# A Networked Cyber-Physical System Testbed for Undergraduate Education

Erick J. Rodríguez-Seda, Paul J. Frontera, and Joseph Bradshaw

***Abstract*—This paper presents a networked cyber-physical system (CPS) testbed used in an undergraduate course project-based learning activity. The low-cost, portable testbed can be used in wide-ranging activities to include exploration of modeling and simulation, controls, communications, networks, embedded systems, and cyber security. The activity presented focuses on a subset of this list, providing students with an opportunity to demonstrate understanding of course lecture material while working in a small team.**

## I. Introduction

Colloquially, the term Cyber-Physical System (CPS) refers to a system that integrates computing, communication, and control technologies to regulate the performance of a physical process [1]. As shown in Fig. 1, a typical CPS is comprised of multiple spatially distributed nodes (e.g., sensors, actuators, computers, and controllers) that share information (e.g., commands and measurement signals) with the aim of regulating some physical quantity. When different nodes are integrated via shared communication networks, the system is often referred as a networked CPS.

The development and operation of networked CPSs demands trained engineers and computer scientists that can deal with the complex and interdisciplinary nature of the systems. Personnel with knowledge of physics, modeling, controls, computing, and communications as well as an understanding of the synergy among such different domains are required [2]–[4]. Several institutions and academics have developed curricula, courses, and programs in response to this need [5]–[10]. The United States Naval Academy (USNA) Cyber Science Department began offering the Cyber Operations undergraduate major in 2013 in collaboration with several departments including Computer Science, Mathematics, Electrical and Computer Engineering, and Weapons and Systems Engineering departments. The Cyber Operations major provides students with a technical and non-technical foundation in diverse areas of computing (e.g., programming, data structures, information assurance, and cryptography) and engineering (e.g., computer architecture, communications, and controls).

The Cyber Operations major requires students to take an introductory course on the control of CPSs, SY202 Cyber Systems Engineering, during their second year. The course focuses on the design, control, and analysis of CPSs as well as the relationships and interactions between the cyberspace and physical domains. SY202 integrates lectures, current events, real-world examples, and hands-on projects to provide students a fundamental understanding of CPSs they will encounter while serving as officers in the U.S. Navy and Marine Corps. This paper presents the course's final hands-on activity and describes the experimental networked CPS testbed used to facilitate project-based learning. The project covers topics on communication networks, control systems, modeling of physical systems, performance evaluation, data collection, vulnerabilities assessment, launch of cyber attacks, and the effect of cyber attacks and disturbances on the closed-loop system operation. In contrast to other similar CPS examples [6], the proposed testbed allows students to explore the effects of both the cyberspace and physical domains by means of a real-world mechanical system. Other features of the testbed include portability, low-cost, and modularity; which makes the testbed suitable for any type of classroom.

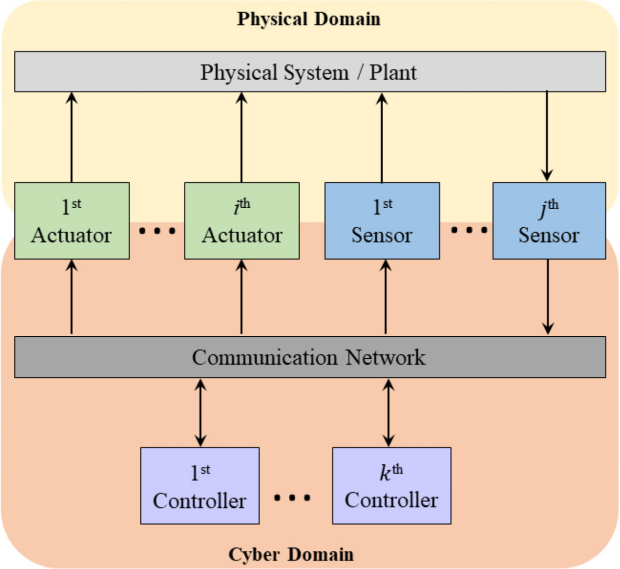


Fig. 1. Networked CPS.

