Design Automation of Cyber-Physical Systems: Challenges, Advances, and Opportunities

Sanjit A. Seshia, *Senior Member, IEEE*, Shiyan Hu, *Senior Member, IEEE*, Wenchao Li, *Member, IEEE*, and Qi Zhu, *Member, IEEE*

***Abstract*—A cyber-physical system (CPS) is an integration of computation with physical processes whose behavior is defined by both computational and physical parts of the system. In this paper, we present a view of the challenges and opportunities for design automation of CPS. We identify a combination of characteristics that define the challenges unique to the design automation of CPS. We then present selected promising advances in depth, focusing on four foundational directions: combining model-based and data-driven design methods; design for human-in-the-loop systems; component-based design with contracts, and design for security and privacy. These directions are illustrated with examples from two application domains: smart energy systems and next-generation automotive systems.**

***Index Terms*—Cyber-physical systems, design automation, formal verification, formal specification, machine learning, synthesis, human-robot interaction, security, privacy, energy management, automotive engineering.**

I. Introduction

A CYBER-PHYSICAL system (CPS) is an integration of computation with physical processes whose behavior is defined by *both* computational and physical parts of the system [1]. Embedded computers and networks monitor and control the physical processes, usually with feedback loops where physical processes affect computations and vice versa. Depending on the characteristics of CPS that are emphasized, they are also variously termed as *embedded systems*, the *Internet of Things*, the *Internet of Everything*, the *Industrial Internet*, etc. Examples of CPS include today's automobiles, fly-by-wire aircraft, medical devices, power generation and distribution systems, building control systems, robots, and many other systems. As an intellectual challenge, CPS is about the *intersection*, not the union, of the physical and the cyber worlds. It is not sufficient to separately design, analyze, and understand the physical components and the computational components, and then to connect them together. To enable the integration of different components including computation, networking, and physical processes, we must understand and design for their interaction.

CPSs have been around for a long time, but it is only recently that the area has come together as an intellectual discipline. As a result, even though tools and techniques for the design automation of CPS exist in certain categories, there is not yet a widely-used design methodology, supported by tools, for CPS as there is, for example, for digital circuit design. Additionally, CPS are more complex than integrated circuits along several dimensions. Indeed, there is not a single "design space" for CPS as there is for digital circuits; in fact, the commonalities in the design problems for different CPS applications arise from the combination of the following features. Today's CPS are *heterogeneous* entities that span the cyber and physical worlds, hardware and software, sensors and actuators, etc. They are also increasingly *distributed* systems, often of a *large scale*. They must operate in highly *dynamic* environments and for dynamically-changing objectives, and therefore, must be *adaptive*. Finally, many CPS operate in concert with *human* operators, and the human aspect of the design of such systems must be carefully considered. We detail this combination of characteristics in Section II and make the case that, taken together, this combination of characteristics needs significant advances in the theory, techniques, and tools for design automation of CPS.

This need is a significant opportunity for the design automation community. The opportunity extends across the entire design process including specification, modeling, language design, programming, simulation, verification and validation, synthesis equivalence and refinement checking, mapping, performance analysis and optimization, interface design, network design, testing, debugging, diagnosis and repair, etc. We contend that each of these categories needs more advances in fundamental theory, techniques, and tools in order to make the design of CPS as routine and their behavior as predictable as the design and operation of digital systems is today. We need new design methodologies for CPS with impact comparable to that of the register transfer level (RTL) design flow for digital circuits. Moreover, the opportunity to create new design methodologies for CPS is amplified by the growing availability of data, both on the design of systems and on their operation in the field. In this paper, rather than enumerating the many specific opportunities for design automation of CPS,

Manuscript received April 15, 2016; revised July 6, 2016 and October 7, 2016; accepted October 31, 2016. Date of publication December 8, 2016; date of current version August 18, 2017. The work of S. A. Seshia was supported in part by the National Science Foundation under Grant CCF-1139138, Grant CCF-1116993, and Grant CNS-1545126, in part by DARPA under Grant FA8750-16-C-0043, in part by the Toyota Motor Corporation through the CHESS Center, and in part by the TerraSwarm Research Center, one of six centers of STARnet, a Semiconductor Research Corporation program sponsored by MARCO and DARPA. The work of W. Li was supported in part by the National Science Foundation under Grant CCF-1646497, and in part by DARPA under Grant FA8750-16-C-0043. The work of Q. Zhu was supported in part by the Office of Naval Research under Grant N00014-14-1-0815 and Grant N00014-14-1-0816, and in part by the National Science Foundation under Grant CCF-1553757 and Grant CCF-1646381. This paper was recommended by Associate Editor X. Li.

