ASEE 2022 ANNUAL CONFERENCE
Excellence Through Diversity
MINNEAPOLIS, MINNESOTA, JUNE 26TH-29TH, 2022 ASEE

Paper ID #36736

Using Project Based Learning (PBL) with Control Theory

Stephen Andrew Wilkerson (Assistant Professor)

Stephen Wilkerson swilkerson@ycp.edu received his PhD from Johns Hopkins University in 1990 in Mechanical Engineering. He retired from the Army Research Laboratory (ARL) Aberdeen Proving Grounds after 33 years of service. During the last 15 years of Stephen Wilkerson's work for the U.S. Army; his focus was on unmanned systems mainly drones and small robots. During his career with ARL he has been an instructor at the United States Military Academy West Point for three years and the exchange scientist to Germany. He is currently an Associate Professor at York College of PA. His current research interests include unmanned systems, drones, control theory, and astrophotography.

Stephen Andrew Gadsden

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. Dr. Gadsden has been the recipient of numerous international awards and recognitions. Most recently, in January 2022, Dr. Gadsden and his fellow air-LUSI project teammates were awarded NASA's prestigious 2021 Robert H. Goddard Award in Science for their work on developing an airborne lunar spectral irradiance instrument. He also earned the 2019/2020 University Research Excellence Award for the College of Engineering and Physical Sciences based on his research activities at the University of Guelph. Dr. Gadsden is also a 2019 SPIE Rising Researcher award winner based on his work in intelligent estimation theory, and a 2018 Ontario Early Researcher award (ERA) winner based on his work in intelligent condition monitoring strategies. He was also awarded the 2019 University of Guelph Faculty Association (UGFA) Distinguished Professor Award for Excellence in Teaching in the College of Engineering and Physical Sciences based on his teaching effectiveness and pedagogy.

Andrew Lee

Using Project Based Learning (PBL) with Control Theory

This paper is an Evidence Based Practice submission for Project-Based Learning of Automated Control Theory. The approach uses PBL rather than a traditional lecture and exam to evaluate student learning. Control theory and system identification can be very mathematically heavy. Whether analyzing these systems with *Laplace* transfers, in *State Space* or with differential equations in the time domain these problems can be tedious. Not surprisingly, it is often difficult for students to grasp the direct effects of the mathematical methods on controlling the system or plant. In other words, it can be difficult for students to fully grasp how to physically change their system in the same manner that mathematics does. To mediate this source of confusion most control course curriculums include some form of laboratory experiments. More often than not this requires the learning of new software like LabView to drive the control experiments. It also includes other hardware that can be equally expensive. This can leave students dependent on expensive hardware and software to control even the simplest of systems.

In this paper we compare two approaches; one with MATLAB and another more direct approach with an Arduino based controller. In both cases the students need either prior knowledge of MATLAB or Arduino programming to complete the experiments. For the see-saw experimental apparatus that we developed, we provide the details and costs associated with making and developing these experiments. We show the contrast between the two experiments using student surveys and provide the results. We also examine the differences between student comprehension as a measure of merit of the two methods. In particular we examine a low-cost 1st order temperature apparatus developed at BYU and then develop a 2nd order see-saw balancer of our own. In both cases these devices cost less than $50.00 to produce and our students were able to build these without much difficulty. The usefulness of physical models in control curriculums cannot be underestimated. Students getting hands-on experience controlling mechanisms or circuitry benefit from these real-world experiences. We provide the data taken from the students as evidence of merit.

Introduction

Project Based Learning (PBL) has many advocates for classroom motivation and learning. Blumenfeld et. al. [1] presents an argument for PBL helping students learn. Also examined in this study is the sustainability issue with regards to PBL and motivation. Wilkerson et.al [2-3] has shown good student motivation in learning new topics using PBL in a capstone design program. In these efforts it was shown that students will self-motivate and therefore gain more in-depth knowledge then what might be found in a typical classroom environment where only specific areas are covered. Strobel et. al. [4] research indicated that PBL is superior for long-term retention. However, in this same study traditional approaches appeared to be more effective for short-term. Control theory is a topic where examples are paramount in a student's understanding of the topic. Control theory techniques are mathematically heavy and therefore learning through visualization is likely to be a key factor. Therefore, it can be argued that understanding the outcome of a control method requires some hands-on experimentation.

It is not surprising that most control books have numerous examples and suggestions for projects. Norman Nise [5] and Richard Dorf [6] texts included numerous mathematical representations of physical systems. Furthermore, both texts make extensive use of Matlab to plot and display results. Nonetheless, the need for hands-on projects is evident and often included in control programs. Quanser[1] is one company that produces high-end control experiments that can be used in the classroom. These control experiments often use Lab-View[2] and require a substantial financial commitment by the institution to supply these for the curriculum. An additional feature to the expense of these experiments is the amount of time spent learning to use yet another software and or hardware platform. In the end the students are able to control an apparatus, but may not have a mastery of the methods, but rather the ability to use the software and hardware provided by the vendor.