The paper is organized as follows. Section II presents a background of the course providing the context for the testbed project-based learning activity. Section III describes the experimental testbed. Section IV describes a series of project-based activities for students using the testbed. Finally, section V and VI discuss several other potential project-based activities to explore using the testbed along with some

This work was supported in part by USNA's Center for Cyber Security Studies under Gift Fund 87050.

E. J. Rodríguez-Seda, P. J. Frontera, and J. Bradshaw are with the Department of Weapons and Systems Engineering at the United States Naval Academy, Annapolis, MD. {rodrigue, frontera, bradshaw}@usna.edu

lessons and recommendations from the authors' experience.

## II. Cyber Systems Engineering Course

The Cyber Systems Engineering course introduces USNA Cyber Operation major students to the design, synthesis, control, and study of CPSs, the interconnections and communications between those systems, and the vulnerabilities and cyber threats they face. The course blends engineering principles with cyber fundamentals, providing students with a comprehensive and integrative teaching approach. At the end of the course, students should have an overall understanding of the composition and function of networked CPSs and the interplay between the cyberspace and physical domains.

### *A. Course Learning Outcomes*

A CPS course is interdisciplinary by nature and, as such, must synergistically create a balance between the learning objectives of a computer science program with those of an engineering one [8], [11]. ABET, a global accreditor of university programs in computing and engineering, has recently released proposed accreditation criteria for Cyber Security programs taking into account the computer science and engineering nature of the discipline.[1] Similarly, the Cyber Physical Systems Virtual Organization (CPS-VO), an alliance among academic and industry leaders created to facilitate the development of CPS sciences, has developed its own criteria.[2] Based on both criteria, the course adapted the following learning outcomes:

- an ability to analyze a problem, and to identify and define the computing requirements appropriate to its solution;
- an ability to design, implement, and evaluate a computer-based solution to meet a given set of computing requirements in the context of the discipline;
- an ability to communicate effectively with a range of audiences about technical information;
- an ability to analyze and evaluate systems with respect to maintaining operations in the presence of risks and threats.
- an ability to design and conduct simulations and tests of a CPS and to analyze the results; and
- an ability to understand how design decisions in the cyber domain affect the physical domain and vice versa.

The first four learning outcomes are based on ABET's criteria while the last two are taken from CPS-VO's recommendations.

### *B. Course Description*

The course format consists of two 50 minutes lecture and one 110 minutes hands-on laboratory session per week for a total of 16 weeks. It assumes students to have basic knowledge of physics and calculus as well as a general understanding of programming languages, including C and C++. Intentionally, the course does not require knowledge of dynamics, electrical engineering, or differential equations–topics typically included in engineering majors but omitted by most computer sciences disciplines.

The course is roughly divided in four parts. The first part focuses on the physical realm of CPS. It covers modeling of mechanical systems, discussion of transfer functions, stability, and a general understanding of the time response of first and second order systems. The second part centers on computing and control. It discusses the use of microcontrollers, actuators, and sensors, as well as the implementation of Proportional-Integral-Derivative (PID) controllers. The third part focuses on communication, with special emphasis on serial communication and networks. The fourth part centers on design considerations, cyber treats, and large-scaled CPSs. A complete list of topics and the lecture time spent on them is given in Table I. Project-based learning activities conducted during a 50 minute lecture are not included in this table.

TABLE I

Cyber Systems Engineering Topics

| **Topic** | **Lecture Hours** |
|---|---|
| CPSs and Mechatronics | 1 |
| Modeling of Mechanical Systems | 4 |
| Laplace Transform and Transfer Functions | 2 |
| Time System Response and Stability | 2 |
| Embedded Systems and Microcontrollers | 2 |
| Actuators | 2 |
| Sensors | 1 |
| PID Control | 2 |
| Serial Communications | 2 |
| Controller Area Network (CAN) | 1 |
| Industrial Control Systems and SCADA Systems | 1 |
| Vulnerabilities and Cyber Attack Models | 2 |
| Resilient Control | 1 |

Each week in the duration of the course has a project-based learning activity of 110 minutes where students can apply, either via simulation or hardware, material learned in the preceding weeks. The work and material is cumulative and aims to educate students on tools necessary for the the design, implementation, validation, and analysis of CPSs. The course culminates with an 10-hour hands-on project. The final course project presented in the following sections is designed to effectively assess all learning outcomes listed in section II-A.