S. A. Seshia is with the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, Berkeley, CA 94720-1770 USA (e-mail: sseshia@eecs.berkeley.edu).

S. Hu is with the Department of Electrical and Computer Engineering, Michigan Technological University, Houghton, MI 49931 USA.

W. Li is with the Department of Electrical and Computer Engineering, Boston University, Boston, MA 02215 USA.

Q. Zhu is with the University of California at Riverside, Riverside, CA 92521 USA.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCAD.2016.2633961

we focus on an exposition of selected foundational directions (see Section III). We illustrate these directions with examples from two application domains: 1) *smart energy systems* and 2) *next-generation automotive systems* (Section IV). This paper concludes in Section V with an outlook to the future for design automation for CPS.

This paper is not intended to be an exhaustive survey of work on design automation of CPS. We focus on selected topics that we believe hold much promise for future work. Certain important recent efforts that fall outside these topics are not covered. The reader is referred to other excellent articles for a broader view of the landscape for design automation of CPS (see [2]–[5]).

II. Challenges

The unique design challenges for CPSs emerge from the following combination of characteristics.

1) *Hybrid:* As mentioned earlier, CPS is about the intersection of the computational and physical worlds. For this reason, the modeling, design, and analysis of CPS requires effective theory and tools to reason about hybrid systems that combine discrete and continuous dynamics.
2) *Heterogeneous:* The components of a CPS are of various types, requiring interfacing and interoperability across multiple platforms and different models of computation.
3) *Distributed:* In today's CPSs, components are typically networked, and can be separated physically and/or temporally.
4) *Large-Scale:* The size of CPSs, measured in terms of the number of primitive components a system is made up of, is growing rapidly, leading to a "swarm" of sensors, actuators, computation, and communication devices interconnected and generating vasts amount of data.
5) *Dynamic:* The environment of the CPS evolves continually, and thus the design and operation of the system must account for such dynamic changes in the environment. Moreover, the environment can behave adversarially, actively trying to violate desired system properties.
6) *Adaptive:* Given a dynamic environment, the CPS must adapt to it, possibly online. The system may employ machine learning to adapt to a changing environment. The distinction between "design-time" and "run-time" is thus blurred.
7) *Human-in-the-Loop:* Several CPS operate in concert with humans: they involve human operators or interact with humans and human-controlled systems in their environment. Examples include semiautonomous vehicles (where "self-driving" autonomous controllers must interact with human drivers and pedestrians) and robotic surgical devices (where a doctor or nurse must cooperate with an autonomous controller to achieve their objective). The design of such systems must necessarily consider as a central aspect the role of and interface to the human(s) in the loop.

These characteristics may seem very different from each other. However, in our opinion, the major design challenges for CPS stem from how these characteristics come together in real systems. For example, in order to verify advanced driver assistance systems (ADASs) in automobiles, one must consider that these are *hybrid* systems operating in a *dynamic* environment that interact with *humans* and use *machine learning* components. The design tools must be capable of handling this combination and the resulting concerns.

Thus, the overarching challenge for the design automation community is to develop theory, techniques, and tools for the design of CPS with the above combination of characteristics in order to ensure that the designed systems are dependable, secure, and high performance. In turn, we believe that this challenge needs a *design automation methodology* with the following blend of features.

