2016 American Control Conference (ACC)
Boston Marriott Copley Place
July 6-8, 2016. Boston, MA, USA

Combining Raspberry Pi and Arduino to Form a Low-Cost, Real-Time Autonomous Vehicle Platform

Ryan Krauss

I. Abstract

This paper presents a novel, low-cost autonomous vehicle platform based on the combination of a Raspberry Pi mini-computer, an Arduino micro-controller, and a Zumo track-driven robot chassis. The Arduino allows the control law to be executed at hard real-time intervals while the Raspberry Pi provides additional computing power, a web interface, and wireless data-streaming for control tuning and debugging. The efficacy of the platform in controls education is assessed after using the platform for demonstration purposes in a first course on feedback control. The entire system including the Raspberry Pi and Arduino Uno costs roughly $215.

II. Background and Literature Review

Feedback controls is a powerful and important subject that can be challenging for students to learn. Controls can come across as mathematically intensive and abstract. Experimental projects and demonstrations can increase students' interest in controls and expose them to the practical challenges of implementing control systems [1], [2]. Experimental projects can also help students develop necessary skills for building and debugging control systems. In a 2009 IEEE Controls System Society survey, 72% of industry respondents felt that "hands-on experience" is the area that most needs to be strengthened to better prepare control engineers for successful careers [3].

Setting up feedback control experiments can be challenging, especially if the control law needs to be executed at hard, real-time intervals. There are many commercially available real-time data acquisition and control solutions. Experiments based on these commercially available options have been used in controls courses and have been well-received by students [4], [5], [6].

Given the cost of engineering laboratories and the pressures on many university budgets, low-cost or student owned experiments have attracted some attention in the literature. Several researchers have presented ideas for low-cost or students-owned plants and investigated how to use these plants to enrich controls courses and facilitate different pedagogical techniques [7], [8], [9], [10].

Another challenge associated with experimental control systems is choosing the right hardware/software combination to allow students to program and implement the control law. Some researchers ask their students to program microcontrollers directly in C [11], [12], while others seek to insulate students from traditional programming entirely, relying instead on software that allows a high-level description of the control law using a block-diagram interface. There are commercially available systems for real-time control using a block-diagram interface. Much work has also been done on creating open-source, real-time systems [13], [14], [15].

Another hardware/software approach presented in the literature combines a microcontroller with a computer to create a real-time feedback control system [16], [17]. The computer in this approach does not need to be running a real-time operating system. The microcontroller enforces the real-time execution of the control law, reads all of the sensor signals, and sends all of the actuator signals while the computer provides a higher-level programming language and additional computational and memory resources.

This paper presents a low-cost platform for controls education that possesses some unique strengths:

- The total cost of the system including the robot chassis, motors, Raspberry Pi, and Arduino is $215.
- The Arduino/Raspberry Pi combination provides hard real-time control at update frequencies of 100-150Hz.
- The Raspberry Pi allows large amounts of data to be collected and wirelessly transmitted for each test.
- All of the software used is free and open-source.
- The user interface for the students can adjust from very simple to exposing all of the details of the integrated mechatronic system. Students in a first course on feedback controls can run tests using only the simple web interface.

III. Introduction to the Raspberry Pi + Arduino Approach

Raspberry Pi and Arduino are both popular and inexpensive boards for learning programming and allowing user code to interact with real-world systems through various sensors and actuators. While Raspberry Pi and Arduino are alike in size and cost, they are quite different in what they are and how they are used.

An Arduino is a microcontroller programmed in C. The primary way to interact with an Arduino is through a serial-to-USB cable that sends ASCII characters. The Arduino is a microcontroller and not truly a computer and that leads to several advantages and limitations. One of the primary advantages of a microcontroller is that it can easily execute tasks at hard real-time intervals based on a timer interrupt. An Arduino is also capable of interacting with many sensors and actuators directly through on-board analog-to-digital converters, PWM generators, and pin interrupts for fast encoder reading. An Arduino Uno has an 8-bit microprocessor running at 16 MHz with 32 KB of flash memory and 2 KB of SRAM.

In contrast, the Raspberry Pi is a mini-computer running a Linux operating system. The Raspberry Pi 2 has a quad-core 900 MHz processor and 1 GB of RAM. The Raspberry Pi is primarily programmed in Python. As a Linux computer, it is easy to remotely connect to a Raspberry Pi using SSH or VNC.