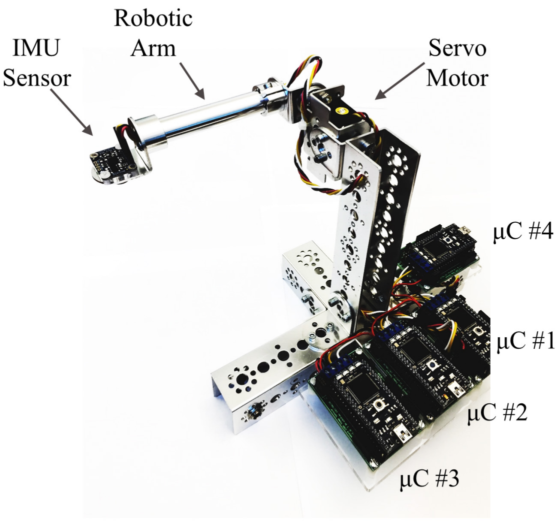
## III. Low-Cost Networked CPS Testbed

A comprehensive curriculum on CPS education must expose students to hands-on experience [4]–[6]. However, hands-on laboratory sessions can be challlengined to implement due to costs, space limitations, and the unreliability or uncertain nature of equipment [5]. This paper presents a networked CPS testbed used as the final project of USNA's Cyber Systems Engineering course that is compact in size, easy to transport, easy to operate, and relatively inexpensive. Despite its simplicity, the testbed provides a general overview of all components and their interactions within a CPS.

The networked CPS testbed, illustrated in Fig. 2, consists of a single-link robotic arm (a cylindric-shaped link) mounted on a aluminum frame, an inertial measurement

[1] http://www.abet.org/blog/news/abet-seeks-feedback-on-proposed-accreditation-criteria-for-cybersecurity-academic-programs/

[2] https://cps-vo.org/group/edu/workshop



t!]

Fig. 2. Networked CPS Testbed.

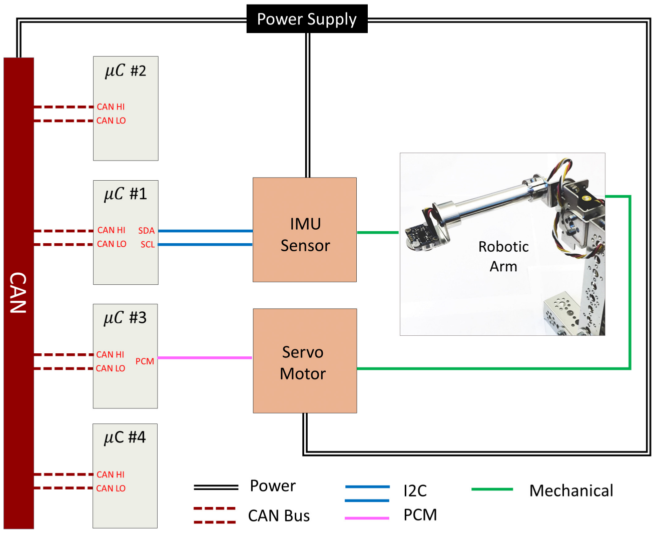


Fig. 3. Networked CPS Functional Block Diagram.

unit (IMU), a servo motor, a set of four microcontrollers ($\mu$C) mounted on Controller Area Network (CAN) Bus-ready printed circuit boards (PCB), and a 12 volts DC adapter to power all electronics. The robotic arm represents the plant and the angular position of the arm is the physical process to be regulated by the CPS. The arm along with the supporting cross-shaped frame are built using a Tetrix® Kit. The kit comes with modular segments of different shapes that can be easily assembled and cut to form the desired supporting structure.

The proposed testbed has a Bosch BNO055 IMU (sensor) mounted at the free end of the robotic arm. The sensor integrates an accelerometer, a gyroscope, and a geomagnetometer in a single package to provide angular position information about the arm. The BNO055 sensor calibrates itself when first powered and can provide angular position in radians to a $\mu$C using the Inter-Integrated Circuit (I$^2$C) communication protocol. Other position or orientation sensors could be employed in lieu of the Bosch BNO055. The BNO055 along with all other electric and mechanical parts used to build the testbed were chosen based on availability within USNA's Technical Support Department.