More recently Hedengren et. al. [7-8] has shown that simple experiments can be done inexpensively in a control lab using an Arduino with some simple programming techniques. The components of this 1st order system have been developed into a kit that cost less than $40.00 (See figure 1).

Figure 1) Dynamics and Control Lab by John Hedengren[3]

---

[1] Quanser: https://www.quanser.com/

[2] Lab-View: https://www.ni.com/en-us/shop/software/products/labview-control-design-and-simulation-module.html

[3] Dynamics and Control Lab: http://apmonitor.com/che436/index.php/Main/PhysicalLab

These experiments were also shown to be expandable to use with MATLAB and Simulink without making the entire curriculum about learning new tools. In this paper we look at the PBL lab given by Hedengren, et. al. on a 1st order system and then expand to include similar techniques with a 2nd order see-saw device that the students can build and learn from. For both devices we provide some examples of the basic experiments and how to implement a Proportional Integral Derivative (PID)[4] controller. PID controllers [9-12] are universally used for many modern control systems as they are easy to tune and provide sufficient fidelity for many applications. In the second case we provide the details on how to build your own 2nd order see-saw system and then examine learning outcomes from the two PBL labs. The see-saw is an inherently unstable system, and this makes the control system relevant to a number of control techniques. Designs and a parts list are provided for future users. Examples are given using the system as a stand-alone device and then using MATLAB/Simulink models to guide. Finally, we provide some observations from both labs for the learning activities and outcomes.

1st Order System

The details of the Heat Transfer, 1st order system, were developed by Hedengren, et. al. and provide an excellent controls introduction. The details can be found on the web and the kit can be purchased for $35.00. Since we did not develop this PBL kit, we briefly summarize the learning advantages/disadvantages that we observed using this PBL lab in Table 1.

Table 1. Advantage and disadvantages observed in Heat Transfer PBL

Advantages	Disadvantages
Inexpensive, can also be built from scratch.	System is not real time. Slow response requires considerable time to repeat and fix experiments.
Provides a stable system that can be assumed linear.	1st order system not as good for rise time and overshoot experiments
Extremely easy to control and program. Very well-designed project-based learning lab.	Not good for demonstrating root locus design techniques
Easy to collect data for plots and easy to develop a PID controller with.	No vibrations or instability in the system.
Relatively easy to model and can be developed from principles of Heat Transfer. Excellent for developing a FOPDT[5] Can also be extended to include non-linear effects.	Forcing function is on or off. The power can be adjusted, but in general it is an on or off function.
Works well with MATLAB and Simulink in real time	Time consuming.

Figure 2 shows the FOPDT experiment compared with the experimental results. As can be seen by the plot the model and experiment are in excellent agreement.

---

[4] PID Proportional Integral Derivative Controllers: https://en.wikipedia.org/wiki/PID_controller

[5] FOPDT: First Order Plus Delay Time mode: https://apmonitor.com/pdc/index.php/Main/FirstOrderSystems .

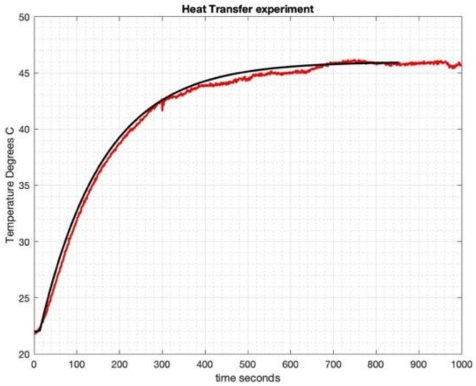


Figure 2 FOPDT model of Temperature control Lab

2nd Order System Design

Our Second Order system is still undergoing an evolution. For completeness, a brief description for the design development is given here. We started with several different concepts shown in Figure 3(a, b, c). The initial idea (a) with magnets proved difficult for students to balance or even to model. Making that model push on the permanent magnets (reverse the poles) made it easier to control, but still difficult for most students. Also, including a spring on the model made it easier to control. The second design (b) was better but unstable when the ball moved too far out on the seesaw the servo struggled to correct, making the system jumpy. The biggest issue was the weight of the ball versus the torque of the servo. To mathematically model that system required a coupled model of the ball and the beam [13]. Other models were also available [14], but these were beyond the scope of this introductory class. An easily reducible representation was desired, this is possible with our final design. The final design and our focus in this paper is given in (c).

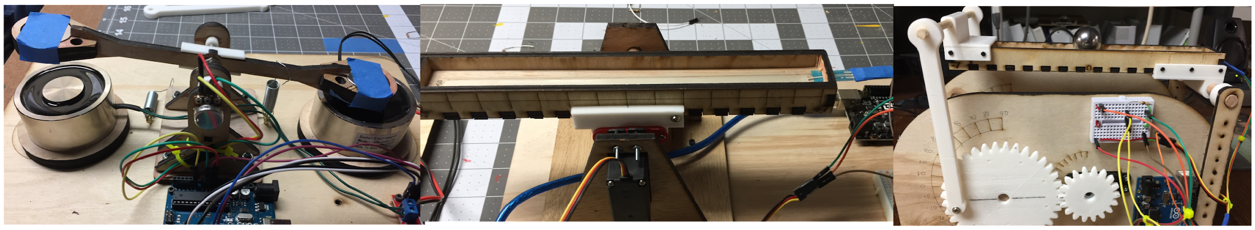

a) Magnet Seesaw b) Direct Drive Center Mount c) Seesaw Gear Driven

Figure 3. Evolution of seesaw control project

The unit was further simplified by the fact that the servo could be controlled directly by the Arduino unit without an H-bridge or other additional circuitry. In fact, the wiring only required 7-8 wires and one 10KΩ resistor. A complete wiring diagram is provided in Appendix A. The drawings and all of the 3D printed parts can be found on Thingiverse[6] along with a parts list and build notes.

