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

Paper ID #37796

A Novel Cart/Pendulum System for Teaching Dynamic Systems and Feedback Control

Ryan W Krauss (Associate Professor)

Ryan Krauss teaches dynamic systems, feedback control, robotics, and mechatronics. He is also actively involved in first-year, cornerstone design. He teaches at a medium-sized state university in the Midwest.

A Novel Cart/Pendulum System for Teaching Dynamic Systems and Feedback Control

Abstract

This paper presents a novel cart/pendulum system for teaching dynamic systems and feedback control and discusses the use of the system in a class project. The cart has a pendulum attached to it that can be used for vibration suppression control in the downward position or for stabilizing the inverted pendulum in the upward position. A line sensor is attached to the front of the cart for line following. The cart/pendulum system has been designed to perform three different experiments. The cart is controlled using the combination of a Raspberry Pi and two Arduinos. Students program their control logic in Python.

The class project is to program the robot to compete in three different events in a robot triathlon. The first event involves vibration suppression of the pendulum after it is given an initial flick (disturbance). The second event consists of racing around an oval line-following track while suppressing the vibrations with the pendulum in the downward (stable) position. The third event is to see how long the robot can balance the pendulum in the inverted (unstable) position.

Student learning is assessed using final project reports, exam questions, and a comparison of student evaluation comments between this year and last year.

Introduction

This paper presents a novel, low-cost cart/pendulum robotic system for teaching dynamic systems and control and discusses its use in a class project. The cart/pendulum robot has been designed to perform three different experiments: vibration suppression of the pendulum, line following with or without vibration suppression, and stabilization of the pendulum in the inverted position. Pictures of the left and right sides of the robot are shown in Figure 1.

The robot performs real-time feedback control experiments and uses the combination of a Raspberry Pi and two Arduinos to form a wireless, open-source system. The students programmed the robot in Python while lower-level Arduino code was provided to them.

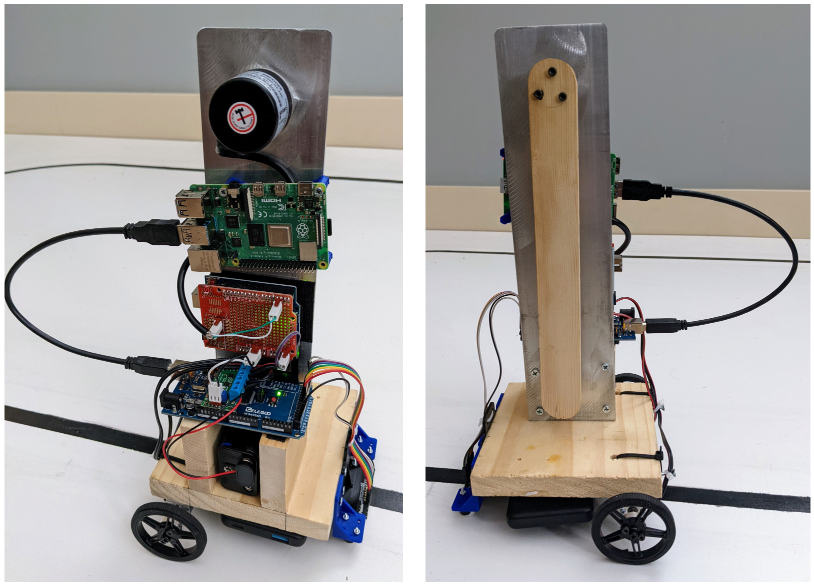

Figure 1: Pictures of the robotic cart/pendulum system (left and right hand side views).

Literature Review

Dynamic systems and feedback control can be abstract and mathematically intensive. Courses on these subjects are sometimes taught from a purely theoretical point of view. Feedback control experiments can be extremely valuable in deepening understanding and making the content more concrete [1, 2].