The testbed employs a Hitech HS-475HB servo motor to actuate the arm and regulate its position. The servo motor has a rotational range of approximately $\pm\pi/2$ radians ($\pm 90\,\text{deg}$) and accepts a Pulse Code Modulated (PCM) voltage signal as input.

For processing and computation, the testbed uses a total of four mbed LPC1768 microcontrollers mounted on CAN Bus-ready PCB developed at USNA. Each $\mu$C (refer to Fig. 2) has a different role within the network.

The first $\mu$C collects and disseminates data from the sensor. Angular position data in radians is read from the IMU using the I$^2$C protocol and then writen to a CAN message providing the measured angular position of the arm every 0.05 s. The CAN message is accessible to all other $\mu$Cs connected to the bus. The second $\mu$C acts as the controller or decision unit; it reads CAN messages coming from the sensor, determines arm position error, computes the control signal for the actuator (i.e., servo motor), and writes a CAN message with the control signal. The third $\mu$C regulates the position of the arm by reading control signal messages from the CAN bus and passing a PCM signal to the servo according to the message received. Finally, the fourth $\mu$C emulates any other device with access to the CAN network, e.g. another sensor or $\mu$C controlling a different process. In this project, the fourth $\mu$C acts as a malicious agent that injects false data into the system by posing as the IMU sensor with the aim of disrupting the closed-loop system performance. Fig. 3 is the functional block diagram of the overall testbed along with the connection for each $\mu$C.

As noted previously, all $\mu$Cs are interconnected using a CAN network. CAN, a serial communication protocol originally developed for the automobile industry, has been adopted for multiple control applications and sectors due to its low cost and compatibility with multiple off-the-shelf devices [12]. All nodes or $\mu$Cs with access to the network can read and write messages onto the bus. Each message carries the identification number of the sender along with the data. The identification number of the sender determines what a node (i.e. a $\mu$C) does with the data in each message. Because CAN lacks participant authentication, a malicious node can spoof data in the CAN bus by using another node's identification number [13].

Overall, the testbed is convenient for hands-on instruction of networked CPSs. It is small in size and weight, only measuring 25 cm long, 20 cm wide, and 25 cm tall with the robotic arm extended and weighing about 4 kg. Its compact size makes the testbed highly portable, allowing the students to take the CPS anywhere, including outside of the classroom. In addition, it provides modularity as other $\mu$C and devices can be easily added to the CAN network. Finally, the prototype testbed is relatively inexpensive, costing about

$420 per unit (refer to Table II for a detailed budget). Use of cheaper alternatives, specially other $\mu$Cs, can further reduce the cost.

TABLE II
LIST OF PARTS AND COSTS

| **Part** | **Cost per unit**[a] | Quantity | Total Cost |
|---|---|---|---|
| Tetrix Parts | $40 | 1 | $40 |
| BNO055 Sensor | $30 | 1 | $30 |
| Hitech HS-475HB | $20 | 1 | $20 |
| mbed LPC1768 | $60 | 4 | $240 |
| CAN PCB | $20 | 4 | $80 |
| Power Adapter | $10 | 1 | $10 |
| Total | – | – | $420 |

[a] Costs are estimated based on current market prices and rounded up to the nearest tens.

## IV. Project-based Learning Activity

Students are presented with the networked CPS testbed described in section III to engage in 10-hour project-based learning activity near the end of the course. The activity is performed in teams of three students and seeks for the students to obtain the following specific learning outcomes:

- To put in practice all concepts discussed throughout the course to include:
  - the design of closed-loop systems
  - the use of actuators, sensors, and microcontrollers to regulate a physical process
  - the use of serial communication and communication networks
  - the assessment of system performance
  - the design and launch of cyber attacks on control systems
- To identify and understand the hardware necessary to control and sense within a simple CPS, in this case, a single-link robotic arm
- To programm a $\mu$C for the design of control systems
- To learn how to build and use CAN in a control system
- To evaluate via experiments the effects of external disturbances and cyber threats on the closed-loop performance of the system

These project specific learning outcomes facilitate the attainment of course learning outcomes presented in section II.

