

Developing Experiments to Teach Control Theory

Dr. Stephen Andrew Wilkerson, P.E., York College of Pennsylvania

Stephen Wilkerson (swilkerson@ycp.edu) received his PhD from Johns Hopkins University in 1990 in Mechanical Engineering. His Thesis and initial work was on underwater explosion bubble dynamics and ship and submarine whipping. After graduation he took a position with the US Army where he has been ever since. For the first decade with the Army he worked on notable programs to include the M829A1 and A2 that were first of a kind composite sabot munition. His travels have taken him to Los Alamos where he worked on modeling the transient dynamic attributes of Kinetic Energy munitions during initial launch. Afterwards he was selected for the exchange scientist program and spent a summer working for DASA Aerospace in Wedel, Germany 1993. His initial research also made a major contribution to the M1A1 barrel reshape initiative that began in 1995. Shortly afterwards he was selected for a 1 year appointment to the United States Military Academy West Point where he taught Mathematics. Following these accomplishments he worked on the SADARM fire and forget projectile that was finally used in the second gulf war. Since that time, circa 2002, his studies have focused on unmanned systems both air and ground. His team deployed a bomb finding robot named the LynchBot to Iraq late in 2004 and then again in 2006 deployed about a dozen more improved LynchBots to Iraq. His team also assisted in the deployment of 84 TACMAV systems in 2005. Around that time he volunteered as a science advisor and worked at the Rapid Equipping Force during the summer of 2005 where he was exposed to a number of unmanned systems technologies. His initial group composed of about 6 S&T grew to nearly 30 between 2003 and 2010 as he transitioned from a Branch head to an acting Division Chief. In 2010-2012 he again was selected to teach Mathematics at the United States Military Academy West Point. Upon returning to ARL's Vehicle Technology Directorate from West Point he has continued his research on unmanned systems under ARL's Campaign for Maneuver as the Associate Director of Special Programs. Throughout his career he has continued to teach at a variety of colleges and universities. For the last 4 years he has been a part time instructor and collaborator with researchers at the University of Maryland Baltimore County (<http://me.umbc.edu/directory/>). He is currently an Assistant Professor at York College PA.

Dr. Stephen Andrew Gadsden, McMaster University

Dr. S. Andrew Gadsden is an Associate Professor in the Department of Mechanical Engineering at McMaster University and is Director of the Intelligent and Cognitive Engineering (ICE) Laboratory. His research area includes control and estimation theory, artificial intelligence and machine learning, and cognitive systems. Dr. Gadsden completed his Bachelors in Mechanical Engineering and Management (Business) and then earned his PhD in Mechanical Engineering at McMaster in the area of estimation theory with applications to mechatronics and aerospace systems. He worked as a postdoctoral researcher for nearly three years at the Centre for Mechatronics and Hybrid Technology (Hamilton, Ontario). He also worked concurrently as a Project Manager in the pharmaceutical industry (Apotex Inc., Toronto, Ontario). Before joining McMaster University, Dr. Gadsden was an Associate/Assistant Professor at the University of Guelph and an Assistant Professor in the Department of Mechanical Engineering at the University of Maryland, Baltimore County (USA). He worked and continues to work with a number of colleagues in NASA, the US Army Research Laboratory (ARL), US Department of Agriculture (USDA), National Institute of Standards and Technology (NIST), and the Maryland Department of the Environment (MDE). Dr. Gadsden is an elected Fellow of ASME, is a Senior Member of IEEE, and is a Professional Engineer of Ontario. He is also a certified Project Management Professional (PMP). Dr. Gadsden is a reviewer for a number of ASME and IEEE journals and international conferences.

