A Graduate Curriculum in Cyber–Physical Systems

John A. Stankovic, Homa Alemzadeh, and Brad Campbell
University of Virginia, Charlottesville, VA, USA

John Lach
The George Washington University, Washington, DC 20052 USA

Lu Feng
University of Virginia, Charlottesville, VA, USA

Cody Fleming
Iowa State University, Ames, IA 50011 USA

Jonathan Goodall
University of Virginia, Charlottesville, VA, USA

Toluwalogo Odumosu
James Madison University, Harrisonburg, VA 22807 USA

Daniel Quinn, Yuan Tian, and Kelley Tobler
University of Virginia, Charlottesville, VA, USA

Editor's notes:
A group of academicians from the University of Virginia has developed a graduate curriculum for cyber–physical systems education. The curriculum follows the National Academy of Engineering report on the need for such content consisting of multidisciplinary classes and professional development skills.

— Mohammad Abdullah Al Faruque, University of California Irvine

At the core of society's grandest challenges lies the drive toward a smarter planet. The National Academy of Engineering's (NAE) grand challenge themes of sustainability, health, security, and joy of living all involve a need to sense, analyze, and actuate upon our world with efficient, safe, secure, and efficacious engineered systems. Realizing such systems requires a deeper understanding of the interfaces between the cyber and physical worlds, and this need has led to the establishment of the field of cyber–physical systems (CPSs). Although CPS, as a discipline and application enabler, has evolved tremendously over the past decade, it has become apparent that current graduate training does not sufficiently prepare students for fundamental discovery and innovation in CPS, or effective translation and application development. Indeed, the U.S. companies indicate that it is the lack of trained CPS personnel that inhibits their ability to develop new products and services [1]. Current CPS-related graduate training is lacking in three critical ways. First, the graduate curriculum itself is inadequate with courses not instilling the integrative knowledge needed for new scientific discovery and translational applications in the field of CPS. Second, most students do not have a sufficiently robust experience of convergence activities as part of their training. This lack is present not only in traditional engineering and computing education, but also extends into analyses of target application domains and associated grand challenges. Third, explicit professional development is often absent in graduate training. Such exposure is critical in CPS, given the field's potent role in our ever-evolving smart world and requires attending to social sensitivities with respect to ethics, safety, communication, and policy. With the help of a National Science Foundation Research Traineeship grant, at the University of Virginia School of Engineering and Applied Sciences, we have developed a new interdisciplinary graduate CPS curriculum to address these issues. Currently,

Digital Object Identifier 10.1109/MDAT.2020.3043376
Date of publication: 8 December 2020; date of current version: 20 May 2021.

2168-2356/20©2020 IEEE

we are developing a CPS certificate program; in the future, we plan to offer master's and PhD degrees in CPS.

In this article, we describe the principles, skills, and content for a new set of classes at the core of the new graduate CPS curriculum. We developed these classes based on a central principle that an effective CPS educational program cannot simply collect current classes, but must redesign the classes to emphasize the intersection of the cyber and physical issues. This principle was articulated in an NAE report [1] titled "a 21st Century CPS Education" and found at https://www.nap.edu/catalog/23686/a-21st-century-cyber-physical-systems-education. A short version of the main points in this NAE report was published in [2]. This article essentially describes our actions taken to address the issues in these reports and describes the resulting curriculum consisting of core classes, in-depth classes, and professional development. The first year of the new curriculum has now been completed and we also describe lessons learned to date.