The project is introduced through a MATLAB Simulink simulation of the testbed project-based learning event. This graphic simulation introduces the students to the interconnections within the testbed CAN bus network and system response during nominal operation, in the presence of a disturbance, and to a cyber attack. The students are provided with a library of blocks that mimic the functions performed by each $\mu$C and the CAN bus communication. The students combine these blocks as shown in Fig. 4.

This introductory activity provides the students with an understanding of the testbed components in an environment free from potential hardware malfunctions or programming implementation errors. System response is studied both with and without sensor noise to emphasize the challenges associated with actual systems. Students are asked to determine the proportional and integral gains required to meet required performance specifications. Fig. 5, the system response of the robotic arm, allows students to synthesize the course lecture material for future comparison with experimental results. Students producing this plot observe the effect that a disturbance has on their system and how the controller should respond. Additionally, by simulating an exaggerated noisy measurement, students are introduced to the challenges that exist in CPS implementation.

Students transition to the testbed immediately upon completion of the simulation portion of the project. Students are provided with functioning microcontroller binary files for the actuator ($\mu$C #3) and sensor ($\mu$C #1) nodes. This code is provided so that students will focus on writing the code required to produce a working controller to position the robotic arm and the associated cyber attack on the controller. Additionally, students are provided with a rough outline of the code necessary for reading and writing CAN bus messages.

The hands-on activity is decomposed into two major sections: design, implement, and study the system response of a Proportional-Integral (PI) controller code on $\mu$C #2 and design, implement, and study the system response of a stealth cyber attack on $\mu$C #4. The first major section is further decomposed into defined smaller increments as follows:

- Read sensor data from the CAN bus
- Parse sensor data
- Write data to the CAN bus
- Command the robotic arm to a specified angle
- Implement a PI controller to meet specified requirements

The authors discovered that students benefited from this decomposition; groups attempting to immediately implement the PI controller code without verifying functionality of the preceding steps invariably failed on their first attempt.

After implementing the PI controller, students study the system response, both nominal and with a disturbance, to compare their testbed results with those obtained through simulation. Figure 6 is an example of the response obtained from the testbed. This reflection reinforces the value of conducting simulation prior to hardware implementation.

The project-based learning activity has the students design a stealth cyber attack [14], [15], implement the attack, and study the system response of the testbed. Of note, this activity was not decomposed as the first section. The authors found that students took cues from the implementation of the PI controller and decomposed the task into smaller portions on their own. Once implemented, students were directed to vary the frequency and magnitude of the attack to gain insight into its effect on the testbed. Figure 7 provides an example of the testbed response to a stealth attack of 0.02 rad occurring every 0.5 seconds.

Students are required to submit a team-written technical report that captured their experiences. This report facilitates