1) *Cross-Domain:* The hybrid and heterogeneous nature of CPS means that the tools for their design must necessarily be cross-domain. For example, there is a need for techniques for co-simulating different components of a CPS, such as the mechanical aspects of a robot's motion with the electronic and software processes that control its actions.
2) *Component-Based:* The increasing large scale of CPS implies that the only way to deal with growing complexity is to perform design in a modular fashion. Specifically, there is a need for establishing libraries of reusable, verified components with clearly specified *interface contracts*. Tools for enabling such component-based, contract-based design are essential.
3) *Learning-Based:* The growing amount of data on CPS, coupled with the need for systems to be adaptive and handle dynamic environments points to the need for CPS design based on data-driven learning. However, such learning must be coupled with principled model-based design (MBD) and formal methods that can give guarantees on correct operation. The development of such learning-based design automation techniques is an important need going forward.
4) *Time-Aware:* One of the key aspects connecting the cyber and physical worlds is time. In particular, in order to understand the joint dynamics of the cyber and physical components of a CPS, one must come up with a suitable abstraction of time that accurately captures their joint evolution. The distributed nature of many CPS adds another level of complexity, potentially varying the notion of time across different components of the system. CPS design tools must be time-aware and encapsulate suitable abstractions in order to ease the design process.
5) *Trust-Aware:* The design of distributed CPS that operate in dynamic, adversarial environments must address fundamental issues of trust. Security and privacy, which previously were secondary concerns, have now become top design concerns for CPS. Moreover, the cyber-physical nature of systems is bringing new security and privacy concerns to the fore. Tools for design automation must be able to model threats, design for them, and analyze systems for vulnerabilities.
6) *Human-Centric:* It is becoming increasingly clear that design automation tools for CPS must both address the human aspect of design and of the systems being designed. Tools must complement human ingenuity by automating the tedious aspects of design while allowing humans to express their creativity as well. Similarly, given the growing importance of human-in-the-loop CPS in everyday life, it is critical to develop tools to help model, design, and verify such systems.

In the following sections, we explore the opportunities for CPS design automation in more depth.

III. Foundational Directions

We list four directions that, in our opinion, highlight the foundational aspects of design automation for CPS. Each of

these directions involves developing a unique set of features that we described in the previous section. Moreover, each direction represents a significant shift from the traditional paradigms in design automation. These directions, however, are not orthogonal to each other and should be viewed as addressing different but cross-cutting aspects of automating CPS designs. For example, the combination of model-based and data-driven approaches (Section III-A) may very well be applied to the analysis and synthesis of human-in-the-loop systems (Section III-B).

A. Model-Based Design Meets Data-Driven Design

MBD is a paradigm for system design in which the design process begins with the creation of high-level models which are then used to guide further development, simulation, verification, and testing of the system. MBD has found industrial use in the field of embedded systems, particularly in automotive and avionics applications [6], [7]. The MBD approach seeks to place an emphasis on abstract, mathematical modeling as a first step before getting into low-level details of the implementation. The availability of such models, with associated formal (mathematical) specification of desired/undesired behaviors, can aid in simulation and verification early in the design process, thus weeding out bugs in the logic of the system at a point where the cost of finding and fixing them is still relatively low, and improving overall system dependability.

In certain settings, however, the model-based approach falls short. Consider, for example, a system operating in a highly variable, uncertain environment, such as a self-driving vehicle. In this case, constructing a good model of the environment *a priori* can be very difficult. Instead, one might rely on extensive field testing to collect data about the environment of the vehicle, and then employ algorithms that learn from the data in order to compute the optimal control strategy. Moreover, the genesis of such an approach goes back several years, to ideas such as *adaptive control* [8], [9]. At the present time, a prominent example of this data-driven, learning-based approach are Google's (mostly) self-driving cars [10]. The seeming success of this approach, paired with the availability of increasing amounts of data, leads one to ask: is the data-driven approach the right one?

We argue instead that one needs a combination of model-based and data-driven approaches. Today's CPSs need to be both *dependable* and *adaptive*. A model-based approach facilitates the use of formal methods—computational proof techniques—to improve dependability. A data-driven approach facilitates adaptation by learning from the data. For CPSs that operate in safety-critical or mission-critical settings and dynamic, uncertain environments, both approaches are essential.

The confluence of MBD with data-driven design has produced several exciting directions for future work. We elaborate on two particularly compelling and foundational directions.

1) Formal Inductive Synthesis: How can we employ data-driven learning to improve MBD?

In order to answer this question, let us examine the process of MBD. The first step is to create models, including requirements on the system to be designed, and assumptions on its operating environment. One must gain assurance, through the use of systematic simulation and proof methods, that the model of the system, when composed with the model of its environment, satisfies the desired requirements. Next, one must generate implementations from the models in a systematic manner that guarantees that the behavior of the implementation conforms to the model. Such conformance checking requires additional verification. The implementations also need to be mapped to a physical platform and various platform-specific requirements must be verified, such as conformance to timing requirements.