Analysis

---

[6] Thingiverse: https://www.thingiverse.com/thing:4671664

Students are shown how to program the Arduino to collect the data from the potentiometer soft pot and how to set the angle of the servo. Code to do this is also provided in Appendix A. The soft pot has an analog range of 0-1023. The servo range is from 0 to 180 with 90 degrees being the neutral point. The rolling of the ball, assuming no slip and small angles, can be shown to be proportional to the acceleration.

$$\sum_{i=1}^{n} M_i = I_{Ball}\ddot{\theta}, \text{ where } I_{Ball} = \frac{2}{5}mr^2 \text{ and } sin\ sin\ (\theta) \ = \ \theta$$

This leads to a simple assumption that our model can be approximated by $\ddot{x} = \alpha\theta$ or $\alpha\theta + \beta, where\ \alpha\ and\ \beta\ are\ constants.$ By conducting a simple roll test, we obtained an approximation of where α = .9. Figure 4 shows that comparison.

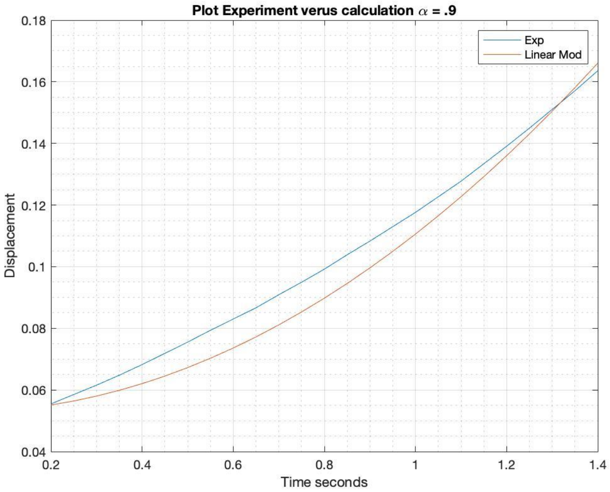


Figure 4 Alpha calculation

The initial position corresponds to the zero location on the soft pot that coincides with the inscribed "0" on the number line and that is approximately the mid-point or 512. Experiments can be done by giving the ball a push and that would correspond to an initial velocity or simply placing the ball in a position like 800 on the soft pot. Then the system will attempt to relocate the ball to the midpoint. Students are allowed to experiment with the device prior to making this simple model. By experimenting with first proportional K_p only, they will find that the system oscillates back and forth as would be expected. Adding in a derivative K_d term they should be able to get the ball to arrive at the midpoint without too much difficulty. However, they quickly find that they have little control on the rise time or settling time or for that matter the number of oscillations. Moreover, a properly tuned controller is difficult to achieve when adjusting these constants heuristically. This is where the model and root locus can be added into the project. Our Simulink Model as shown in Figure 5.

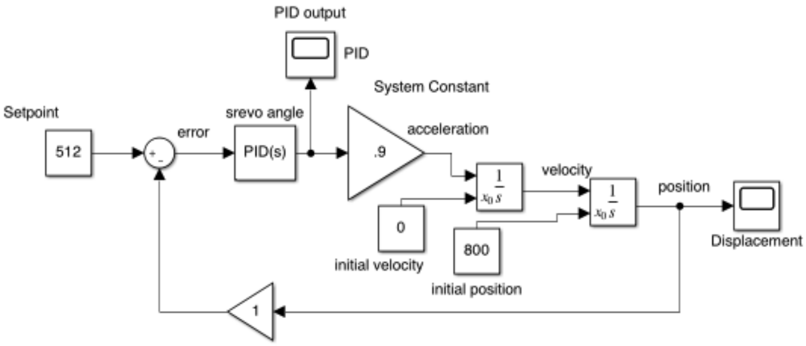


Figure 5. Simulink model

The Simulink model is surprisingly accurate and exhibits the same behavior as the experimental device. Figure 6 shows a Proportional P, Proportional Derivative PD, and a full PID Simulink model results. Finally using the MATLAB tune function new PID values can be found and are shown in Figure 7. Using the tune function the desired control features can be obtained.