Class design—overview and metrics

To develop classes in our new curriculum a two-part design methodology was followed: 1) direct use of the results of a three-year NAE CPS study and 2) combining these results with a logic model design [3]. Many universities and industries were involved in the NAE study, thereby providing greater insights to requirements than we could have alone. The subsequent report articulated the technical principles, skills, and professional development required for CPS. The technical principles and skill sets that are necessary for effective CPS research include the following topics: embedded programming, real-time computing, wireless sensor networks, signal processing, machine learning (ML), control, formal methods, safety, security, physical system dynamics, policy, and professional development. We coalesced the core principles of these topics into five core classes, following a logic model approach. A logic model articulates program inputs and activities, outcomes and impacts, evaluation activities, and formative changes. As an important point, note that the intent of these classes is to instill a basic understanding of the key aspects of these topics, not to create domain experts in any one topic. Each class brings together a set of topics rather than treating them independently. Each class also emphasizes principles behind how the cyber and physical aspects interact, how topics in the class relate to each other, provides hands-on team-based projects, and supports professional development skills. Full syllabi for these classes are available at https://engineering.virginia.edu/link-lab/education. Of course, other topics are important, and they are covered by in-depth classes (see the "In-depth classes" section).

This was the first year of the program, so long-term results are unknown. Our approach to evaluation combines: 1) standard course by course evaluation; 2) use of an external evaluator; and 3) employing an Industrial Advisory Board. Scores on all five core classes were between 4.3 and 4.8 out of 5. To date, the external evaluator performed a survey before the students began the program and at the end of the first year. This will continue for each year of the program. In the surveys and as metrics of interest, students were asked to rate their confidence for conducting CPS research, participating in interdisciplinary research, leading research projects, giving presentations, discussing research with other researchers, and developing professional writing skills as a result of the classes, testbeds, and professional development activities. The Industrial Advisory Board provides feedback every six months.

Five core classes

Overview of the core classes

Our curriculum includes five core classes (see Table 1) of which students must take CPS1 and at least two others. These classes may be taken in any order. Students must also take at least three in-depth classes (see the "In-depth classes" section). Currently, students also take other classes as dictated by their respective departments and obtain degrees from those departments. This CPS curriculum includes students from six programs: civil engineering, computer science (CS), electrical engineering, computer engineering, mechanical and aerospace engineering, and systems engineering. To describe

Table 1. Core CPS classes.

CPS1	Testbeds, Applications, and Policy
CPS2	Embedded Systems
CPS3	Signal Processing, Machine Learning, and Control
CPS4	Dynamical Systems
CPS5	Formal Methods, Safety, and Security

the content of the classes and how they support CPS principles and skills, we briefly describe each core class below.

Since this is a highly multidisciplinary program meant to train students in CPS from six programs, students enter with varying backgrounds. Because of this, we made the prerequisites general and they consist of requiring undergraduate engineering or CS degree. Note that all of these majors include calculus and programming backgrounds.

CPS 1: Testbeds, applications, and policy

Overview: The CPS 1 course has two main facets: 1) a section that covers the fundamentals of large-scale sensing systems and Internet of Things (IoT) testbeds and 2) another section that addresses policy issues in CPS, user-centered design, and the ethics of cyber–physical, intelligent, and autonomous systems. The goal of the class is to introduce students to practical CPS engineering design problems where they are expected to work within real-world constraints and create their CPS designs considering the policy and ethical dimensions of their work.

Learning objectives: The core learning objective of this course is for students to be able to evaluate the tradeoffs and implications of design decisions they make in terms of the effects on users, nonusers, and the performance of other parts of the system. This requires students to understand how multiple layers of any CPS interact, and evaluate how technical decisions have social implications. Secondary learning objectives include how to interface with data streams and sensors from an existing testbed, and understanding how to apply relevant ethical frameworks and legislation to CPS.

Integrative aspects: This course is fundamentally integrative. As a required course in the program, it includes students with many technical backgrounds from *all* of the relevant programs. It also includes both technical content on sensing systems and a structure for understanding how those technical systems affect users and nonusers. The two facets converge at the conclusion of the course with the final project. We also undertake an examination of contemporary and pertinent policy challenges of artificial and intelligent systems (AI/S) and a full discussion of the IEEE's Ethical Aligned Design framework and its applicability to CPS.