The Zumo Shield for Arduino is a small, track driven robot that is designed to plug directly into an Arduino as a shield. The robot is roughly 10cm by 10cm and costs roughly $100. The shield includes integrated motor drivers and a reflectance array used for line following applications. Combining the Zumo Shield with an Arduino allows students who understand basic Arduino programming to work with line following control directly on the Arduino. However, it is difficult to get large amounts of data from the Arduino to a computer wirelessly. Allowing students to view the sensor data vs. time is a key component in helping them learn to design and debug feedback control systems.

This paper presents an approach for combining a Raspberry Pi with an Arduino attached to a Zumo Shield to create a line following autonomous vehicle that returns large data sets for guiding student control design.

IV. System Description and Assembly

Figure 1 shows the autonomous vehicle platform being presented in this paper. The Arduino mounts upside down on top of the Zumo shield. The Zumo shield includes a reflectance array sensor used for line following. The Zumo robot is differential drive and the motor drivers are integrated into the Zumo shield. In order to assemble the platform as shown, two holes need to be drilled in the blade on the front in order to allow the cell phone battery charger to be zipped tied

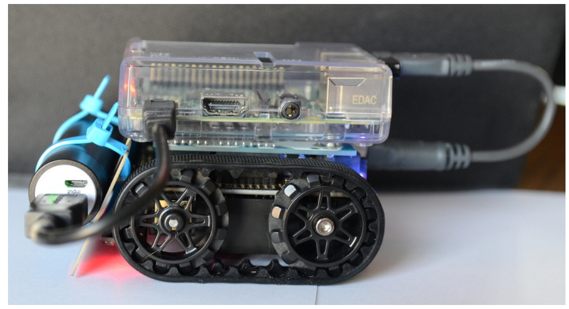

Fig. 1. The autonomous vehicle platform being presented in this paper: a Zumo robot controlled by the combination of a Raspberry Pi and an Arduino.

to the blade. The cell phone charger is used to power the Raspberry Pi via a USB cable. The Raspberry Pi is attached to the Arduino using Velcro®. The only other connection necessary is a USB cable from the Arduino to the Raspberry Pi for serial communication.

Tabel I lists the components of the Zumo autonomous vehicle platform and the cost for each at the time the paper was written.

TABLE I
COMPONENTS OF THE ZUMO AUTONOMOUS VEHICLE PLATFORM

Component	Price	Source
Zumo Robot Chassis (assembled with motors)	$99.95	Pololu
Raspberry Pi 2 (Rev. B)	$41.95	SparkFun
Arduino Uno	$24.95	SparkFun
USB cable A to B 6"	$4.99	Amazon
USB cable A to micro B 6”	$2.95	Amazon
16 GB micro SD card	$8.19	Amazon
Edimax Wifi Adapter	$9.99	Amazon
RPi case (optional)	$8.95	Amazon
Anker Battery (optional)	$9.99	Amazon
AA batteries	$2.00	Aldi
total	$213.91	

V. Real-Time Control Approach

Figure 2 shows a block diagram of the Raspberry Pi and Arduino working together to create a versatile real-time feedback control system. Either device could probably provide a decent control system by itself, but the combination of the two provides some really useful features. There are two primary drawbacks to using only the Arduino. First, the students would have to learn to program the Arduino directly in C. If the line following robot is simply being used to supplement a traditional first course in automatic controls, this level

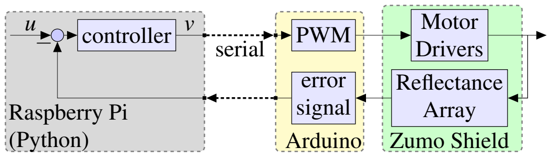

Fig. 2. Block diagram showing the Raspberry Pi and Arduino working together to create a real-time feedback control system.

of microcontroller programming may be too much. The bigger issue is that using the Arduino by itself makes it very difficult to stream data back to a computer for control tuning, debugging, and other educational purposes. The addition of the Raspberry Pi solves these problems by providing an easy-to-use web interface and wifi connections via SSH or VNC.