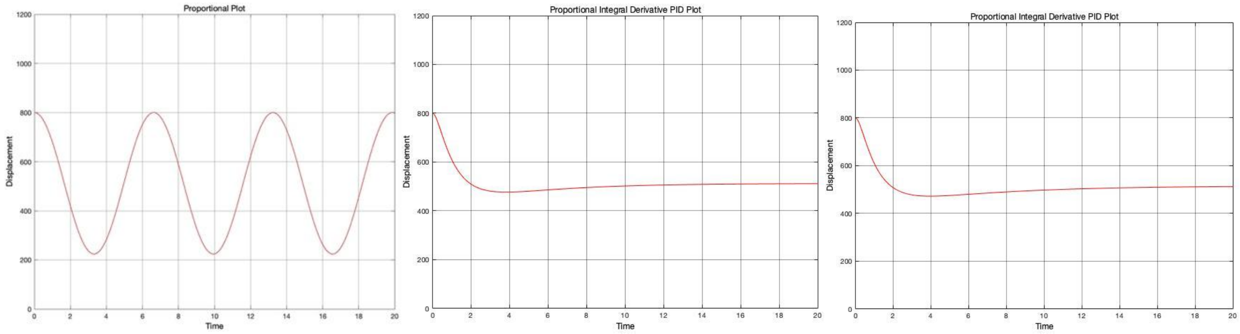
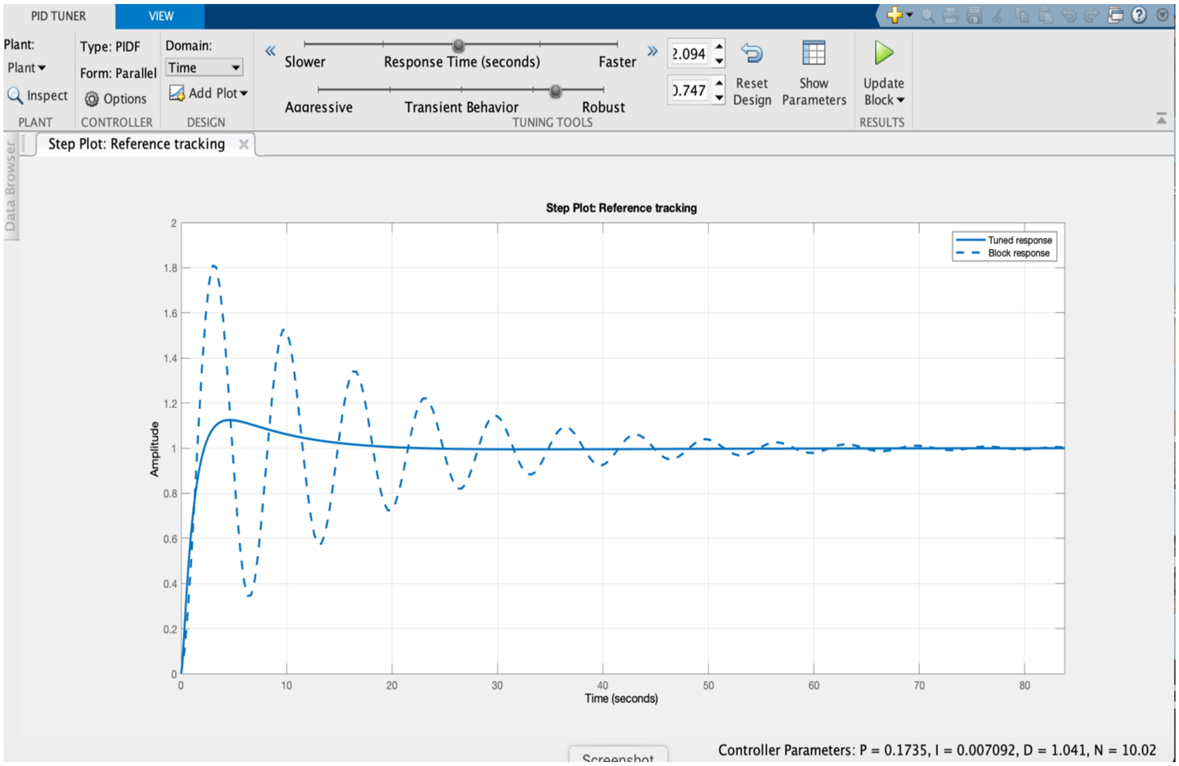


Figure 6. P, PD and PID Simulink Model Results.



Type: PIDF

Form: Parallel

Domain: Time

Controller Parameters: P = 0.1735, I = 0.007092, D = 1.041, N = 10.02

Figure 7 MATLAB Tuned model

Using the equivalent ratios found from the model the experimental apparatus can be shown to exhibit the same behavior. Moreover, the model can be tuned for specific rise time overshoot parameters or a known damping ratio ζ. This also proved an excellent time to show the relationship between the root locus and the design. It is important to note that there are some differences in the way the PID function is formulated in MATLAB and a PID of the form $s^2K_i + sK_d + K_p$. In general, the students were better able to program a controller system while relating it to a purely mathematical model with this simplified model. Examining a root locus plot for the system with a 5 to 1, K(s+5), ratio between the Kd gain and the Kp gain allows the students to experiment with both theory and experiment. Figure 8 shows that plot. In the coming semester we will expand this by having different teams with different requirements and having the students present their findings with the models and experiments.