Design projects: The course culminates with a team-based design project that requires students to design a realistic sensing application using the specific technologies covered in the technical section. This technical design must be realistic and address an important application need. The students are also required to analyze and justify their designs utilizing the ethical and policy frameworks discussed in class. Students must orally defend not only their technical design choices but also show and explain how they considered users *and* nonusers when creating their system design.

CPS 2: Embedded systems

Overview: Embedded systems are special-purpose computers at the core of CPS that monitor and control physical processes through real-time interactions with sensors and actuators. This course introduces a CPS approach to the design and validation of embedded systems by focusing on the real-time operating systems (RTOSs) that enable concurrent execution of data acquisition, processing, and control tasks while satisfying real-time constraints on resource-constrained microcontrollers.

Learning objective: By taking this course, the students learn about the design principles for real-time systems, the building blocks of embedded system architectures, and the fundamentals of RTOS. They acquire hands-on technical skills and teamwork experience in embedded software development and become familiar with the real-world applications of RTOS for CPS and the IoT.

Integrative aspects: The lectures integrate an introduction to the CPS specific challenges in the design of embedded systems with the core technical concepts in real-time embedded systems, including memory management, hardware–software interfacing, interrupt handling, multitasking, thread synchronization and communication, semaphores, and real-time scheduling. Mini-projects are hands-on programming exercises that help the students learn embedded C programming and apply the concepts learned in the lectures to the design, implementation, and validation of an RTOS running on the ARM Cortex. Through these projects, the students become familiar with the ARM embedded programming environments, Texas Instruments (TI) microcontrollers, and the practice of version control using Git. Paper presentations and discussions expose the students to advanced topics and emerging research areas, such as state-of-the-art RTOS for IoT, real-world applications of RTOS, and dependability, safety, and security of embedded systems.

Design projects: For the final project, teams of students build upon the mini-projects to design a CPS application on top of their RTOS. They then propose innovative ideas for extending their application or the RTOS with new features or functionalities. At the end of the course, they present their innovative designs and working demos to the class and participate in a competition. They also submit a technical report in the IEEE or ACM conference paper format.

CPS 3: Signal processing, ML, and control

Overview: Signal processing, ML, and control are central tenets of CPS. This class first covers an introduction to sensors, sensor properties, and associated hardware and software platforms. It then introduces linear systems, signals, convolution, the discrete Fourier transform, how such mathematics are used in CPS applications, and how they execute on hardware/software platforms. ML is discussed as an extension to classical signal processing. ML topics include entropy, information gain, decision trees, random forests, neural nets, deep neural nets, convolutional neural nets, and long short-term memory (LSTM). The class also covers discrete feedback control. After CPS motivation for control, the control topics include system identification modeling, z-transforms, proportional (P) control, proportional–integral (PI) control, and proportional–integral–derivative (PID) control. Root locus and pole placement techniques are used.

Learning objectives: The goals of this class are to introduce the key principles in each of these areas and to understand how they relate to each other and are applied in real CPS. By taking this class, students will be able to understand the basic theory of signal processing, controls, and ML, and then develop algorithms for these three areas using common and state-of-the-art programming languages and tools.

Integrative aspects: Programming assignments on smartwatches are used to have students utilize the signal processing and ML concepts for a CPS application. For the smartwatch projects, students work in teams. The students use the ML tool WEKA and are exposed to PyTorch. Control topics are motivated by CPS and various software and platform issues are addressed including the impact of networking, operating systems and real-time scheduling algorithms on control loops. A MATLAB tool that addresses these control implementation issues is introduced. An integrating application is also presented that demonstrates the use of all the principles in the class, including sensing, hardware, signal processing, ML, and control and how they are all necessary for a single application. A lecture on ethics is included that questions the ethical use of sensing and subsequent analysis via signal processing and ML in various CPS applications.

Design projects: In addition to the integrative design projects mentioned above, students also perform a team-based final project and produce a written six-page conference style report and present their hands-on projects with demos.