There are several issues associated with bringing experiments into feedback control courses. One challenge is that control actions need to occur at hard real-time intervals on the order of milliseconds. Additionally, hardware and software are needed to read sensor signals and send actuator commands. There are multiple commercially available solutions to these problems and many of them have been used successfully in feedback control education [3, 4, 5]. However, most of these approaches are fairly expensive.

There have been many recent advances in low-cost controls experiments, including some that are referred to as student-owned experiments. In these courses, students sometimes purchase a hardware kit in lieu of a textbook [6, 7, 8, 9].

This work falls under the umbrella of low-cost feedback control experiments for education.

System Description

The robot is powered using two DC motors in a differential drive configuration. The motors are driven by an H-Bridge shield connected to an Arduino Mega. The robot has two primary sensors: a reflectance sensor array used to sense line position and an encoder used to sense pendulum position. The line sensor is connected directly to the Arduino Mega. An Arduino

Uno is dedicated two responding to encoder interrupts to read the pendulum rotational position. The Arduino Mega and Uno communicate via i^2c. One novel feature of the robot is that the pendulum can operate in two configurations: the stable, downward position or the upright, unstable position. In the downward position, experimental system identification can be performed and used in vibration suppression control design.

A Raspberry Pi is used to allow the students to wirelessly connect to the robot over Wi-Fi. The Raspberry Pi provides a remote desktop login environment and also allows the students to program higher level controls in Python. The Raspberry Pi and Arduino Mega send small amounts of data back and forth in real time over a USB serial connection. The digital control frequency is limited to roughly 100 Hz. This relatively low control frequency is partially due to latencies in USB serial communication, but another contributing factor is that reading the reflectance sensor array can take as long as 5 milliseconds.

The combination of an Arduino and Raspberry Pi to form a feedback control system offers several advantages. It is easy to use an Arduino to enforce real-time execution of the control law at the milliseconds level; because the Arduino is a microcontroller that does not really have an operating system, there is nothing to interrupt the real-time execution. There are also many shields for attaching sensors and actuators to Arduinos as well as sensors designed to be connected to Arduinos. However, Arduinos are programmed in C and have very little RAM or computational power. That is where the Raspberry Pi comes in: it can be programmed in Python, has substantial RAM (along with SD card storage if needed), and has a much more powerful processor than the Arduino. Using the Raspberry Pi to log data from a test allows for plotting, data analysis, and instantaneous feedback on the robots' performance on a particular test.

Fall 2021 was the first year that students were allowed to write their code exclusively in Python using this real-time serial approach. When the cart/pendulum system was first introduced in Fall 2020, students were expected to write Arduino C code. A comparison of students' comments on the course evaluations between 2020 and 2021 regarding coding will be presented in the assessment portion of this paper.

Competition Description

The robotic cart/pendulum system was used in a junior level course on dynamic systems and feedback control as part of an end-of-semester competition. The competition included three events:

- Event 1: Vibration suppression of the pendulum
- Event 2: A line following race that included vibration suppression
- Event 3: Stabilization of the pendulum in the unstable position (vertically up)

Event 1 Description: Vibration Suppression

For the vibration suppression event, the instructor flicks the pendulum to start it swinging. The students send a command from Python over the serial connection to the Arduino to tell it to start the test. The students' code must collect some amount of data before switching on the vibration suppression control. The students' code must output a variable that indicates when the vibration suppression control is switched on. The students' earn a score for this event that depends on the settling time that will be measured from when the vibration suppression control turns on. Settling time will be determined by finding the largest encoder value after the vibrations suppression control turns on. The pendulum is considered settled when it stays within ± 4% of that largest encoder value. An example of an output graph for event 1 is shown in Figure 2.