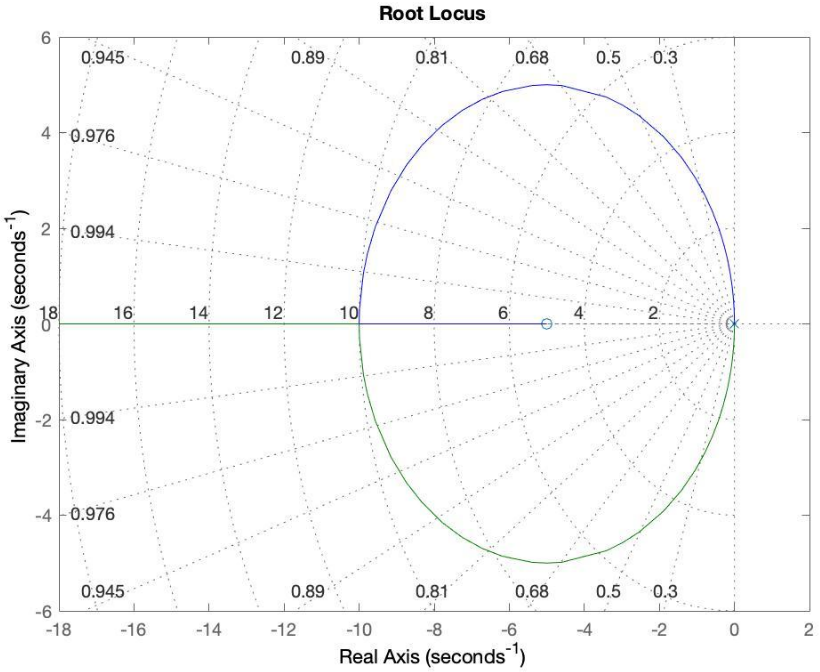


Figure 8 Root locus with a 5 to one P to D ratio

Results

The seesaw PBL exercise was used at two different universities so far with large groups of students. The students were instructed to code a PID controller for a servo motor in order to have a ball bearing settle at a desired setpoint. The position feedback was measured using a soft potentiometer. The main goal of these experiments was to give students experience in programming discrete/digital controllers. Classical control theory at the undergraduate level focuses mostly on continuous control and Laplace transforms. Students often find these concepts to be abstract and have a hard time relating them to physical systems. By programming their own controller, students are able to directly observe the effects of adjusting proportional, derivative, and integral gains on system performance. Furthermore, it allows students to look at the system holistically. Students must code their controller to gather and store measurements from a sensor and clock, feed that data into an algorithm, and output a signal to an actuator to achieve a desired performance. This also reinforces how to program derivatives and integrals into a discrete PID controller which students often forget.

Students recorded their results using the Arduino Serial Monitor and Arduino Serial Plotter. They were asked to use several different gain values to observe the effects on performance parameters such as overshoot, settling time, and steady state error. This greatly enhanced their understanding of PID controllers as they were able to observe the results in real time and graphically.

We believe that the new devices are a significant improvement to the labs conducted prior; which did not incorporate system modeling in MATLAB and Simulink. System modeling is a very important aspect of merging theory with real physical systems. The Arduino required direct programming to control the system. This attribute helped bridge the gap between what can be done in Simulink and what the code is actually doing to control the system. Therefore, the addition of Simulink modeling along with the Arduino programming helped bridge the understanding between block diagrams seen in textbooks and a discrete PID program for the microcontroller. We used a progression of controllers to include a bang-bang controller and then onto Proportional (P), Proportional Derivative (PD) and finally a Proportional Integral Derivative (PID) control systems. This enabled students to understand what each portion of the controller did for the system. This also helped relate the controller back to the 2nd order differential equation used for the mathematical approximation.

Student Data

Table 2 provides the control questions given to the students. Of particular interest were questions 1 and 2 which relate their perceived knowledge before the class and afterwards. Figure 9 provides the average values and the confidence intervals at the 95% level.

Table 2. Control Questions

#	Questions	#	Questions
1	Rank your controls knowledge prior to this controls class	8	MATLAB was helpful in visualizing system behavior
2	Rank your controls knowledge after this controls class	9	I now understand how a feedback system works, closed loop
3	Hands on experiments enhanced my learning Project 1	10	Simulink made system modeling easier
4	Hands on experiments enhanced my learning Project 2	11	I better understand steady State Errors
5	The projects improved my understanding of system modeling	12	My understanding of system noise improved
6	Using the projects helped me understand system parameters better	13	My understanding of noise reduction improved
7	The project helped me understand mathematical representations of systems		

From questions 1 and 2; overall the students believed that they learned a good deal in the class and the confidence interval shows that the value is significant. Questions 3 and 4 indicate that the student felt they benefited from the two hands-on experiments in the class. The average score for the second experiment using a 2nd order system was slightly larger than the one for the 1st order system, but the confidence intervals indicate that the difference is not significant. Questions 5-7 were asked in the contents of the projects. In other words, did the projects in this class help improve your understanding of the topic? These three questions received mixed reviews and there appeared to be more scatter in the data based on the standard deviation. Questions 8 and 10 were used to gauge the tools used in the project and whether they helped students better understand the topic. Question 9 asks if the students felt they understood the feedback mechanisms in control. Finally, questions 11-13 are related to the understanding of system errors and noise that were covered by the projects. These scores in general were lower than the other questions and indicate that there is room for improvement in these areas. They also had standard deviations slightly greater than 1. In all there were 45 students surveyed for the first time offering this class at the school. By time of the presentation of this material we will have additional data to consider.

