com/maxon/view/catalog/. Accessed: March 13, 2015.

[21] J. V. Nickerson, J. E. Corter, S. K. Esche, and C. Chassapis, "A model for evaluating the effectiveness of remote engineering laboratories and simulations in education," *Comp. & Edu.*, vol. 49, no. 3, pp. 708–725, 11/01 2007.

[22] R. Kelly and J. Moreno, "Learning PID structures in an introductory course of automatic control," *IEEE Trans. Edu.*, vol. 44, no. 4, pp. 373–376, 2001.

[23] G. Henderson. (2014) WiringPi GPIO interface library for the Raspberry Pi. Retrieved from http://wiringpi.com. Accessed: September 26, 2014.

[24] Mathworks. (2014) Raspberry Pi support from Simulink. Retrieved from http://www.mathworks.com/hardware-support/raspberry-pi-simulink.html. Accessed: July 29, 2014.