CPS 4: Dynamical systems

Overview: In this class, students explore the role of solid and fluid mechanics in CPS design. To help students see the direct applications of what they are learning, the curriculum includes a heavy emphasis on design projects, as well as four guest lectures by subject-experts in: 1) multivehicle crash prevention; 2) model-based unmanned air vehicle (UAV) control; 3) urban flood prevention; and 4) microscale self-powered design.

Learning objectives: Students learn how classical mechanical systems are modeled, including the basics of rigid body dynamics and archetypical systems like spring–mass–dampers, pendulums, and wheeled robots. Students also learn the basics of continuum mechanics, including how to model beams, bridges, pipes, tanks, pumps, and turbines. As each type of model is introduced, students learn how these models can be integrated into CPS design, e.g., how modeling relates to PID-tuning, linear stability, model-predictive control, and energy consumption/production.

Integrative aspects: Throughout the course, the emphasis is placed on the elements of mechanical modeling that are most relevant to CPS. Special attention is paid to reference frames, for example, because of their role in autonomous robotics. Input/output frequency responses are also treated in depth because of the role they play in control and human-in-the-loop problems. Design projects and guest lectures integrate concepts from smart cities, smart health, and autonomous vehicles into the curriculum. The design projects were chosen specifically to build off concepts introduced in CPS1, especially privacy/ethics, robustness/safety, and wireless sensing/actuation.

Design projects: *Unit 1*: Student teams design concepts for either 1) laser pointers for hand-tremor patients or 2) weathervanes that measure turbulence. *Unit 2*: Student teams design energy harvesters that use piezocantilevers to charge a capacitor. *Unit 3*: Working in pairs, students design a model-based controller to stabilize a tethered dual-rotor system built specifically for the class. Students compete to create the most robust and effective algorithm. The design projects are meant to improve team building, entrepreneurship, and leadership skills, so written and oral presentations are key elements of how the projects are graded.

CPS 5: Formal methods, safety, and security

Overview: It is becoming more and more important to assure the safety and security of CPS since many CPS applications are safety-critical and life-critical. This course teaches students the foundational concepts of formal methods that enable the model-based design of safety-critical CPS with mathematically rigorous guarantees, as well as important safety and security issues in various CPS applications.

Learning objective: The overall objective of this course is to enable students to learn not only the fundamental knowledge of CPS and formal methods, but also important skills of creative thinking, collaborative working, and continuous learning of new techniques after the course is over.

Integrative aspects: This course includes lectures on formal methods concepts, including formal specifications (e.g., temporal logics, safety, and liveness requirements), formal modeling techniques (e.g., finite transition models, timed models, hybrid and dynamical models, and probabilistic models), and formal verification and synthesis techniques (e.g., model checking, satisfiability (SAT) & satisfiability modulo theories (SMT) solver, reactive synthesis, and runtime monitoring). These lectures use extensive CPS examples to introduce the theoretical concepts of formal methods. This course also offers opportunities for students to read, present, and discuss the state-of-the-art research papers on various topics of CPS safety and security, including various applications of medical CPS (e.g., pacemaker verification), automotive CPS (e.g., controller synthesis for autonomous driving), smart cities and smart homes (e.g., security and privacy in smart cities), human-in-the-loop CPS (e.g., a man in the middle attacks of CPS), and ML-enabled CPS (e.g., falsification with deep learning-enabled CPS).

Design projects: Students are assigned hands-on programming assignments to learn the use of formal methods tools (e.g., PRISM probabilistic model checkers and Z3 SMT solver) on CPS-related case studies (e.g., road network generation using the Z3 solver for testing autonomous cars in virtual reality environment). At the end of the course, students collaborate in groups to complete final course projects; they need to come up with their own ideas for the project, write a project proposal, present their research findings in class.