Mr. Andrew Lee

Lee has applied his mechanical engineering knowledge and STEM teaching skills as a camp assistant in the Take Flight Robotics program, a summer experience designed to engage and inspire high school interested in STEM fields. In 2015, Lee helped participant

Developing Experiments to Teach Control Theory

by

Stephen Wilkerson York College of Pennsylvania

Andrew Gadsden McMaster University

Andrew Lee University of Guelph

This paper presents two Project Based Learning (PBL) control experiments that are low cost and non-proprietary. Moreover, both of these projects can be built by the students for under one hundred dollars in a short amount of time. Nonetheless, the development of these devices was not without difficulties. This is probably why many classes still do not include homemade control experiments or when they do the experiment becomes problematic. This paper provides an outline of two control experimental devices and a discussion of where the difficulties occurred in developing them. Sufficient details are provided herein to allow others to replicate the PBL experiments, minus the problems, and to expand on what is presented in this limited offering. The use of these devices has undergone an evolution over the past three years, and the paper include details of the projects use, benefits, and drawbacks. Finally, the paper provides observations of student learning from several semesters of use with recommendations for the future.

It was observed that developing control laws for a particular system can be confusing for many students. More often than not they don't know where to start. Not surprisingly, to facilitate a more complete understanding of control theory many classes include some hands-on experiments with various devices. Unfortunately, these experiments often require the learning of new software and hardware like LabView¹ and their products to create the control law. Moreover, the hardware and software requirements, not to mention the expenses, can detract from the purpose originally intended. Not surprisingly, the process of learning new unrelated materials can result in students not fully understanding how to control even the simplest of systems without the use of proprietary equipment and software. Fortunately, most engineering and computer science curriculums include work with microprocessors like the Arduino and programming in C or with MATLAB². This enabled the opportunity to develop some limited

¹ Labview is a product of National Instruments Corp: <https://www.ni.com/en-us/shop/labview.html>

² MATLAB is the trade mark of Mathworks: <https://www.mathworks.com/products/MATLAB.html>

experiments for use in classes. This paper presents two experiments that have been continually improved and redesigned over the past several years.

Introduction

As part of the process used to create Project Base Learning (PBL) experiences with students in control theory, it was important to give students hands-on experiments with actual hardware. Horáček [1] did a survey reporting on building control laboratories to support basic courses in control theory. The paper reports on how to build a new control laboratory with practical experiments. The goal of Horáček's paper was to assist others in building practical control experiments while saving time and money. Not surprisingly, there are numerous control simulations and mathematical models on control theory available, but far fewer hands-on *Do It Yourself* (DIY) experiments. Notwithstanding, this paper attempts to develop hands-on control experiments where the mathematics and programming are not hidden in layers of proprietary software and hardware equipment. Moreover, an attempt was made to give students a more in-depth multifaceted experience where they understand the nuances of the sensor, actuators and the programming used in the control experiments. Finally, this paper develops two control experiments that can be reproduced for less than one hundred dollars each. They can be built by students or faculty in a reasonable amount of time and lead to increased understanding of how to develop and implement the control of an apparatus. In these experiments, students were allowed to explore multiple paths to a solution. These included, direct programming with open source Arduino software, MATLAB traditional coding, and Simulink. Students were allowed to determine which method they wanted to use in each case.

Larmer and Mergendoller [2] point out that PBL needs to have two criteria to make it meaningful. The first stipulation was that students must perceive that the work is meaningful and the second was that it fulfilled an educational purpose. Both projects presented in this paper underwent numerous iterations that involved student input. Students built and programmed the devices and provided feedback on the problems encountered and, in many cases, how to fix those problems. As pointed out by Larmer and Mergendoller this resulted in a certain amount of "*Buy In*" by the students in the process beyond just learning control theory. The result of a several years processed are the two projects presented here. Both project's hardware designs are somewhat fixed, but their use is still undergoing small improvements with the hardware/software used to control them. Each of the project's designs are described and the evolution they are still undergoing is discussed herein. There are also descriptions of how student-use drove their further development.

PBL is well documented in the literature. Previous works were used to guide the efforts presented here [3,4]. Text book descriptions and other PBL projects were also referenced in the creation of the current project [5-8]. While PBL techniques were employed these were not the focus of this particular effort and are referenced here to acknowledge their contribution to the project's overall success. Notably, accurate models of a similar experiment are detailed by Bolívar-Vincenty et.al. [9] and used in teaching lectures but these are also not covered here. Modeling these experiments were not the focus of this offering but are acknowledged here as an important aspect of control theory.