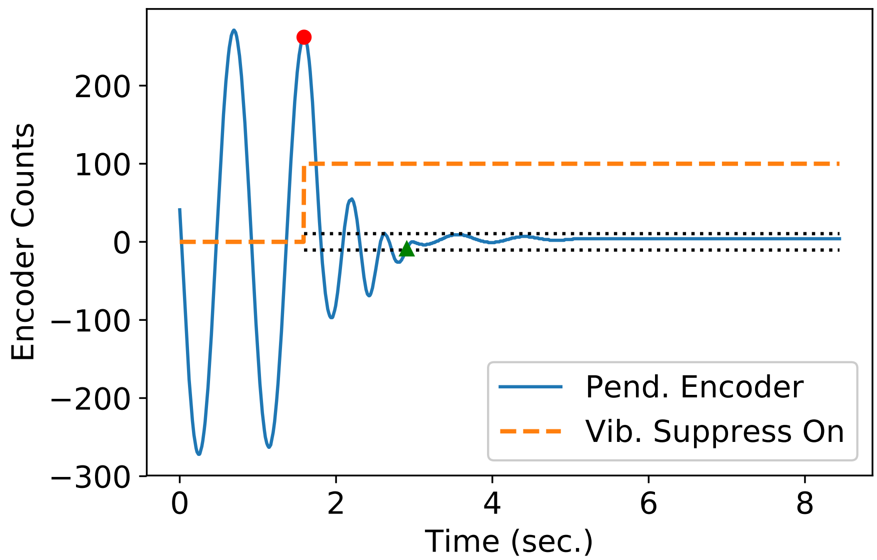

Figure 2: An example of the vibration suppression output graph for event 1.

Event 2: Vibration Suppression Plus Line Following

For the third event, the robot will race around an oval track, following the black painted line while also suppressing pendulum vibrations. The pendulum encoder counts must remain with ± 30 counts for the entire event (all 3 laps). If either wheel touches the line, that run is disqualified.

Event 3: Balancing the Pendulum in the Vertically Up Position While Completing as Many Laps as Possible

For this event, students must program the robot to balance the pendulum in the vertically up position for as long as possible. The event ends when the pendulum falls below horizontal.

Pedagogical Goals and Relationship to the Curriculum

This project takes place in a junior-level dynamic systems and controls course where the students are expected to learn to use the Laplace transform to predict the response of dynamic systems and they are also introduced to feedback control. The course places heavy emphasis on understanding the relationship between pole locations and system response. The course format consists of three credit hours of lecture and one credit hour of lab.

The students were first introduced to line following robots in a first-year cornerstone design project. In the first year course, most students try to implement line following control using an elaborate system of if/then statements. One of the smaller pedagogical goals of this project is to help students understand that feedback control, typically in the form of PD control, is more effective than the elaborate if/then statements they previously used and it is also much easier to code.

This project had three primary pedagogical goals:

1. to drive students deeper in their understanding of the connection between pole locations and the response of a dynamic system
2. to help students see how feedback alters the pole locations of a closed-loop system
3. to help students know how to convert a block diagram into Python code for running a real-time feedback control experiment

A smaller, secondary pedagogical goal was to help students appreciate that transfer functions and block diagrams apply to real-world systems and that the block diagrams can truly represent *any* dynamic system. When the project was initially assigned, some students responded as if they are starting from scratch and as if they have no idea how transfer functions and block diagrams apply to this new, complicated project.

The visible vibration of the pendulum seems to be helpful in deepening student understanding of feedback control. Students look at the vibrations of the pendulum and know that the direction the cart needs to move depends on the position of the pendulum and even on the velocity of the pendulum and which direction it is currently moving. But some of them did not immediately see the connection between these intuitive physical thoughts and PD control. These ideas led to a really important discussion in lab where students began to recognize that the feedback control logic represented in a block diagram (like the one shown in Figure 5) helps the system determine the timing and magnitude of the input needed to suppress the vibrations.

Competition Results and Assessment

The project was considered pedagogically successful for several reasons. First, all teams were able to successfully complete the first two events in the competition. For event 1, all teams were able to use feedback control to cause the vibration of the pendulum to die out faster then it would have otherwise. For event 2, all teams were able to complete three laps around the track without the pendulum moving beyond the established threshold. In order to achieve this, not only did student teams have to combined vibration suppression control with line following control, but they also had to consider how to program the robot to accelerate from rest without causing excessive initial vibrations.