Discussion of the core classes

Each of these core CPS classes focuses on the CPS-relevant aspects of the subject, which allows them to maintain graduate-level technical depth but be taught in less time. As an example, consider the feedback control aspect of CPS3. A feedback control domain expert *with experience in CPS* may not need this module and place out, but a feedback person with no CPS experience may still need the module. Other students without a control background would strongly benefit from this material, e.g., most CS students do not take control theory. We also strongly believe that a module of a class, like this feedback control module, can be taught with enough depth to be useful to all CPS students. Anecdotally, this feedback control module has been taught to CS graduate students for multiple years prior to this program, and we are aware of at least 12 PhD theses that use material they learned in this module. For example, they were able to employ system identification (ID) for modeling and PI or PID control to improve the performance of their solutions.

Many other CPS education programs create 1 or 2 new classes and use other existing classes, e.g., a classical feedback control class. While this does provide more depth on certain topics, we find that such existing classes do not directly address CPS issues, so students are sometimes unable to apply what they learned. Existing classes are also not appropriate for all the students in this multidisciplinary program, so we created classes with multidisciplinary team projects that leveraged student's diverse backgrounds. Furthermore, to cover the needed principles and skills with existing classes would require too heavy a course load (e.g., more than ten classes); by customizing new classes, we cover the principles via modules with many fewer classes. Our hypothesis is that the reduced and focused material is better suited for building a foundation of CPS skills, particularly for nondomain experts. We carefully designed the classes and obtained approvals from curriculum committees of each department, so that students can obtain an MS or PhD in their home department and the certificate without having to take additional classes.

CPS2, the embedded systems course, is taught as a co-convened undergrad/grad course. The undergraduate section of this course is targeted for senior students in electrical and computer engineering (ECE) while the graduate section is open to the ME, MSc, and PhD students from six different programs. Note that if the National Science Foundation Research Traineeship (NRT) students already have the knowledge, they can get a waiver to place out. This is the only one of the five classes that is co-convened at this time.

In this program, students learn integrative skills that they do not learn in their respective departments. They also learn significant material from other departments that they would not learn otherwise. For example, how many civil engineers learn formal methods or computer security? How many CS students learn signal processing and feedback control? Because the program is so interdisciplinary, it is difficult to assess these skills using traditional departmental evaluation methods. We are therefore using an independent external evaluator to assess these skills and outcomes. Also, we have recently created a short introduction about CPS research and technology areas based on the National Science Foundation CPS Research and Education Model and how different core CPS courses in our program support the knowledge and skillsets for each. This introduction is presented to the students at the beginning of all the core CPS courses to show the interrelatedness of the courses.

In-depth classes

In addition to the core classes described above, students are required to take at least three in-depth classes. These classes build upon the core material and emphasize the cyber–physical nature of the respective topic of the class. The in-depth classes are also intended to prepare students for research in CPS, in part, because they are focused on building CPS deep domain expertise. The in-depth classes change each semester and are offered within their respective departments. Examples of the in-depth classes offered in the first year of this curriculum are shown in Table 2.

Professional development activities

The National Academies of Sciences, Engineering, and Medicine recently published a report on the importance of effective science communication. They suggest that our traditional approach to acquiring communication skills is not adequate for scientific topics and that “communicating science effectively is a complex task and an acquired skill [1].” In addition, both industry and government have recognized that students need to grow into professionals with the attendant needs for leadership, ethics, policy, and entrepreneurship, among others [4]–[7]. These characteristics are especially lacking in graduate education, which tends to focus primarily on achieving technical depth.

To address this need, we developed and implemented a sustained program of opportunities in professional development. Furthermore, these dimensions of professional development are explicitly and intentionally embedded in each of the core courses of the program and scheduled throughout each semester outside of the course curriculum. Our three-part approach for fostering and measuring professional development is as follows.

First, the CPS curriculum integrates professional development within each of our five new courses. This includes written and oral communication, leadership opportunities through group projects, and ethics embedded in the course content as it relates to the CPS discipline. In many of the courses, student grades are impacted by the efficacy of these skills (e.g., presentations and group projects).