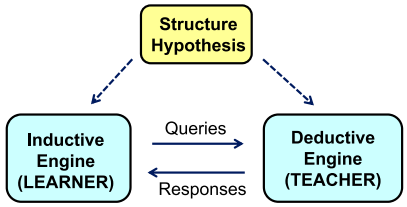


Fig. 1. Three main elements of the SID approach.

It is clear from the above description that *synthesis* is a central and recurring component of the MBD process. Models and specifications must be synthesized. Implementations must be synthesized. Platform-specific features must be synthesized. Perhaps more surprisingly, the verification steps also involve synthesis (albeit a different form): the synthesis of "verification artifacts" such as inductive invariants, preconditions and postconditions, assume-guarantee contracts, ranking functions, etc. In summary, in order to automate the MBD process effectively, one must devise efficient procedures for the synthesis of a variety of *formal artifacts*.

How best can this synthesis be done? One approach is *deductive*, to formulate and systematically apply rules that transform a high-level specification into the artifact to be synthesized. However, it can be difficult, *a priori*, to specify all the needed transformation rules, and the combinatorial search does not usually scale to industrial problems. Can one instead leverage data available from past design experience as well as data generated during the MBD process (e.g., from simulations of models) to automate the tedious aspects of synthesis?

A particularly effective approach that has emerged in recent years is based on the combination of *induction* and *deduction*. We use the term *induction* in its classic sense as the process of inferring a general law or principle from observation of particular instances.[1] Machine learning algorithms are typically inductive, generalizing from (labeled) examples to obtain a learned *concept* or *classifier* [11], [12]. *Inductive synthesis* is the process of synthesis from examples (sample data). *Formal inductive synthesis* (FIS) [13] is the process of synthesizing from examples with formal guarantees, and it is this flavor of inductive synthesis that is relevant in the MBD context.

An effective approach to solve an FIS problem combines three elements: 1) a *structure hypothesis*; 2) *induction*; and 3) *deduction*. The structure hypothesis is an encapsulation of designer insight in a syntactic form. It can take the form of a template, a component library, a partial program, etc. We refer to this approach as the SID methodology, where the three letters stand for the three elements: 1) structure; 2) induction; and 3) deduction. Fig. 1 depicts the above three elements where an inductive engine $\mathcal{I}$ makes queries to a deductive oracle $\mathcal{D}$ and receives responses in turn. A mathematical framework implementing the SID methodology is the *oracle-guided inductive synthesis* (OGIS) approach [13].

The SID methodology has been effectively applied to several practical problems in the design automation of CPSs,

[1]The term "induction" is often used in the EDA/verification community to refer to *mathematical induction*, which is actually a deductive proof rule. Here we are employing induction in its more classic usage arising from the field of Philosophy.

including requirement generation [14], assumption generation [15], controller synthesis [16], switching logic synthesis [17], timing analysis of embedded software [18], [19], and Lyapunov analysis for control [20]. We have just begun to scratch the surface of what is possible with an approach that integrates induction from data with deduction from models, and many exciting future directions beckon. For further details, we refer the interested readers to this papers on SID [21], [22] and OGIS [13].

2) Trustworthy Machine Learning: CPSs that include components based on machine learning have certain distinctive characteristics. First, the mainstream machine learning techniques of today do not perform *exact* learning—i.e., they may have a (hopefully) small mis-classification error. Second, they are only as accurate as the data used to train them with. Thus, if machine learning methods are to be used within safety-critical CPS, we must develop techniques to verify system correctness whilst considering their potential inaccuracies. In other words, we need to develop techniques for *trustworthy machine learning*.

What are the general principles for trustworthy machine learning? This is a nascent topic, and a few proposals are just emerging [23], [24]. Here we highlight some important directions (see [24] for more details).