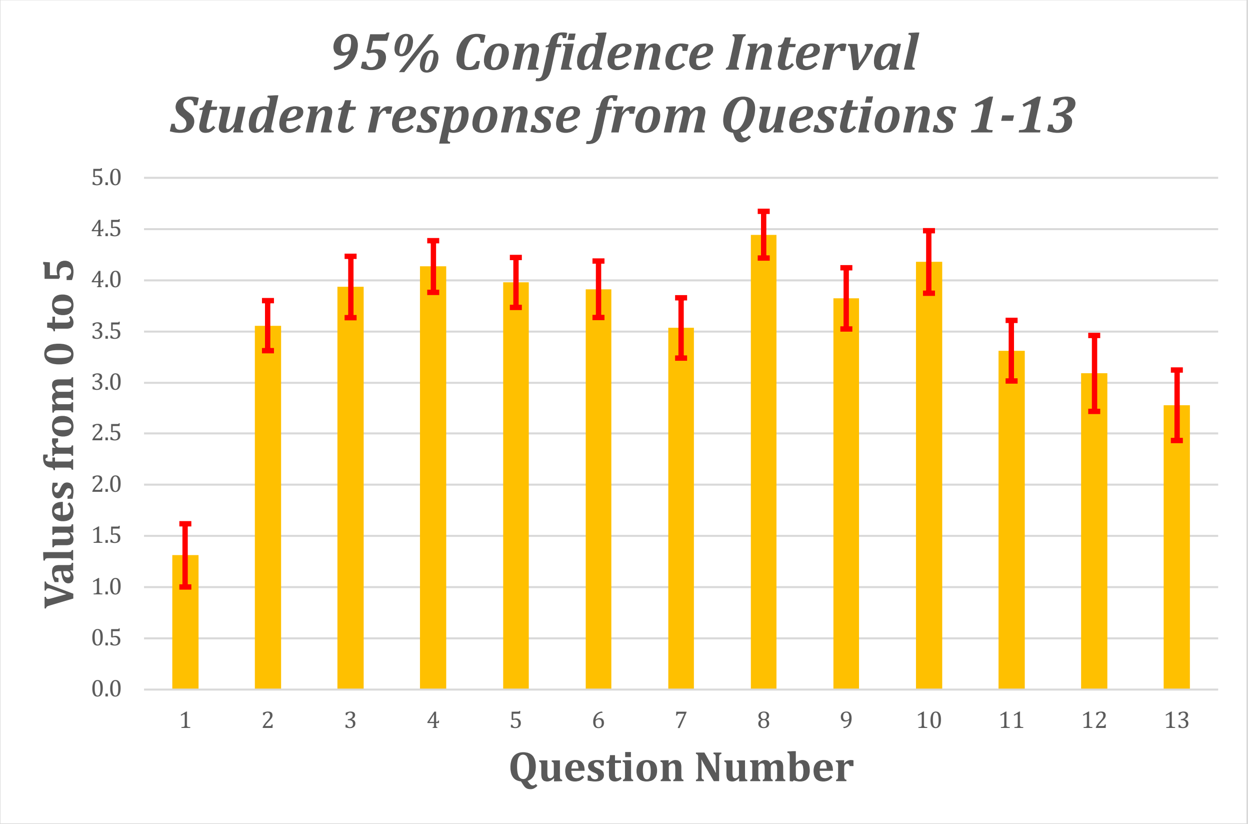


Figure 9 Student Responses to Control Questions.

Conclusions

In this evolution of PBL for control we have seen that the difficult experiments like the magnetic balancing system can be next to impossible for the students to get working. In our first attempt the magnet devices had only limited success. However, that model used a potentiometer for the feedback loop, but that had limited detection range making the system difficult to control. Simpler models like the temperature experiments are good, but don't have the higher order dynamic response or real time feel of the more complicated models. After an initial attempt the magnet model was simplified with the soft pot feedback resulting in more than 75% of the student teams having success.

In the current semester we will include some targeted modeling and experimental labs with the mechanical systems. Initial experimentation with the seesaw version 2 shows promising results. This model also lends itself to modeling with simple Simulink, root locus, and state space models to guide the process. We will include student surveys in our next semester and report on the findings in subsequent publications. Finally, the mechanical see-saw device appears to be scalable. It may be possible to make a miniature self-contained version of the see-saw similar to the temp lab provided by [7-8]. The design details for this device and similar variations are provided for others to use on Thingiverses[7] web site [15-17]. We will allow some of the student teams to undertake this challenge and report back in a subsequent publication. A new control experiment has been developed [18] and will be included in future classes.