Second, the NRT program includes monthly colloquia that are required of the cohort and go beyond what is covered in the core courses. The colloquia involve immersive activities which include both a speaking series as well as interactive sessions where students work in teams and provide briefings to the rest of the cohort and the CPS faculty. Topics include presentation skills, CPS

Table 2. Example of in-depth classes.

CPS and the Internet of Things	Robots and Humans
Human Considerations in Smart Infrastructure	Advanced Digital Design
Forefronts of Civil Engineering	User Experience Design
Human Error in Complex Systems	Hardware-Software Security
Autonomous Mobile Robots	Principle of Modeling for CPS
CPS and Cloud Computing	Mobile Sensing and Health
Statistical Modeling	Modeling and Control of Manufacturing Systems
Risk Analysis	Operating Systems

research showcase, communicating with advisors, leadership, ethics, and entrepreneurship. The activities have been facilitated not only by CPS faculty, but also experts in other areas of academia (e.g., business school and law) and industry.

Third, the students are required to attend—and then report on—at least three professional development activities or events per semester that occur outside the courses or monthly program meetings. While we have outlined general themes, we have also provided the students with the flexibility to pursue their interest and/or individualized areas for growth opportunities. We are leveraging the offerings within the School of Engineering and Applied Sciences, the University of Virginia broadly (including our PhD Plus program which focuses on various professional skills and competencies), and beyond. Example events include the “Three Minute Thesis” competition, Project Management, Writing Abstracts, and the UVA Engineering “Communications for Engineers” course.

Lessons learned

Developing a comprehensive new curriculum for CPS across six established programs is difficult. As with any new discipline, we have been working to educate stakeholders and university leadership about the content of our new courses (and subsequent departmental curricula approvals). We have also addressed concerns about if the material is at the graduate level—especially as we attempt to provide a baseline of CPS knowledge to students from multiple disciplines. One key aspect of this difficulty is that the tension between the interdisciplinary and integrative content of our new classes versus the in-depth domain expertise content of existing classes. While most CPS faculty see the intellectual depth in cross-cutting issues and mapping principles to real-world constraints, some faculty (mostly) outside of CPS feel that there is not enough depth in “their” discipline and therefore does not constitute a graduate level class. For example, CPS3 covers control, but it does not provide traditional depth in control such that a student would become a domain expert in control, but that is NOT the intent. It DOES make the student aware of central control principles, methodologies, tools, and applicability that include realistic CPS issues such as the effect of networking, operating systems, and other nondeterministic delays on control loops. These latter topics are usually not found in domain in-depth control classes. One key argument made to eventually have the classes approved as graduate classes was that we are producing multifunctional expertise, not just multifunctional teams!

Initially, we planned on having each core class taught by two instructors: one with expertise in the physical and a second one with expertise in the cyber. Due to department teaching loads and faculty availability this was not generally possible. Although classes were designed by multiple experts, only CPS1 was co-taught.

Some students entering the program argued that they knew a lot of the material in the core classes. However, none were exposed to the integrative nature and focus on the intersection of the cyber and the physical. As a compromise, students who entered the PhD program with a master’s degree can petition to obtain credit for one core and one in-depth class.

There is a strong demand for this curriculum from students. Our first class of students included 21 PhD students and six master’s students from across the six degree programs. Nine of the PhD students were awarded fellowships through an NRT award and the rest were funded by faculty under other grants. The program has also attracted a diverse group of students with eight of the nine fellows and 14 of the 27 students being from underrepresented groups in STEM. There is a strong demand from master’s students and we intend to expand to those students in the coming year.

Comparison to other programs

CPS is an emerging discipline and is being offered as a focused area of study at a growing number of universities. While CPS research is being conducted at many universities, often at the PhD level, there are relatively few institutions that formally recognize CPS through certificates or degrees.

CPS-related graduate certificates are currently being offered at various institutions and are available to degree and nondegree seeking students. Stevens Institute of Technology [8] has an emphasis on embedded systems. Michigan Tech [9] has a program focused on the safety and security of autonomous vehicles. Wayne State [10] has the most comprehensive certificate program offering six CPS tracks including smart health, smart grid, control and robotics, autonomous vehicles, smart transportation, and sensing.