1) *From Predictions to Explanations:* The nature of such machine learning algorithms must not just be *predictive* but also *explanatory*. In other words, when the machine learning algorithm makes a prediction (e.g., classifies an object in front of a vehicle as a person), it should be able to support that prediction with a suitable "explanation" encoded in a form amenable to formal analysis. Such explanations can then be checked against sufficient conditions for safe operation that have been derived at design time.
2) *Systematic Training:* Training and test data for machine learning algorithms must be systematically generated. In the ideal case, they must be generated in a manner so as to give formal guarantees about convergence to the target concept to be learned. In many cases, this will require sampling points from a constrained space subject to requirements on the output distribution. This is roughly similar to constrained random verification in electronic design automation, although there are some key differences as well.
3) *Specifications for Learning Components:* One challenge for verifying the correctness of a machine learning component is to formulate its specification, i.e., to make precise what "correctness" means. Since machine learning is often used to perform tasks otherwise done by humans, and given that many of these tasks are versions of the Turing test, it is in general impossible to formalize the specification. Even so, it may be possible to employ instead an end-to-end specification for the overall system that uses machine learning, and to combine that with specification mining to analyze the machine learning component systematically.

The design of trustworthy machine learning components thus points to another rich domain for the integration of machine learning with formal methods. It is thus fertile ground for future work.

B. Human-in-the-Loop Systems

Several CPSs are *interactive*, i.e., they interact with one or more human beings, and the human role is central to the correct working of the system. Examples of such systems include fly-by-wire aircraft control systems (interacting with a pilot), automobiles with "self-driving" features (interacting with a driver), remote-controlled drones (interacting with a ground operator), and medical devices (interacting with a doctor, nurse, or patient). We refer to the control in such systems as *human-in-the-loop control systems* and the overall system as a human CPS (h-CPS). The costs of incorrect operation in the application domains served by these systems can be very severe. Human factors are often the reason for failures or "near failures," as noted by several studies (see [25], [26]). Correct operation of these systems depends crucially on two design aspects: 1) *interfaces* between human operator(s) and autonomous components and 2) *control* strategies for such human-in-the-loop systems.

At the present time, some of the most compelling h-CPS problems arise from the automotive domain. In particular, over the past decade, automobiles with "self-driving" features (otherwise also termed as ADASs) have made their way from research prototypes to commercially-available vehicles. Such systems, already capable of automating tasks such as lane keeping, navigating in stop-and-go traffic, and parallel parking, are being integrated into medium-to-high end automobiles. However, these emerging technologies also give rise to concerns over the safety and performance of an ultimately driverless car. For various engineering, legal and policy reasons, a car that is self-driving at all times may not be a reality for a few more decades. However, semiautonomous driving is already here, and a myriad of scientific and engineering challenges exist in the design of shared human and autonomous control. For these reasons, the field of semiautonomous driving is a fertile application area for CPS design automation. Section IV-B2 has a deeper exploration of this application domain.

In this section, we give an overview of the main challenges associated with the principled design of h-CPS, including the following.

1) *Modeling:* What distinguishes a model of a h-CPS from a typical CPS?
2) *Specification:* How do the requirements change for a h-CPS?
3) *Verification:* What new verification problems arise from the human aspect?
4) *Synthesis:* How can we co-synthesize control and interfaces for h-CPS?

The reader may find a slightly longer exposition of this topic, with a particular focus on semiautonomous driving, in [27].

1) Modeling: The key difference between an h-CPS and a fully-autonomous system is that, in an h-CPS, we additionally have the human operator(s) with whom control must be shared. Therefore, the h-CPS model must contain a representation of the human operator(s) as well as a subsystem that mediates between the human operator(s) and the autonomous controller. We refer to this subsystem as the *advisory controller* (since it guides the human operator) or the *mixed-initiative controller* (since it blends human and autonomous control), and denote it by ADVISOR. The design of the human–machine interface, thus, is also of great importance.

Additionally, in order to give guarantees about an h-CPS system, one must have a reasonable model of the human operator. Modeling humans can be tricky. While there is a large literature on human cognitive modeling, this is usually informal and performed by experts for specialized domains with highly-trained operators (e.g., cockpit flight control). In this

context, it is useful to recall the statement by George Box: "all models are wrong, but some are useful." The principled design of h-CPS requires the judicious use of human models. Our position is to use formal models of human operators that are grounded in empirical data. In other words, we propose that, while the structural form of a model can be informed by expert guidance, the precise model used for design be inferred from observations of human behavior.

To summarize, the key points of differentiation between modeling a h-CPS and modeling a fully-autonomous CPS are as follows.

1) The use of *data-driven* human modeling.
2) The inclusion of relevant aspects of the *human–machine interface*.
3) The presence of the *advisory controller*.