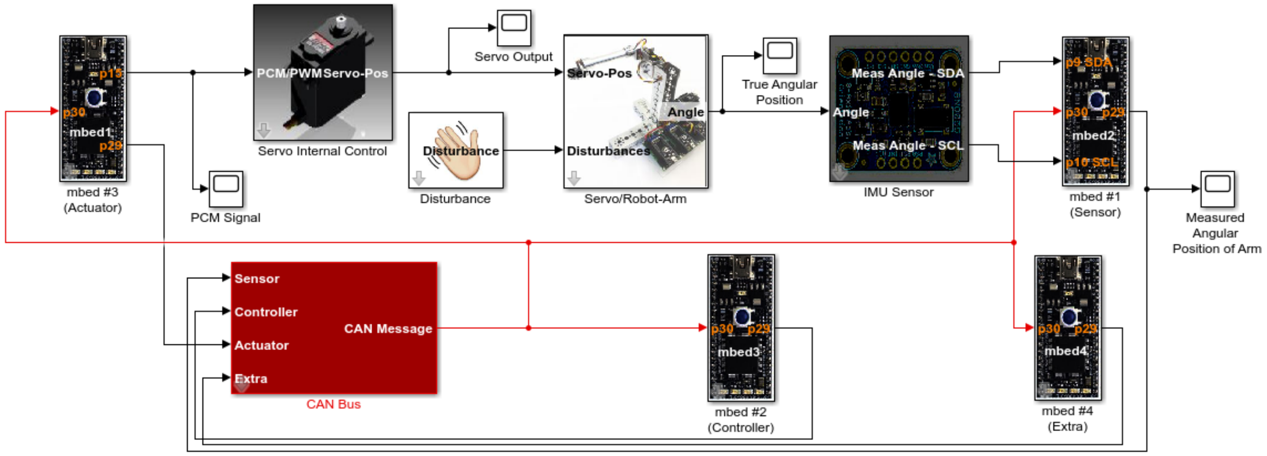


Fig. 4. Simulation Diagram of Networked CPS in MATLAB Simulink

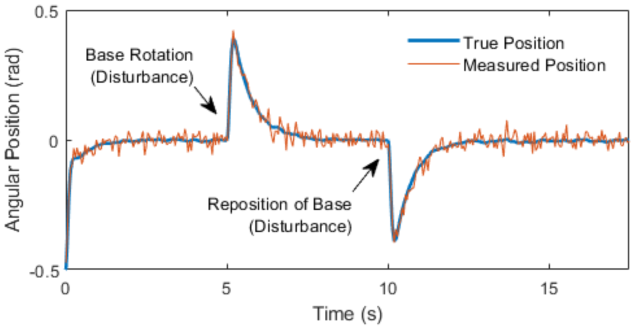


Fig. 5. Simulation of System Response with Disturbance between 5 and 10 s.

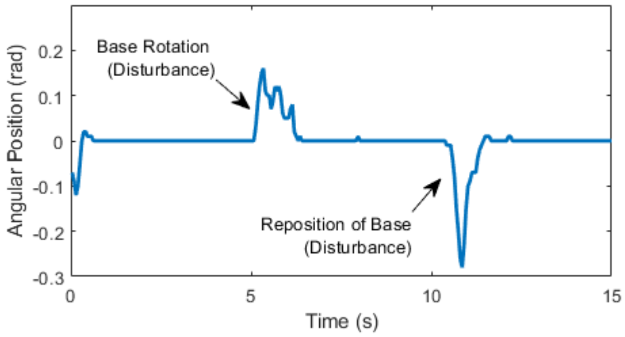


Fig. 6. Experimental System Response with Disturbance between approximately 5 and 10 s.

instructor assessment of the team's progress on meeting course learning outcomes and provides the students an opportunity to formally reflect upon the activity.

## V. Additional Project Extensions

The presented testbed can be used in a wide variety of project-based learning activities. Examples include investigating:

- Ethernet connectivity of multiple testbeds
- Wireless connections over Wifi or XBEE channels

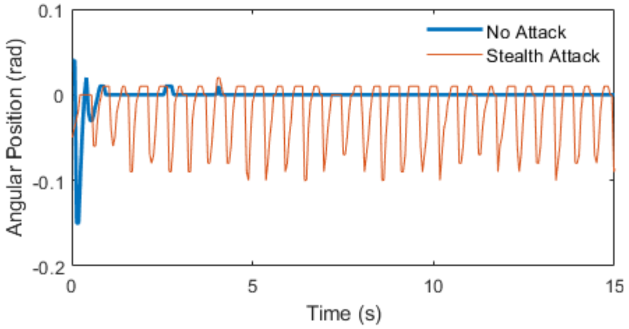


Fig. 7. Experimental System Response under a Stealth Cyber Attack with Magnitude of 0.2 rad and a Frequency of 2 Hz.

- Distributed control systems (tied multiple, consensus, or cooperative control)
- Other forms of cyber attack against networked CPS
- Cyber security measures

This testbed could be used in project-based learning activities that do not have the same desired learning outcomes. For example, students could be provided with working code for the sensor, controller, and actuator microcontrollers and be required to develop the code to connect multiple testbed units together on a wireless network or secure the networked CPS testbed from an instructor initiated cyber attack.

## VI. Lessons Learned, Recommendations, and Future Developments

This paper presented a low-cost, portable networked CPS testbed used for a project-based learning activity along with an example of an activity performed in an undergraduate course. Students produce a MATLAB Simulink simulation in advance of conducting a multiple session hands-on learning activity. This approach provides students an opportunity to demonstrate obtainment of the proposed learning outcomes detailed in section II-A.