Project 1 *The See-Saw Project*

Some of the evolution for the see-saw project as well as an assessment of the educational impact is detailed in another publication [10]. All of the parts can be made by a typical engineering school's shop for less than \$100.00. The frame is laser cut and constructed with some wood glue and screws. It could also be cut on a band saw, if a laser cutter-engraver is not available. The mechanized parts (such as the gears) can be 3D printed and are attached with glue

and screws. The design and all of the parts can be downloaded for free at³. There are also multiple versions of the design available at this location. Figure 1 shows the basic see-saw device and its response with a PID controller.

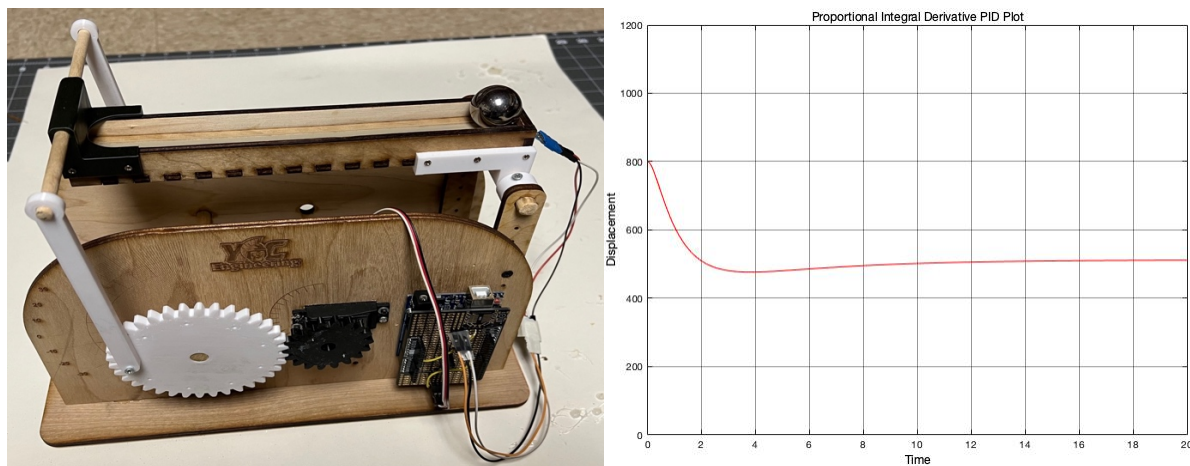


Figure 1. See-Saw Control Mechanism and typical PID Controller.

It was desired to use PBL and have a hands-on experience for the students. Prior efforts employed typical PBL tactics with positive student feedback. In this case, students were allowed to use what they had learned in the classroom and develop a Proportional, Integral, Derivative (PID) controller for the see-saw device. The objective was to keep the ball at a set location. So, for example, if you moved the ball by giving it a push with your finger; the mechanism would return the ball to a predetermined location on the see-saw beam. The sensor returns the position of the ball and the servo tilts the beam up and down to move the ball back and forth. Since the mechanism has only one sensor and one actuator the device could be considered simple in nature. Moreover, the control of the system is computationally and programmatically unpretentious as well. More on this later. Students were encouraged to do some of their own self-discovery with the experiments. While forming a pathway for a controller-solution it was desired that the solution was not taught specifically but rather discovered by the students. The project flow was meant to facilitate this self-discovery attribute as follows:

1. Learn how to read the location of the ball from the *softpot*⁴ sensor. They simply roll the ball back and forth by hand while recording position data. During this segment the students might notice noise in the sensor data. For example, with some of the devices an occasional data spike can occur. These are mainly caused by loose wires and the rolling path of the ball on the sensor. In most cases these single spikes can be ignored. However, by adding a noise or smoothing filter⁵ these issues will result in a more robust controller. Students are encouraged to look into filtering, but this is not required for the PID controller to function properly.
2. Learn how to move the see-saw up and down to move the ball; basically, moving the servo. Understanding the direction that the see-saw moves, the latency and other issues affecting the experiments are found here with some additional experiments.