For the third event, none of the teams were able to keep the pendulum upright for longer than 2 seconds. While this is mildly disappointing, event 3 is still very valuable from a pedagogical standpoint. It is impossible to use experimental system identification or PID tuning on an unstable system. So this event forces students to use a theoretical model and root locus control design.

Assessment of student learning on the project was done using three sources of data:

1. student project reports
2. a question on the lab practical exam related to block diagrams and Python code
3. comparison of performance between this year and a previous year on a final exam question related to pole locations
4. comparison of student comments on anonymous course evaluations between this year and last year to investigate how the real-time serial Python approach impacted student coding frustrations.

Final Exam Pole Location Question

A final exam question was designed to assess student understanding of the relationship between pole locations and the step response of a dynamic system. Students were asked to match the pole location on the left with the step response on the right in Figure 3. Conceptually similar versions of this question was used on the final exam in 2019 and 2021, where essentially only "the numbers were changed". The instructor does not return final exams to students so that the odds of students studying for the final based on previous exams should be low. Final exam scores on this question from 2019 and 2021 are compared in Figure 4. Note that 2019 was before the robot car/pendulum was introduced. The data does not lend itself to simple statistical analysis because the data is discrete and non-normal. The data is discrete because matching four pairs of things lead to scores of either 0, 25, 50, 75, or 100% (for some reason, the instructor gave "partial credit" in 2019 leading to some scores of 12.5%). The data is not normally distributed because most of the students performed well on this question. Still, the mean score on this question improved by 11.9% from 2019 to 2021.

Lab Practical Exam Question

A question on the lab practical exam was designed to assess whether or not students have learned the connections between the Python code they use to implement feedback control and the block diagram of the system. In this question, students where shown a block diagram of a feedback control system that included labels on various arrows, as shown in Figure 5. Students were then shown corresponding Python code (shown in Figure 6) and asked to match each arrow with the corresponding code.

A histogram of scores on this question during the 2021 lab practical is shown in Figure 7. The average score was 68.7%. While the scores on this question were somewhat disappointing, one of the purposes of the lab practical exam is to ferret out students who are allowing their lab partners to do too much of the work and, as a result, are not learning as much as they should in lab. This was the first year that this particular question was used on the lab practical exam, so no comparison to previous years is possible.