Master's degrees in CPS are currently being offered at a few institutions. The University of California, Irvine, offers a Master of Embedded & CPS, which focuses on embedded systems. Courses in this program specifically focus on "hardware and software, sensor networks, real-time systems, AI, ML, security, control, and entrepreneurship" [11]. Vanderbilt University's Master of Engineering degree program provides a foundation with core CPS courses and including design and then allows the students to select application areas to focus further. The application areas are focused on "Embedded Systems, Control Engineering, Robotics, Transportation Engineering, and Biomedical Engineering" [12].

In comparison, our approach has tried to be more fundamental and comprehensive by designing five new classes that *focus* on the intersection of the physical and cyber, push the science that underlies CPS applications, is multidisciplinary across six programs, and is based on the NAE report for principles and skills required for CPS jobs and research. Furthermore, by spanning the program across six departments, we were able to build technical depth into each module of the newly developed core courses. As a result, none of the disciplines in the program are "secondary"—the program covers each discipline equally, thereby teaching students the underlying science of CPS before they split off into the in-depth classes.

The CPS discipline has evolved tremendously over the last decade. Education has not kept pace and solutions that involve one or two new classes or merely collecting current classes are not sufficient. A new discipline with its own degrees is needed. The curriculum we created teaches material at the intersection of the cyber and physical without sacrificing depth in the individual disciplines in a way that is both sustainable over the long term and reproducible at multiple schools. The curriculum is not only focused on the technical aspects, but includes training for professional development, ethics, policy, teamwork, and leadership skills. ■

Acknowledgments

This work was supported by the National Science Foundation NRT program under Grant 1829004.

References

[1] National Academies Press. (2016). *A 21st Century Cyber–Physical Systems Education, Report, Committee on 21st Century Cyber–Physical Systems Education*. [Online]. Available: http://www.nap.edu/catalog/23686/a-21st-century-cyber-physical-systems-education

[2] J. A. Stankovic, J. Sturges, and J. Eisenberg, "A 21st century cyber–physical systems education," *Computer*, vol. 50, no. 12, pp. 82–85, Dec. 2017.

[3] *Logic Model*. Accessed: Jun. 25, 2020. [Online]. Available: https://en.wikipedia.org/wiki/Logic_model

[4] J. A. Donnell et al., "Why industry says that engineering graduates have poor communication skills: What the literature says," in *Proc. ASEE Annu. Conf. Expo.*, 2011, p. 13.

[5] Infusing Ethics Selection Committee, *Infusing Ethics Into the Development of Engineers: Exemplary Education Activities and Programs*. Washington, DC, USA: National Academies Press, 2016.

[6] J. C. Palmer et al., "LEADing the way: A review of engineering leadership development programs," in *Proc. 123rd ASEE Annu. Conf. Expo.*, Jun. 2016, pp. 1–16.

[7] R. Schuhmann et al., "Engineering leadership education—The path forward," in *Proc. ASEE Annu. Conf. Expo. (ASEE)*, 2015, p. 20.

[8] Stevens Institute of Technology. Accessed: Aug. 1, 2020. [Online]. Available: https://www.stevens.edu/school-systems-enterprises/graduate-certificates

[9] Michigan Tech. Accessed: Aug. 1, 2020. [Online]. Available: https://www.mtu.edu/gradschool/programs/certificates/autonomous-cyber-physical-systems/

[10] Wayne State. Accessed: Aug. 1, 2020. [Online]. Available: https://engineering.wayne.edu/cyber/graduate-certificate.php

[11] University of California, Irvine. Accessed: Aug. 1, 2020. [Online]. Available: https://grad.uci.edu/academics/degree-programs/ssgpdp/MasterofEmbeddedCyber-PhysicalSys.php

[12] Vanderbilt. Accessed: Aug. 1, 2020. [Online]. Available: https://engineering.vanderbilt.edu/academics/m_eng/CPS/index.php

John A. Stankovic is the BP America Professor and the Director of the Link Lab, University of Virginia (UVA), Charlottesville, VA, USA. His research interests are in cyber–physical systems, smart health, and the Internet of Things (IoT). Stankovic has a PhD in computer science from Brown University, Providence, RI, USA. He is a Fellow of IEEE and ACM.