³ <https://www.thingiverse.com/thing:4671664>

⁴ *Softpot* sensor: <https://www.sparkfun.com/products/8680>

⁵ Noise Filters: <https://blog.opticoncontrols.com/archives/1319>

Friction between the ball and *softpot*, while low, benefit the system's controllability. The importance of understanding the neutral position of the system becomes evident during these experiments. Students find the servo angle range where the ball will not roll on its own. So, for example, the neutral position might be with the servo at $90 \text{ degrees} \pm 2 \text{ degrees}$. So, whenever the servo is within those positions the ball does not move.

3. Using 1&2 above, create a simple bang-bang controller to move the ball back and forth over a single set point. Students are shown how to do this. i.e. The *softpot* is mapped to a range of 0-1000. Let's say the set point is 500. If the ball's location is > 500 the see-saw moves up and if it's < 500 it moves down. When this is implemented the ball slowly rocks back and forth about the set point. Students can experiment with delay times in the loop and servo speeds, and range of motion, during this portion of the exercise.

After this the students are mostly on their own to develop a PID controller based on what they have already learned during the semester about control theory.

4. Develop a Proportional P, Proportional Derivative PD and Proportional, Integral, Derivative (PID) Controller. The error between the set-point and the actual position time and a multiplication gain of K_P can be substituted in the bang-bang controller and the ball will continue to rock back and forth. If a good proportional gain term K_P and delay time are found, the ball will actually settle out at a location near the set-point, but typically not the desired location and not immediately. By taking into account the error and previous time-delayed error the system can be tuned to settle out almost immediately with an appropriate derivative gain term K_D multiplied by the error differences. Finally, by combining the error time-terms with an appropriate K_I multiplier the ball can be forced to settle almost immediately and at the desired or set-point location.

In the first year, the *softpots* were a source of some frustration due to the noise that routinely occurred in the device's connections. However, by incorporating Amphenol FCI Clincher Connector⁶ and an Arduino overlay board these issues were all but eliminated. The added cost to the device was negligible. Now the device required only 6 push pin wire connectors greatly reducing the complexity of the wiring and chance for loose and wriggling wires. This simplified the process and enabled the students to focus on the task at hand; namely writing a good controller. During the first year the students built the devices from scratch. That first year not all of the student teams were successful in the control of the see-saw. After the improvements already discussed everyone had a controller that worked the following year. It was concluded that the enhancements/improvements were working. Not surprisingly, each year the devices became more reliable and worked better as new ideas and fixes found their way into the most recent design.

Some of the odd issues that have occurred include students leaving their device in a hot car where the 3D printed gears warped, dropping the mechanism, pulled out wires, and other various problems encountered when allowing students to freely work on the devices in their possession. Nonetheless, the students could be routinely seen working on the project, experimenting, and improving their controllers outside of lab and class time. This is always

⁶ Amphenol FCI Clincher Connector: <https://www.sparkfun.com/products/14197>

encouraging for faculty. Three of the survey queries questions pertaining to this class and project device were: *Using the project enhanced my learning*, *Using the projects helped me understand system modeling better*, and *Using the project helped me understand system parameters better*. The scale 0 was strongly disagree and 5 Strongly agree. That graph is presented below in Figure 2 with zero as the lowest ranking and five as the highest.