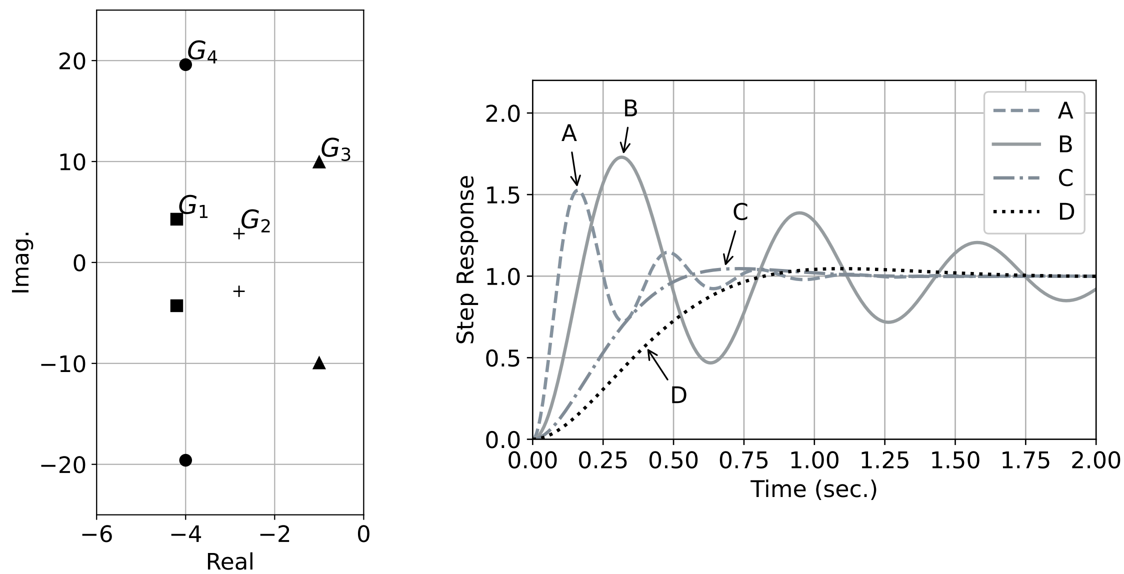

Figure 3: Final exam question for matching pole locations (left) to the corresponding step response (right)

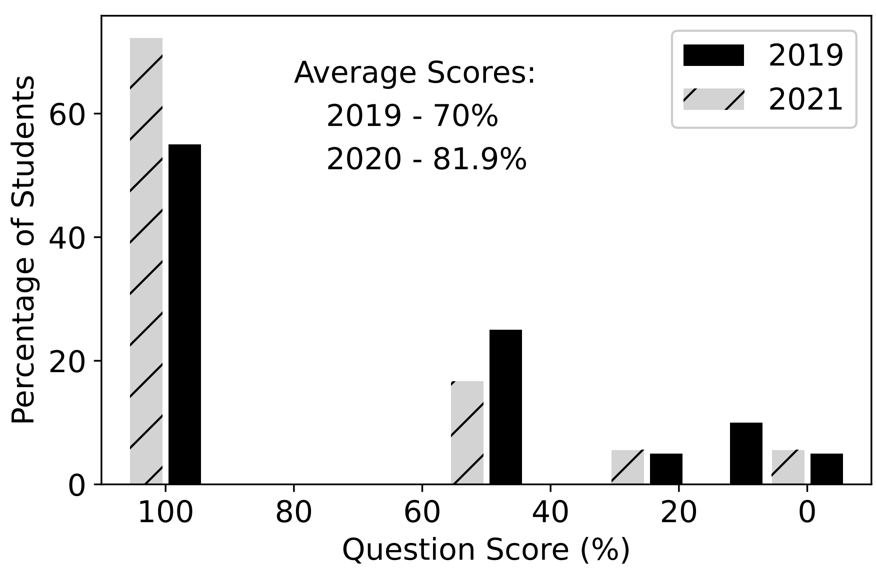

Figure 4: Histogram comparing results on a final exam question regarding pole locations and step response from the year before this project was introduced (2019) to this past year (2021)

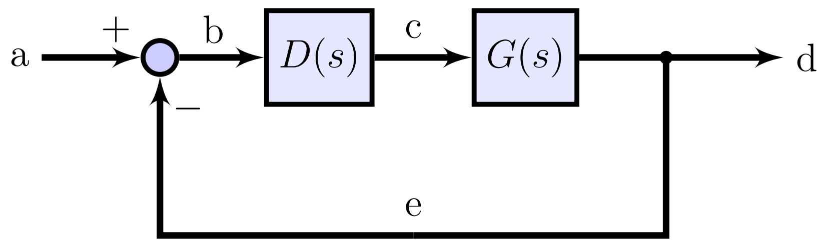

Figure 5: Block diagram with wire labels for the lab practical exam question matching Python code to the block diagram.

```python
for i in range(N):
    error[i] = u[i] - encoder[i-1]
    delta_e = error[i] - error[i-1]
    e_dot = delta_e/dt
    motor_speed[i] = kp*error[i] + kd*e_dot

    serial_utils.WriteByte(ser, 1)
    serial_utils.WriteInt(ser, i)
    serial_utils.WriteInt(ser, motor_speed[i])

    nvect[i] = serial_utils.Read_Two_Bytes(ser)
    t_ard[i] = serial_utils.Read_Two_Bytes(ser)
    u_echo[i] = serial_utils.Read_Two_Bytes_Twos_Comp(ser)
    encoder[i] = serial_utils.Read_Two_Bytes_Twos_Comp(ser)

    nl_check = serial_utils.Read_Byte(ser)
    assert nl_check == 10, "newline problem"
```