Alternatively, it may also be possible to control the Zumo robot using only a Raspberry Pi. However, the Zumo chassis is designed as an Arduino shield. This means that the electrical connecitons assume the pinout of the Arduino and the example code for reading the sensor array and driving the motors is written for the Arduino.

The synergistic combination of a Raspberry Pi and an Arduino allows the Arduino to control the real-time execution of the control law, read the sensor signals, and send the PWM signal to the motors. The Raspberry Pi provides the web interface, the wifi connection, additional computational power and RAM, and captures data for tuning and educational purposes. The primary limitation of this approach is that the USB-to-serial connection limits the digital control frequency to 100-150 Hz.

VI. Source Code

The source code for this project is available on github: https://github.com/ryanGT/zumo_serial. The main Arduino file is line_follower_RT/line_follower_RT.ino and the main Raspberry Pi file is cherrypy_krauss_1.py. Running the Python file on the Raspberry Pi starts the web interface that is discussed in the next section.

VII. Web Interface

If the robot is being used in a mechatronics course or an advanced digital control course, students could program the Arduino and/or the Raspberry Pi directly. However, if the platform is being used in a first course on automatic controls, it may be desirable to insulate the students from microcontroller coding. To that end, a simplified web interface has been created. Figure 3 shows a screen-shot of the web interface for running PID tuning tests. In order to run a test, the students

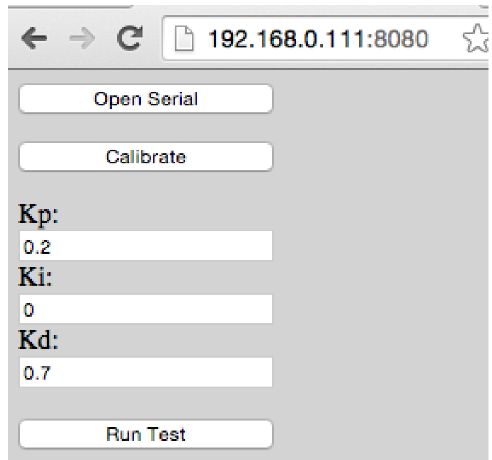

Fig. 3. Screen-shot of the simple web interface for running a test.

first press the Open Serial button to initialize the USB-to-serial connection between the Raspberry Pi and the Arduino. Once the serial connection is established, the students go back to the main page and press the Calibrate button. This button executes the calibration routine for the reflectance array line sensor. During the calibration, the robot rotates in a circle and records values for the sensor array as it passes over the black line and white background. The code for the calibration routine is provided by Pololu, the maker of the Zumo shield.

Once calibration is done, the students can input the PID gains and press the Run Test button. The robot then drives around the track for one lap. At the conclusion of the test, a graph of the reflectance array sensor error is shown along with some summary data that includes the lap time and the total error from the reflectance array for the lap. A screen-shot from a completed test is shown in Figure 4.

VIII. Lecture Demonstrations

So far, the robot platform has been used in two lecture demonstrations as part of a first course on feedback controls. The first demonstration was designed to show how feedback control can help systems reject disturbance inputs. The robot was tasked with driving straight down a board that had been deliberately tilted. The control system would have to compensate for the lateral pull of the component of gravity acting down the slope of the roadway. The Zumo robot is shown on the laterally inclined roadway in Figure 5. In the first part of the demonstration, the robot is simply commanded to send the same voltage to both motors under open-loop control. Predictably, the lateral pull of gravity causes the robot to veer off the road. The reflectance array is used

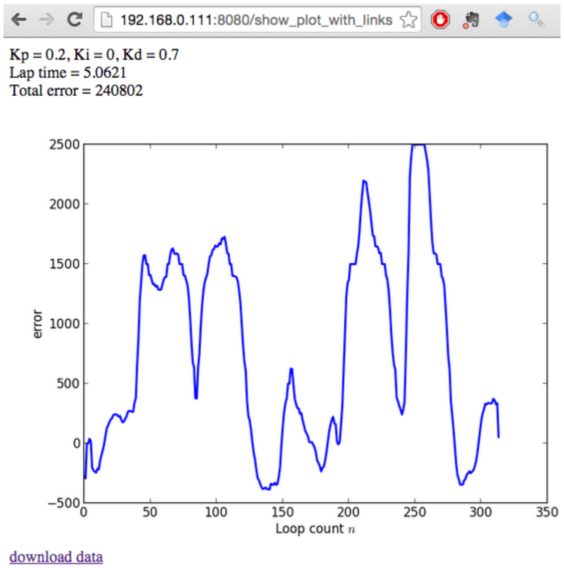

Fig. 4. Screen-shot from a completed test using the web interface. Multiplying the loop count n by the digital control time step *dt* gives time in seconds. The error is in counts and measures the line position on the line sensor reflectance array.