Future implementation of this project will include increased emphasis on other forms of cyber attack. For example, students could implement a replay attack in addition to the stealth attack. Additionally, students should implement cyber security techniques to secure the testbed from these attacks. Course learning outcomes may be assessed using this project-based learning activity in future offerings.

## Acknowledgment

The authors would like to thank Daniel Rhodes from the Technical Support Department at USNA for his support in building the testbed.

## References

[1] K.-D. Kim and P. R. Kumar, “CyberPhysical Systems: A Perspective at the Centennial,” *Proceedings of the IEEE*, vol. 100, no. Special Centennial Issue, pp. 1287–1308, May 2012.

[2] A. Platzer, “Teaching CPS Foundations With Contracts,” Philadelphia, PA, Apr. 2013, p. 4.

[3] E. A. Lee and S. A. Seshia, *Introduction to Embedded Systems: A Cyber-Physical Systems Approach*, second edition ed. Cambridge, Massachuetts: The MIT Press, Dec. 2016.

[4] M. Trngren, M. E. Grimheden, J. Gustafsson, and W. Birk, “Strategies and considerations in shaping cyber-physical systems education,” *ACM SIGBED Review*, vol. 14, no. 1, pp. 53–60, Jan. 2017.

[5] D. Hristu-Varsakelis and W. Levine, “An undergraduate laboratory for networked digital control systems,” *IEEE Control Systems Magazine*, vol. 25, no. 1, pp. 60–62, Feb. 2005.

[6] J. M. Fuertes, R. Vill, J. Ayza, P. Mars, P. Mart, M. Velasco, J. Ypez, G. Torres, and M. Perell, “Hands-on course in networked control systems,” in *2012 20th Mediterranean Conference on Control Automation (MED)*, July 2012, pp. 1468–1473.

[7] P. J. Martin, “An Interdisciplinary Controls Curriculum for Cyber-Physical Systems Education,” Philadelphia, PA, Apr. 2013, p. 3.

[8] W. Taha, R. Cartwright, and R. Philippsen, “A First Course on Cyber Physical Systems,” Montreal, Canada, 2013, p. 3.

[9] C. Brown, S. Schall, J. Schultz, S. Simon, D. Stahl, S. Standard, F. Crabbe, R. Doerr, R. Greenlaw, C. Hoffmeister, J. Monroe, D. Needham, A. Phillips, and A. Pollman, “Anatomy, dissection, and mechanics of an introductory cyber-security course’s curriculum at the United States naval academy.” Haifa, Israel: ACM Press, July 2012, pp. 303–308.

[10] J. Hussey and J. Shaha, “Educational Approach to Cyber Foundations in an Undergraduate Core Program.” Rochester, NY: ACM Press, Oct. 2017, pp. 21–26.

[11] M. Trngren and E. Herzog, “Towards integration of CPS and systems engineering in education.” ACM Press, 2016, pp. 1–5.

[12] H. Chen and J. Tian, “Research on the Controller Area Network,” in *2009 International Conference on Networking and Digital Society*, vol. 2, May 2009, pp. 251–254.

[13] Y. Xie, L. Liu, R. Li, J. Hu, Y. Han, and X. Peng, “Security-aware signal packing algorithm for CAN-based automotive cyber-physical systems,” *IEEE/CAA Journal of Automatica Sinica*, vol. 2, no. 4, pp. 422–430, Oct. 2015.

[14] F. Pasqualetti, F. Dorfler, and F. Bullo, “Control-Theoretic Methods for Cyberphysical Security: Geometric Principles for Optimal Cross-Layer Resilient Control Systems,” *IEEE Control Systems*, vol. 35, no. 1, pp. 110–127, Feb. 2015.

[15] B. Croteau, D. Krishnankutty, K. Kiriakidis, T. Severson, C. Patel, R. Robucci, E. Rodriguez-Seda, and N. Banerjee, “Cross-Level Detection Framework for Attacks on Cyber-Physical Systems,” *Journal of Hardware and Systems Security*, vol. 1, no. 4, pp. 356–369, Dec. 2017.