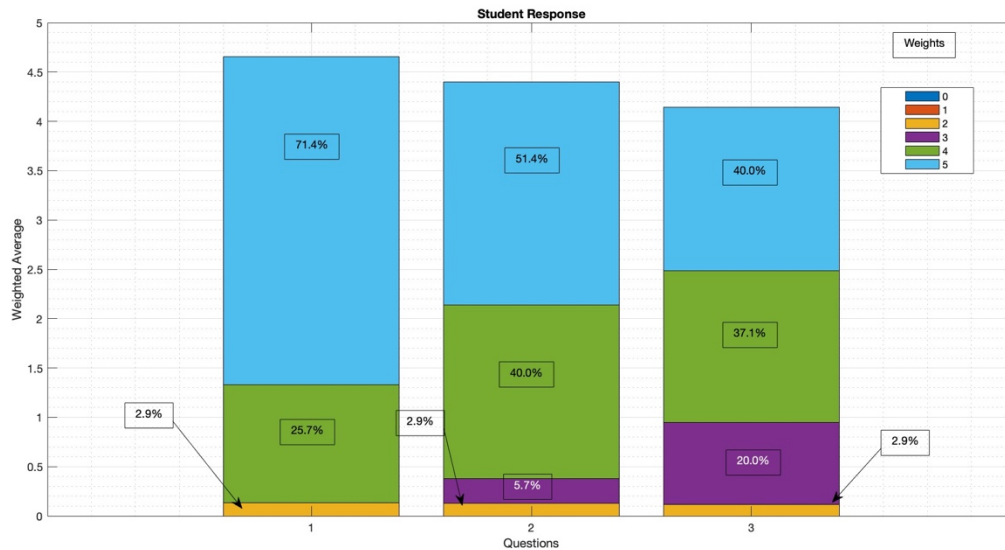


Figure 2. Survey Questions Targeted for Student Learning on the see-saw experiment

Project 2 Ping-Pong-Ball Levitation Device

This project was more difficult to design, control, and program for students. There were some initial difficulties blowing the ball up the tube with a small fan. The motor fan combinations resulted in a marginally controllable system and the motors typically required more voltage than was desired. With the initial system the motors needed a powerful external power supply to drive the fan and blow the ball up the tube. Worse still, at full power the ball barely went up. Increasing fan speed and power made things surprisingly worse.

Keep in mind that the purpose of the experiment wasn't to design ducted fan systems, rather to build a control system that students could use and to focus on the control of the system. Some initial experiments showed that the fan and duct efficiency could be improved. During this phase a series of wind tunnel experiments were developed examining some of the influences pertaining to efficiently of blowing a ping-pong ball up a plastic tube. It was desired to have the system as small as possible, so ducted fan size was considered, as were motor size, rotational speed and current. Number of fan blades were also considered, as was the twist of the airfoils on the blades (*constant pitch blades*) and the tolerances between blade diameter and duct housing. With some research into these devices the systems efficiency was greatly improved. The motor size was reduced and in the end 2 AA batteries could power the system and blow the ball up the tube efficiently. In the end (4) 1.5V AA or (4) 1.5V D type batteries were used to power the system.

Another issue that was encountered included the use of the L298 *Drok* motor controller. While quite inexpensive, that particular device's design used a fair amount of energy robbing the system of efficiency. Moreover, there were timing issues with turning the motor on and off that caused difficulties. The functionality was also greatly improved by using a 30N06L Mosfet.

Once again, the design made use of an overlay board for the Arduino Uno so that students would not spend time struggling with pushpin connections to get their control algorithms to work. Use of a Jameco 7.2V DC 14,260RPM motor (part number 177498), also helped the design. Finally, using a HCSR04 ultrasonic sensor to sense the distance to the ping pong ball seemed to make sense. Figure 3. Shows the system and the components used to make the device.