---

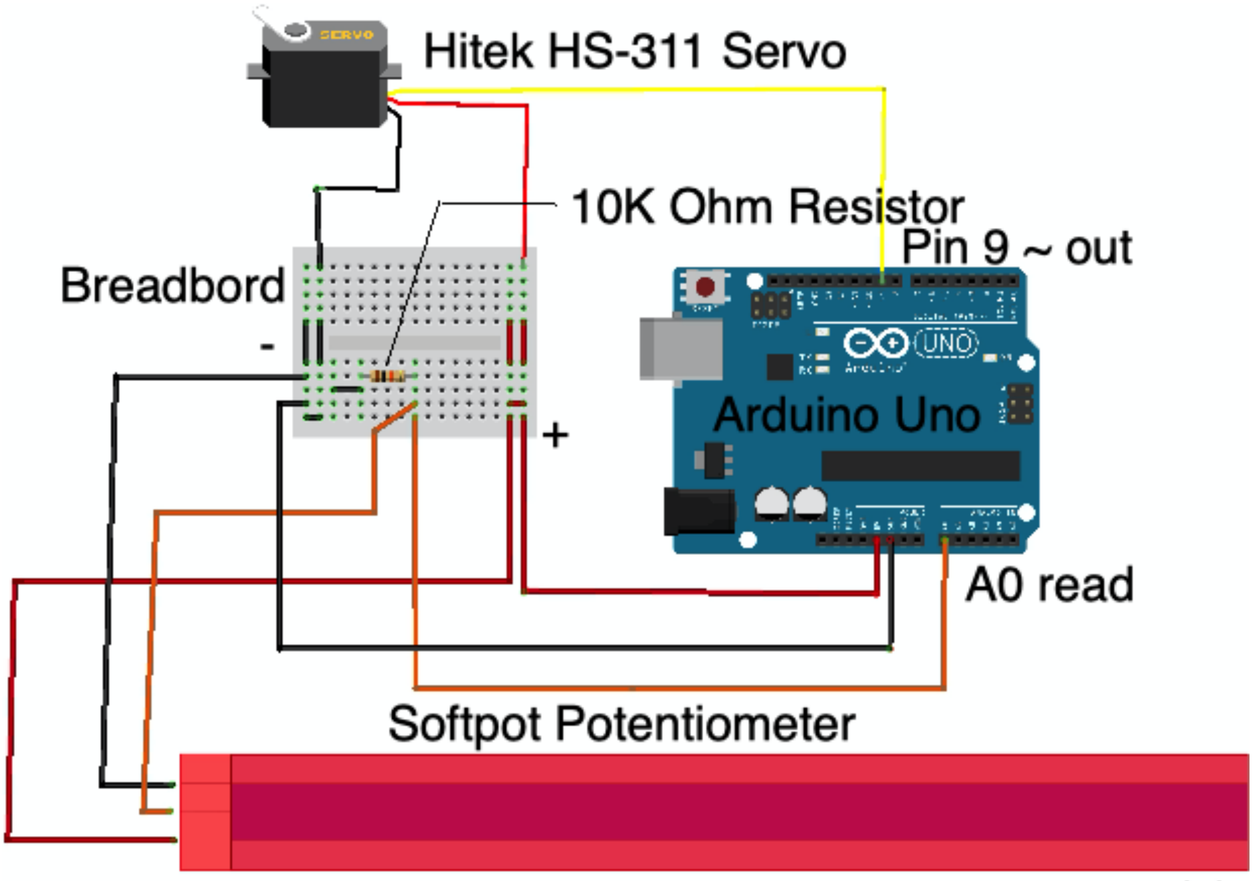
[7] Thingiverse, has a free compendium of plans that can be 3D printed, laser cut or used to construct a number of useful projects: https://www.thingiverse.com/