Fig. 5. Rear view of the Zumo robot on a roadway with a lateral incline.

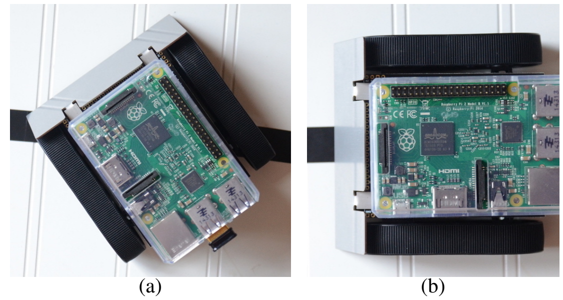

Fig. 6. Starting and stopping points for the initial condition rotation test.

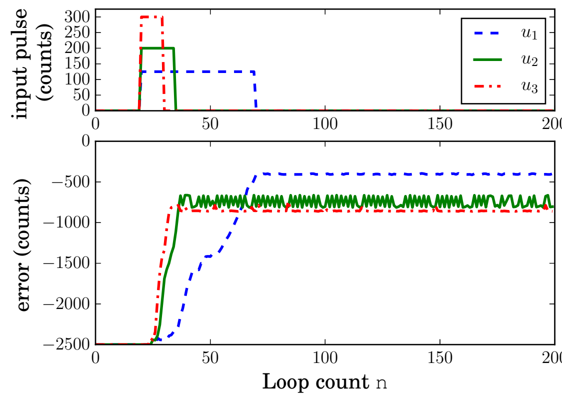

Fig. 7. Open-loop test results for the initial condition rotation test. Loop count n times *dt* gives time in seconds. The error equals zero when the line is in the center of the reflectance array.

for feedback control in the second part of the demonstration and the robot is able to easily stay on the roadway to the end. A video of this demonstration was created and posted online: https://youtu.be/rNm-AMTSUzU

In a second lecture demonstration, the robot was used to show that feedback can alter pole locations and affect the transient response of a system. The robot was programmed only to rotate in place (i.e. no forward velocity) and was placed so that the line was off to the far corner of the reflectance array sensor. The starting position for this demonstration is shown in Figure 6 (a). The desired stopping point with the line near the center of the reflectance array is shown in Figure 6 (b). For the open-loop part of the demonstration, different amplitude pulses were given to the motor drivers (amplitude is in counts and 400 counts corresponds to full speed). The widths of the pulses were tuned experimentally to get the robot to finish the test with the line near the center of the reflectance array. Open-loop results are plotted in Figure 7.

Closed-loop results for the initial condition rotation test are shown in Figure 8. During the demonstration, students were asked what aspects of the response were changing as the proportional gain K_p was changed. The connection between damping ratio and closed-loop pole locations was discussed.

The final use of the Zumo robot platform will be in a student control design competition later this semester (probably Nov. 2015). Student teams will compete to see who can minimize tracking error and lap time simultaneously while the Zumo robot races around a small line-following track.

IX. Assessment Procedure and Results

The Zumo robot was incorporated into a series of lectures that teach students the potential benefits and the risks of using feedback control. The effectiveness

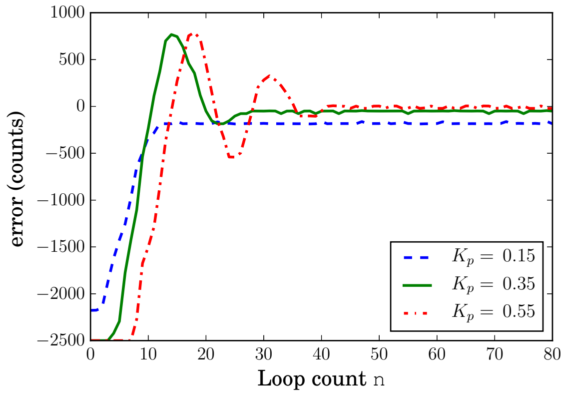

Fig. 8. Closed-loop test results for the initial condition rotation test.