2) Specification: Human CPS have certain unique requirements which need to be formalized as formal specifications for verification and control. In addition to traditional forms of specification, captured through formalisms such as temporal logic, one must also write down specifications relating to the human operator(s) and the human–machine interface. Some initial steps have been taken in this regard [27], [28], formalizing the following meta-specifications.

1) *Safe and Correct Autonomy:* The h-CPS must preserve certain key safety properties at all times, and must guarantee overall correct operation (as captured by a formal specification) at all times when the autonomous agent is in control.
2) *Effective Monitoring:* The advisory controller should be able to monitor all information about the h-CPS and its environment needed to determine which agents (human or autonomous) must be in control. This is a requirement on the types of sensors required and their quality and performance.
3) *Minimally Intervening:* A primary purpose of including an autonomous controller in the system is for human operators *not* to have to be in control at all times. Therefore, we add an optimality requirement: the advisory controller should minimize interventions by the human operator(s) to take back control, where minimality is defined by a suitable cost function.
4) *Prescient:* Time is a central parameter in the design of h-CPS. The advisory controller must be able to *predict in advance* conditions that may require switching control from human or autonomous and vice-versa, or other interventions by the human (e.g., asking to change the navigation goal).

These meta-properties are just a start. Formalizing and specializing these meta-requirements for specific application domains (e.g., semiautonomous driving) and for other human–machine interaction models is a problem that remains to be fully solved, and an important direction for future work.

3) Verification and Synthesis: The verification and control problems for h-CPS depend heavily on the formalisms for modeling and specification. Thus, one needs to define the latter formalisms before the verification and control problems can be effectively tackled. Even so, some general principles are worth stating.

1) *Verification Must Operate on Models Inferred From Data:* It is clear that h-CPS models will include substantial parts that are learned from data that may be incomplete and with learning algorithms that have intrinsic inaccuracies. The models must represent this uncertainty and inaccuracies as first-class entities, and verification algorithms must be adapted to operate on such models. Although some initial results are available [29], [30], much more remains to be done.
2) *Verification Must Provide Quantitative Output:* The bulk of verification techniques target Boolean questions, such as whether a model satisfies a property or not. However, with humans in the loop, there is a lot of uncertainty in the modeling process, and hence Boolean answers may lose substantial information about risk. Better quantitative verification methods must be developed.
3) *Controller Synthesis Must Yield Both Autonomous Controller and Advisory Controller:* Traditional controller synthesis simply solves for the former. However, the requirements on the advisory controller can be very different, such as those that involve human reaction time and features of the human–machine interface. Thus, controller synthesis must involve a *co-design* of controllers and human–machine interfaces.

In summary, the field of h-CPSs is a fertile ground for the CPS design automation community. There are several exciting directions for future work including human modeling, novel specification languages to capture requirements unique to h-CPS, data-driven verification and synthesis, quantitative verification and synthesis, and co-design of interfaces and control.

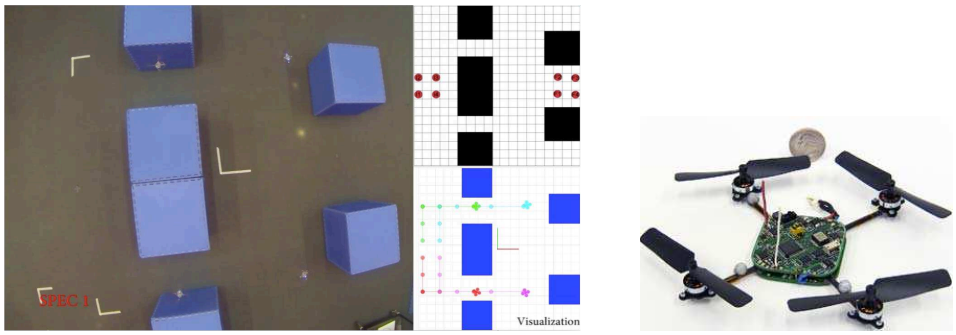
C. Component-Based Design With Contracts

The RTL design flow for digital circuits is one of the major success stories in electronic design automation. An important aspect of the RTL flow is its emphasis on *component-based design*. This methodology is applied at various levels of abstraction: high-level RTL source modules, a library of logic gates and state-holding elements, a technology library, etc. Component-based design has many benefits: reuse, clean interfaces, separation of concerns, etc. Naturally, the question arises: is there a similar component-based design approach for CPSs?