Figure 6: Code snippet for the lab practical exam question (match the lines of code with the block diagram wires shown in Figure 5)

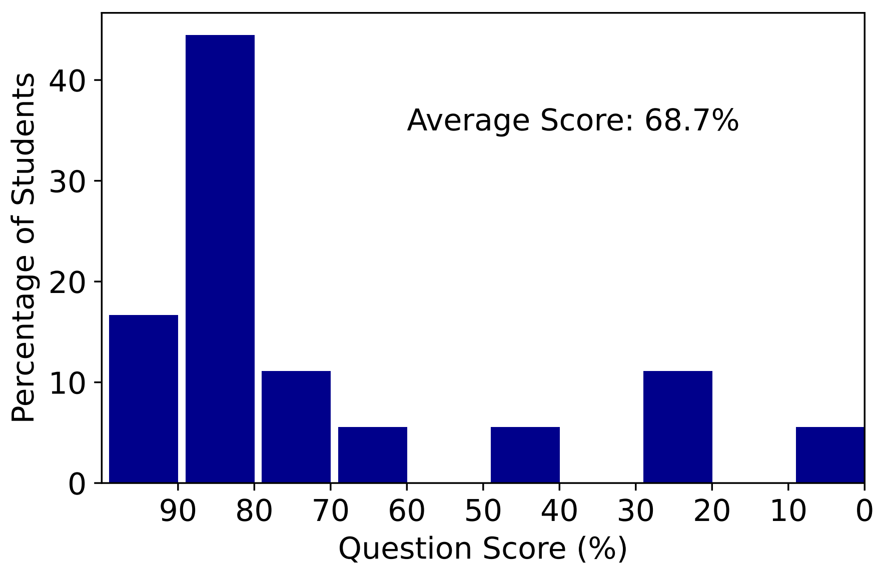

Figure 7: Histogram for student scores on the lab practical question where they needed to match the wire from Figure 5 with the corresponding lines of code from Figure 6

Observations from the Final Project Reports

Given the small number of team reports submitted for the project, no statistical analysis was attempted. However, students demonstrated their learning through their final project reports in several key ways. Most of the reports included a discussion of how the transfer function for the pendulum was found using system identification and a Bode plot. Most of the reports also included a discussion of how the root locus control design approach was used and the connection between pole locations on the root locus and the response of a closed loop system. Several of the reports also included a discussion between PD control and the physics of vibration suppression for the pendulum.

Comparison of Student Comments on the Final Course Evaluation Between 2020 and 2021

As mentioned previously, this was the first year where students programmed the robots entirely in Python without being required to write any Arduino code. In order to make this possible, Python and the Arduino Mega send small amounts of data back and forth in real time over USB serial, leading to this approach being dubbed real-time serial control. Student comments on the final course evaluations were compared between 2020 and 2021 to see if this change affected students' attitudes regarding coding in the class. Admittedly, the sample sizes are small and the analysis is not overly scientific. However, the trends are encouraging.

For the 2020 offering of the course, the words "code" or "coding" occur 18 times between student comments on the lecture and lab portions of the course. Roughly 11 of those were classified as complaints. Here are some representative quotes from 2020 student evaluations:

- "I personally have always had issues with understanding and working with code, python was much more user friendly but I can't code or keep up with some of the things you did in class, but being able to re-watch lectures helped."
- "If there was anyway where we could do testing of vibration suppression where the emphasis was on the understanding and not your ability to read and write code, that would be way better".
- "Any class that's sole focus is on understanding code is never easy."
- "That way we can understand the math of what you're doing without having to worry about if we know how to code".
- "Dr. X is a wizard when it comes to coding. This is both a great thing and a bad thing. It was very helpful when we were lost on code that he knew exactly the problem the majority of the time, or if he didn't he would help you figure it out. However, a lot of the time when you had a problem he wasn't direct with what you should change in the code so he doesn't have to tell you upfront. This was very irritating because of the amount of time it would take to solve a simple error in the code that could've been fixed if he would've simply told us, rather than making us think since most of the time it was a minimal aspect to the overall idea of learning. You can tell Dr. X relies a lot on python for his teaching, but in a class of X majors who all disliked the ridiculous amount of coding classes we had to take in lower level, emphasis should be on the subject and not on 'can you code' ".