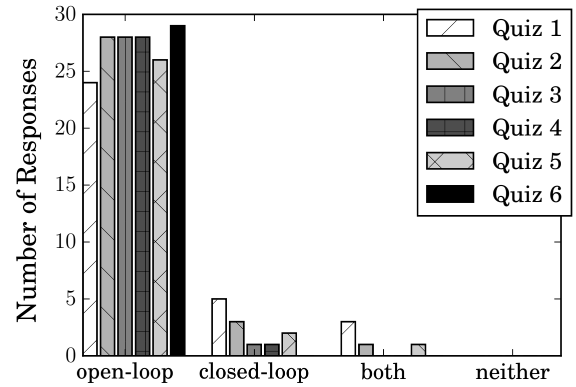

Fig. 9. Assessment results for pre/post quiz question 2: Which control approach is affected more by disturbance inputs?

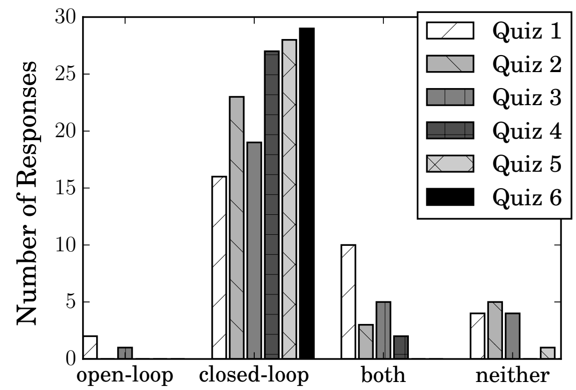

Fig. 10. Assessment results for pre/post quiz question 3: Which control approach can change the damping coefficient of a system?

of the demonstrations and lectures was assessed using a series of pre/post multiple choice quizzes. Quizzes were administered at the beginning and end of three lectures. Two of the lectures included demonstrations involving the Zumo robot while the third lecture did not; it was a traditional lecture/discussion. Quizzes 1 and 2 were given before and after the lecture containing the disturbance rejection demonstration. It is worth noting that the disturbance rejection video that was posted online was made available to the students after the first lecture demonstration (in between quizzes 2 and 3). Quizzes 3 and 4 were given before and after the lecture on how feedback can affect the transient response of a system. Quizzes 5 and 6 were given before and after the traditional lecture that focused on teaching students that because feedback control can alter system pole locations it can also drive stable systems unstable. It should be noted that 33 students completed quizzes 1 and 2 while only 29 completed quizzes 3-6. So, for questions that involve all 6 quizzes, it may be more meaningful to track the reduction in wrong answers than the increase in correct answers.

Figure 9 shows the results for quiz question 2: Which control approach is affected more by disturbance inputs? While most students correctly guessed that disturbance inputs are a bigger problem for open-loop systems even before the demonstration, there is still an increase in the number of correct answers after the demonstration (from quiz 1 to quiz 2). A YouTube video was also posted after the lecture and some of the students watched the video before the next lecture. This may explain a reduction in the number of wrong answers between quiz 2 and quiz 3.

Figure 10 shows the results for quiz question 3: Which control approach can change the damping coefficient of a system? There is a significant jump in the number of correct responses before and after the lecture and demonstration on this topic (from quiz 3 to quiz 4).

Figure 11 shows the results for quiz question 5: Which approach can change the poles of the system? There is a significant increase in the number of correct responses before and after the second lecture/demo covering the effects of feedback on transient response.

Figure 12 shows the results for quiz question 7: Which approach can drive a system unstable, i.e. cause a stable system to become unstable? The responses to this question show a considerable amount of confusion until the final lecture that focuses on the connection between pole locations and system stability.