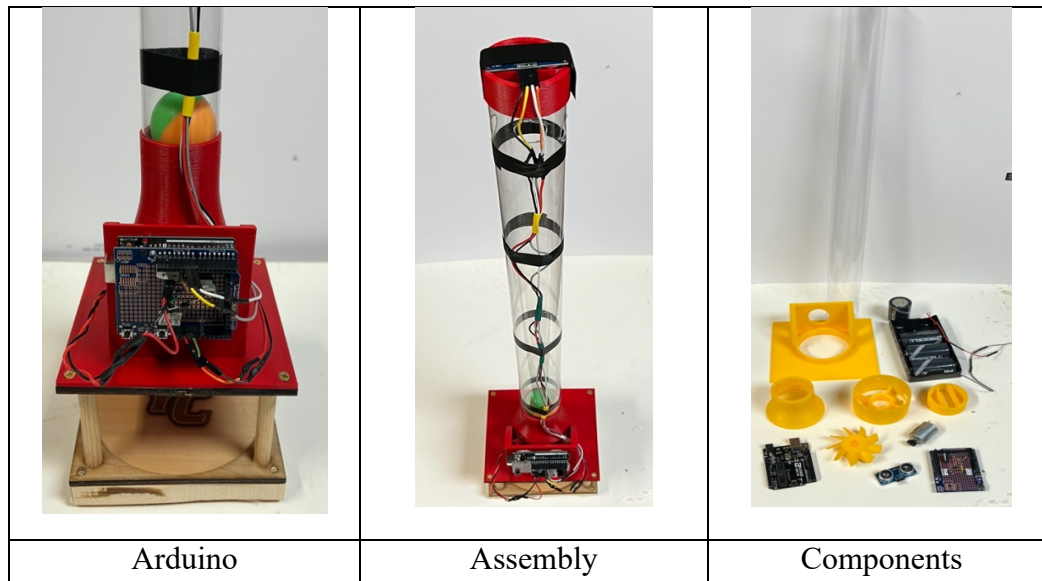


Figure 3. Ping Pong Control System Components

Student self-discovery was enabled by tailoring the initial experiments to form a pathway to a controller solution similar to what was presented in the see-saw experiment. The project flow is similar to the previous experiment. If the student's knowledge in the theory portions of the class are well understood, they can complete the control portions of the project without issue. The sequence tailored here was very much like the one used for the see-saw device and is as follows:

1. First, learn how to read the location of the ball from the Ultrasonic sensor. Students, in a similar fashion to the see-saw experiment, roll the ball back and forth in the tube while collecting data from the sensor. The sensor can be mapped as was done in the see-saw experiment. Latency in the sensor is more of an issue in this experiment as the sensor uses an ultrasonic sound pulse to determine the distance to the ball. Hence, the controller needs to wait long enough to detect the balls location before spooling up the motor. This makes the controller more difficult to tune.
2. Turning the motor on and off: Initially one can just connect the battery to the motor to get the correct polarity and watch the ball float to the top of the tube. Afterwards discovering this, the same "on-off function" can be accomplished with a single Arduino "*Digital.Write*" command to the motor controller. The digital write command turns the motor on and off. However, what is needed for a robust PID controller is PWM signals to test the system's response. At this point most of the students can finish the following two tasks without assistance. In this experiment the students used the "*Digital.Write*" command for the bang-bang controller and the Pulsed-width Modulation (PWM) commands for the PID.

3. Once again, the students are asked to build a Bang-Bang controller by turning the motor on when the ball was below a specific point and off when it is above that set point. Unlike the see-saw the ball drops rapidly, and the bang-bang controller can be seen as wholly inadequate.

After this the students are mostly on their own to develop a PID controller based on what they have already learned during the semester about control theory. For this experiment the conversion from bang-bang to proportional will result in a radically improved controller. Then by adding the derivative and integral portions of the controller the ball can be placed at a set location.

4. Develop a Proportional P, Proportional Derivative PD and Proportional Integral Derivative (PID) Controller. Initially students do this in Arduino C code, but then follow up by having them drive the system with MATLAB code and then Simulink control blocks.

Control and programming of these devices has primarily been focused on using the Arduino controller and software language with a little bit of MATLAB and Simulink use. Students with no prior Arduino experience can easily master coding techniques and write code to control both devices provided here. It is realized that this is not the only method that can be used to develop controllers for these devices. Currently efforts are underway to improve the diversity of approaches using the MATLAB Arduino interface and the Simulink Arduino interface software. During the upcoming semester students will also learn how to drive the system using these new techniques. While these other methods like Simulink are proven to be easier than programming the process directly, having them program the Arduino's is believed to still have value. By using Arduino programming students learn some of the nuances associated with sensors and actuator devices being controlled by a micro-processor. Future efforts may also explore other controller types like fussy logic and neuro-net controllers.