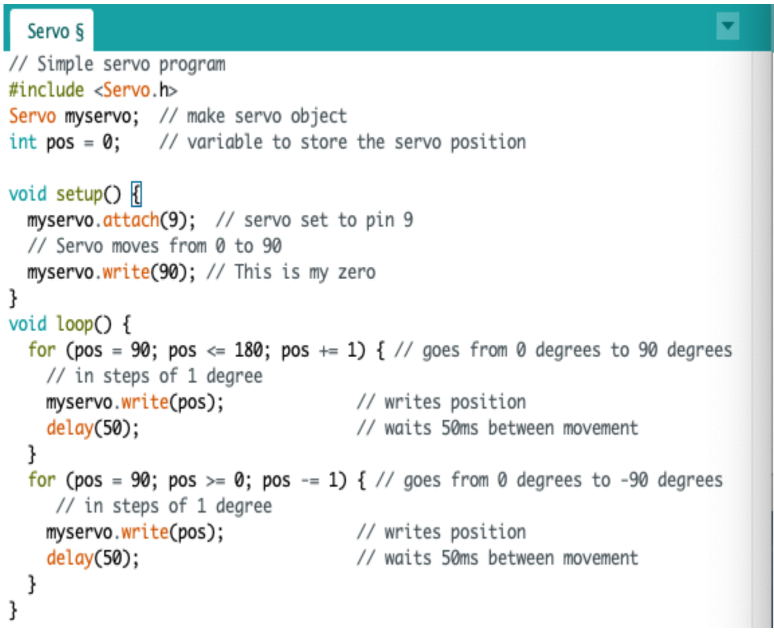
References

1. Blumenfeld, Phyllis C., et al. "Motivating project-based learning: Sustaining the doing, supporting the learning." *Educational psychologist* 26.3-4 (1991): 369-398.
2. Wilkerson, S. A., A. D. Gadsden, and S. A. Gadsden. "An unmanned aerial system for the detection of crops with undergraduate project-based learning." *Autonomous Air and Ground Sensing Systems for Agricultural Optimization and Phenotyping III*. Vol. 10664. International Society for Optics and Photonics, 2018.
3. Wilkerson, Stephen. "Multispectral Analysis of Farm Corn Crops: A Project-Based Learning (PBL) Program Stephen Wilkerson, Assistant Professor, York College, York, PA Joe Cerreta, Assistant Professor, College of Aeronautics, Embry-Riddle Aeronautical University, Worldwide Department of Flight."
4. 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.
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. Porter, Brian, and A. H. Jones. "Genetic tuning of digital PID controllers." *Electronics letters* 28.9 (1992): 843-844.
10. Yu, Cheng-Ching. *Autotuning of PID controllers: A relay feedback approach*. Springer Science & Business Media, 2006.
11. Valério, Duarte, and José Sá Da Costa. "Tuning of fractional PID controllers with Ziegler–Nichols-type rules." *Signal processing* 86.10 (2006): 2771-2784.
12. Datta, Aniruddha, Ming-Tzu Ho, and Shankar P. Bhattacharyya. *Structure and synthesis of PID controllers*. Springer Science & Business Media, 1999.
13. Bolívar-Vincenty, Carlos G., and Gerson Beauchamp-Báez. "Modeling the ball-and-beam system from Newtonian mechanics and from Lagrange methods." *Twelfth LACCEI Latin American and Caribbean Conference for Engineering and Technology*. 2014.
14. Buza, Gergely, and Tamas Insperger. "Mathematical models for balancing tasks on a see-saw with reaction time delay." *IFAC-PapersOnLine* 51.14 (2018): 288-293.
15. Prof Hedengren temperature control lab: http://apmonitor.com/che436/index.php/Main/PhysicalLab
16. See-saw design and details of how to build and program the device: https://www.thingiverse.com/thing:4671664
17. Mini See-saw design with design and build instructions: https://www.thingiverse.com/thing:4703533
18. Ping-pong ball control experiment: https://www.thingiverse.com/thing:4708399

Appendix A

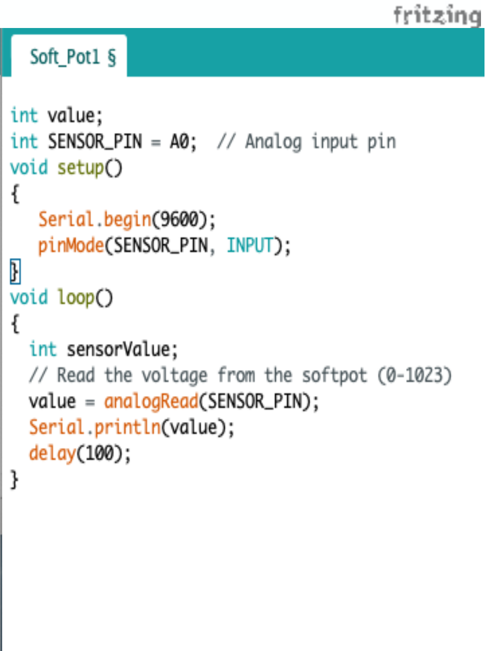
Seesaw Control Unit Wiring Diagram





```
// Simple servo program
#include <Servo.h>
Servo myservo;  // make servo object
int pos = 0;    // variable to store the servo position

void setup() {
  myservo.attach(9);  // servo set to pin 9
  // Servo moves from 0 to 90
  myservo.write(90); // This is my zero
}
void loop() {
  for (pos = 90; pos <= 180; pos += 1) { // goes from 0 degrees to 90 degrees
    // in steps of 1 degree
    myservo.write(pos);              // writes position
    delay(50);                       // waits 50ms between movement
  }
  for (pos = 90; pos >= 0; pos -= 1) { // goes from 0 degrees to -90 degrees
     // in steps of 1 degree
    myservo.write(pos);              // writes position
    delay(50);                       // waits 50ms between movement
  }
}
```



```
int value;
int SENSOR_PIN = A0;  // Analog input pin
void setup()
{
   Serial.begin(9600);
   pinMode(SENSOR_PIN, INPUT);
}
void loop()
{
  int sensorValue;
  // Read the voltage from the softpot (0-1023)
  value = analogRead(SENSOR_PIN);
  Serial.println(value);
  delay(100);
}
```