X. Conclusions

A low-cost platform for autonomous vehicle research and feedback controls education has been presented. The platform costs $215 and is controlled by an Arduino Uno and Raspberry Pi working together. The system is capable of hard real-time control with update frequencies

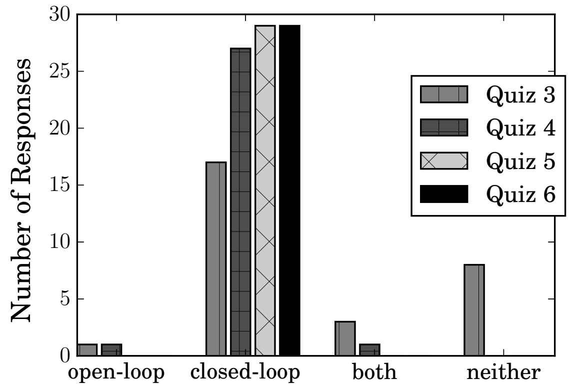

Fig. 11. Assessment results for pre/post quiz question 5: Which approach can change the poles of the system?

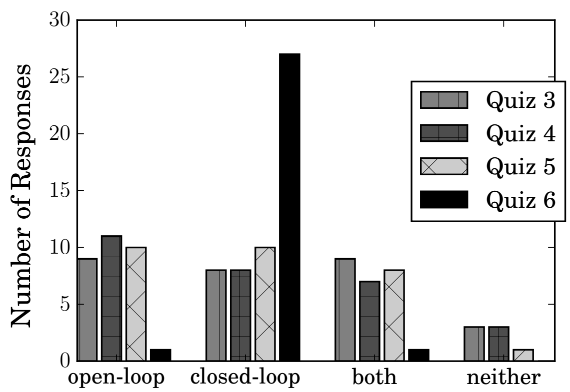

Fig. 12. Assessment results for pre/post quiz question 7: Which approach can drive a system unstable, i.e. cause a stable system to become unstable?

of 100-150 Hz. The system uses only free, open-source software. Students can run experimental tests using the platform through a simple web interface without having to program the Arduino or the Raspberry Pi. At the conclusion of a test, students can download the real-time data for the test. The system has been used in two lecture demonstrations during a first course on feedback controls. Assessment data from pre/post quizzes shows that students are learning important feedback controls concepts as a result of the lectures and demonstrations.

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] Cook, J. A. and Samad, T., "Controls curriculum survey: A CSS outreach task force report," Nov. 2009.

[4] 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.

[5] 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.

[6] 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.

[7] Reck, R. M. and Sreenivas, R. S., "Developing an Affordable Laboratory Kit for Undergraduate Controls Education," *ASME Dynamic Systems and Controls Conference, 2014. Proceedings of the 2014*, ASME, Oct. 2014.

[8] Reck, R. M. and Sreenivas, R. S., "Developing a new affordable DC motor laboratory kit for an existing undergraduate controls course," *American Control Conference (ACC), 2015*, IEEE, 2015, pp. 2801–2806.

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

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

[11] Jack, H. and Blauch, A., "A Modeling and Controls Course using Microcontrollers," *Proceedings of the 2004 ASEE Annual Conference and Exposition*, 2004.

[12] Jack, H. and Blauch, A., "A Modeling and Controls Course using Microcontrollers," *Proceedings of the 2005 ASEE Annual Conference and Exposition*, 2005.

[13] Bucher, R. and Dozio, L., "CACSD under RTAI Linux with RTAI-LAB," *Fifth Real-Time Linux Workshop, Valencia, Spain*, 2003.

[14] Dozio, L. and Mantegazza, P., "Real time distributed control systems using RTAI," *Object-Oriented Real-Time Distributed Computing, 2003. Sixth IEEE International Symposium on*, IEEE, 2003, pp. 11–18.

[15] Bucher, R. and Balemi, S., "Scilab/Scicos and Linux RTAI - A unified approach," *Control Applications, 2005. CCA 2005. Proceedings of 2005 IEEE Conference on*, IEEE, 2005, pp. 1121–1126.

[16] Krauss, R. W. and Croxell, J. R., "A Low-Cost Microcontroller-in-the-Loop Platform for Controls Education," *American Control Conference, 2012. Proceedings of the 2012*, IEEE, 2012.

[17] Krauss, R. W., "Evaluation of a Low-Cost Microcontroller for Real-Time Control Education and Prototyping," *ASME Dynamic Systems and Controls Conference, 2014. Proceedings of the 2014*, ASME, Oct. 2014.