Homa Alemzadeh is an Assistant Professor with the University of Virginia, Charlottesville, VA, USA. Her research interests are in dependable and secure computing, resilient cyber–physical systems, and medical devices and robotics. Alemzadeh has a PhD in electrical and computer engineering from the University of Illinois Urbana–Champaign, Champaign, IL, USA. She is a member of IEEE.

Brad Campbell is an Assistant Professor of computer science and electrical and computer engineering with the University of Virginia, Charlottesville, VA, USA. His research areas include systems for the Internet of Things, energy-harvesting sensors, and smart buildings. Campbell has a PhD in computer science from the University of Michigan, Ann Arbor, MI, USA. He is an ACM member.

John Lach is the Dean of Engineering with The George Washington University, Washington, DC, USA. His primary research interests are in cyber–physical systems, embedded sensor systems, smart and connected health, and body sensor networks. Lach has a PhD in electrical engineering from the University of California at Los Angeles (UCLA), Los Angeles, CA, USA (2000). He is an IEEE Senior Member.

Lu Feng is an Assistant Professor of computer science with the University of Virginia, Charlottesville, VA, USA. Her research interests are in cyber–physical systems and formal methods. Lu has a PhD in computer science from the University of Oxford, Oxford, U.K. (2014). She is a member of ACM and IEEE.

Cody Fleming is an Associate Professor of mechanical engineering with Iowa State University, Ames, IA. His research interests are in safety, security, and design of cyber–physical systems, particularly those with high levels of automation. Fleming has a PhD in aeronautics and astronautics from the Massachusetts Institute of Technology (MIT), Cambridge, MA, USA. He is a member of IEEE and AIAA.

Jonathan Goodall is a Professor of civil engineering and the Associate Director of the Link Lab, University of Virginia, Charlottesville, VA, USA. His research is in water resources, hydroinformatics, and smart cities. Goodall has a PhD in civil engineering from the University of Texas at Austin, Austin, TX, USA. He is a Fellow of ASCE.

Toluwalogo Odumosu is currently a Visiting Associate Professor with the College of Integrated Science and Engineering, James Madison University, Harrisonburg, VA, USA. His primary research interests are digital privacy, mobile and telecommunications policy, engineering ethics, and the Internet of things (IoT). Odumosu has a PhD from Rensselaer Polytechnic Institute in Science and Technology Studies (STS), Troy, NY, USA.

Daniel Quinn is an Assistant Professor with the University of Virginia, Charlottesville, VA, USA. His primary research interests are in fluid dynamics, bio-inspired robotics, smart and connected health, and cyber–physical systems. Quinn has a PhD in mechanical and aerospace engineering from Princeton University, Princeton, NJ, USA (2015). He is an ASEE and APS member.

Yuan Tian is an Assistant Professor of computer science with the University of Virginia, Charlottesville, VA, USA. Her research focuses on the security of cyber–physical system and machine learning. Tian has a PhD in electrical and computer engineering from Carnegie Mellon University, Pittsburgh, PA, USA. (2017). She is a member of ACM, IEEE, and Usenix.

Kelley Tobler is the Program Manager for UVA Engineering's National Science Foundation Research Traineeship (NRT) in cyber–physical systems. Tobler has a bachelor's degree in communications from the University of Arizona, Tucson, AZ, USA.

■ Direct questions and comments about this article to John A. Stankovic, Department of Computer Science, University of Virginia, Charlottesville, VA 22904 USA; stankovic@cs.virginia.edu.