- “Coding is not every [X major’s] bread and butter. Please understand we are not masters of coding by any stretch, and we need a helping hand.”

For the 2021 offering of the course, the words “code” or “coding” show up only 5 times in the student evaluation comments and only one of those would be categorized as negative (the student complained that lab starter code always required “adjustments”). Additionally, there was one strongly positive comment. Here are example quotes:

- “This was the first time I have ever coded in python so adding something to help establish a better knowledge base of the syntax might help early on. There were several assignments I struggled with just from a syntax side.”
- “There were always slight adjustments to given code” (referring to starter code)
- “really good class. only [university] coding class that made sense to me.”

Conclusion

This paper presented a novel robot that can be used for educational purposes in dynamic systems and feedback control courses. The robot has a pendulum with an encoder attached to a differential drive cart that includes a line sensor. The cart was used in a junior-level dynamic systems and control course as part of a final class project/competition. The competition consisted of three events: vibration suppression, line following with vibration suppression, and stabilizing the upward pendulum position. The robot uses the combination of a Raspberry Pi and an Arduino to provide a powerful, wireless, real-time feedback control environment. It costs roughly $300 to build one of these robots and they are entirely open source.

Student learning through the final project involving the robot was assessed in several ways, including exam questions, student project reports, and comparison of student evaluation comments. While the sample sizes are too small to provide statistical significance, the overall picture is that the project reinforces the course pedagogical goals and student learning is happening.

References

[1] Bernstein, D., “Enhancing undergraduate control education,” *Control Systems Magazine, IEEE*, Vol. 19, No. 5, oct 1999, pp. 40 –43.

[2] Bernstein, D., “Control experiments and what I learned from them: a personal journey,” *Control Systems Magazine, IEEE*, Vol. 18, No. 2, apr 1998, pp. 81–88.

[3] Shiakolas, P. and Piyabongkarn, D., “Development of a Real-Time Digital Control System with a Hardware-in-the-Loop Magnetic Levitation Device for Reinforcement of Controls Education,” *IEEE Transactions on Education*, Vol. 46, No. 1, 2003, pp. 79–87.

[4] Kamis, Z., Topcu, E., and Yuksel, I., “Computer-Aided Automatic Control Education With a Real-Time Development System,” *Computer Applications in Engineering Education*, Vol. 13, No. 3, 2005, pp. 181–191.

[5] Salzmann, C., Gillet, D., and Huguenin, P., "Introduction to Real-time Control using LabVIEW with an Application to Distance Learning," *Int. J. of Engineering Education*, Vol. 16, No. 5, 2000, pp. 372–384.

[6] Reck, R. M., "BYOE: Affordable and Portable Laboratory Kit for Controls Courses," *122nd ASEE Annual Conference and Exposition, 2015*, 2015, Paper ID: 13467.

[7] Schinstock, D., McGahee, K., and Smith, S., "Engaging students in control systems using a balancing robot in a mechatronics course," *2016 American Control Conference (ACC)*, IEEE, 2016, pp. 6658–6663.

[8] Bay, C. J. and Rasmussen, B. P., "Exploring controls education: A re-configurable ball and plate platform kit," *2016 American Control Conference (ACC)*, IEEE, 2016, pp. 6652–6657.

[9] Hill, R., "Hardware-based activities for flipping the system dynamics and control curriculum," *American Control Conference (ACC), 2015*, IEEE, 2015, pp. 2777–2782.