Conclusion

The details given here and associated links for the designs will enable anyone wanting to use these devices to reproduce the them. Having the students reproduce the devices is not overly time consuming and either of the designs presented here can be reproduced during a single semester without any difficulty. From observations with these devices it has been noted that the students overall understanding of the functionality by building them is limited. More importantly though, as more of these have been built, new students get the benefit of using them with less issues and wasted no time building them. Student feedback has provided us with evidence that the students do indeed enjoy and appreciate the use of these hands-on experiments in a typical undergraduate control course. In previous semesters where the devices hadn't been used students typically complained that the course had too much math in it. With the devices discussed in this paper, these complaints seem to have dissipated.

This paper presented two PBL control experiments that have undergone developed over the past 3 years. Both are low cost and non-proprietary and can be used for hands-on learning of control. The projects eliminate many of the time intensive development aspects of the devices by providing drawings and 3D printable parts that can be used to produce them. PBL coupled with hands-on projects has shown continued value in many applications to include control theory. Thus far faculty observations of students using these control experiments has provided

an enhanced interest in the topic from what was observed in the past using only math and simulation. Our current sample size of students is approximately 50 students per semester. However, as the class' popularity has grown, so has enrollment and currently there are several students taking the class during the spring semesters as an elective tutorial topic. The controls class is typically only offered in the summer semester.

Furthermore, these experiments provided here are easily expandable. Both experiments can be driven using the Arduino open source coding environment as has been done here. However, these devices can also be controlled using MATLAB Arduino libraries and Simulink block diagrams which have been done only limitedly so far. In the coming semester students will be taught how to interface both devices using the free MATLAB Arduino interface along with the current Arduino alone programming techniques that were used here. Furthermore, Simulink solutions will also be employed to a greater degree in the coming semester. Using the Simulink environment is far more like what students might encounter in industry. Not surprisingly, Simulink greatly simplifies the process making both devices very easy to control. Moreover, Simulink PID and other control features are easily used with these devices including the tune function found in Simulink. This paper offers these devices as an alternative to expensive proprietary control experiments that can be found elsewhere.

References

1. Horáček, Petr. "Laboratory experiments for control theory courses: A survey." *Annual Reviews in Control* 24 (2000): 151-162.
2. Larmer, John, and John R. Mergendoller. "Seven essentials for project-based learning." *Educational leadership* 68.1 (2010): 34-37.
3. Strobel, Johannes, and Angela Van Barneveld. "When is PBL more effective? A meta-synthesis of meta-analyses comparing PBL to conventional classrooms." *Interdisciplinary Journal of Problem-based Learning* 3.1 (2009): 44-58.
4. Blumenfeld, Phyllis C., et al. "Motivating project-based learning: Sustaining the doing, supporting the learning." *Educational psychologist* 26.3-4 (1991): 369-398.
5. Nise, Norman S. *Control systems engineering*. John Wiley & Sons, 2020.
6. Dorf, Richard C., and Robert H. Bishop. *Modern control systems*. Pearson, 2011.
7. Oliveira, Paulo Moura, and John D. Hedengren. "An APMonitor Temperature Lab PID Control Experiment for Undergraduate Students." *2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*. IEEE, 2019.
8. Hedengren, John, and Jeffrey Kantor. "Computer Programming and Process Control Take-Home Lab." *Computer* (2020).
9. Bolívar-Vincenty, Carlos G., and Gerson Beauchamp-Báez. "Modelling the ball-and-beam system from Newtonian mechanics and from Lagrange methods." *Twelfth LACCEI Latin American and Caribbean Conference for Engineering and Technology*. 2014.
10. Wilkerson, Stephen, and Andrew Lee. "Using Project Based Learning (PBL) with Control Theory." *2022 ASEE Annual Conference & Exposition*. 2022