At present, the answer is a qualified "yes." The starting point is to construct the right component library for each application domain. Such a library must capture the the heterogeneous, cyber-physical, dynamic nature of that domain. While MBD languages such as Simulink/Stateflow and NI LabVIEW do offer component libraries, these are often too low-level and without cleanly-specified interfaces with precise semantics. Moreover, such libraries do not always accurately abstract relevant features of the underlying platform, such as timing behaviors.

Fortunately there are some emerging design methodologies that one can build upon. Platform-based design (PBD) [31]–[34] maps a top-down mapping of application-level constraints with a bottom-up propagation of platform constraints to find the right composition of platform components that meets an application's requirements. *Contract-based design* [2] complements the PBD methodology by adding a rigorous notion of formal contracts to ensure that composition of components maintains desired properties. These methodologies provide a framework for component-based design, provided one can come up with the right library of components, rules of composition, and interface contracts.

The challenge is thus shifted to finding the three Cs—components, composition, and contracts—for a given application. As of today, the process of finding these is very domain-specific. As an illustrative example, we discuss how component-based design has been successfully demonstrated for programming teams of robots to achieve coordinated tasks



(a) (b)

Fig. 2. Compositional SMT-driven multirobot motion planning. (a) Top view of sample execution and associated simulation. (b) Nano-quadrotor platform from KMel robotics [37] (reproduced from [35]).

in a laboratory setting [35]. The tasks are specified in a variant of *temporal logic* [36].

In robotics, the traditional motion planning problem is to move a robot from points A to B while avoiding obstacles. However, more recently, there is growing interest in extending this problem along 2-D. The first extension is to impose more complex requirements on the robot, such as visiting certain locations "infinitely often." Such requirements can be conveniently specified in a formal notation such as linear temporal logic (LTL). The second extension is to handle swarms of many robots executing coordinated plans. Such problems arise in many application settings, including persistent surveillance, search and rescue, formation control, and aerial imaging. More complex requirements require more sophisticated methods to ensure that the synthesized plans are provably correct. Scaling planning algorithms to larger swarms requires more efficient algorithms and design methodologies.

Recent work [35] addresses these challenges with a two-pronged approach. First, a compositional approach is employed, where precharacterized motion primitives, based on well-known control algorithms, are used as a component library. Each motion primitive is specified in a suitable combination of logical theories. Second, using an encoding similar to the one used for bounded model checking [38], a satisfiability modulo theories (SMTs) solver [39] is used to find a composition of motion primitives that achieves the desired LTL objectives. Fig. 2 depicts a sample result of this approach, showing the top view of four nano quadrotor robots achieving a desired LTL specification.

These results are only a first step. There are many more problems that remain to be solved, including inferring effective logical characterizations of motion primitives, handling dynamic, uncertain, and adversarial environments, dealing with nonlinear dynamics, incremental planning, and scaling up to an order of magnitude more robots. Even so, it is important to note that the initial demonstration is a successful realization of the PBD vision, where high-level robotics applications are mapped to compositions of motion primitives which are implemented in terms of platform-specific control algorithms. Exploring the full potential of component-based design for CPS remains an important challenge for the future.

D. Design for Security and Privacy

Security and privacy have become two of the foremost design concerns for CPSs today. Security, broadly speaking, is the state of being protected from harm. Privacy is the state of being kept away from observation. With embedded and CPSs being increasingly networked with each other and with the Internet, security and privacy concerns are now front and center for system designers.

There are two primary aspects that differentiate security and privacy from other design criteria for CPS. First, the operating environment is considered to be significantly more adversarial in nature than in typical system design. We refer to this aspect as the *threat model*. Second, the kinds of properties, specifying desired and undesired behavior, are also different from traditional system specifications (and often impose additional requirements on top of the traditional ones). We refer to this aspect as the *security/privacy goals*.

These two aspects are also the dimensions along with we can distinguish the research in CPS security and privacy from the more traditional field of cyber-security. We outline these dimensions below.

1) *Threat Models With Physical Characteristics:* CPS provides new attack surfaces that lead to new threat models that have not arisen in traditional cyber-security. One such class of threat models come under the category of *physical attacks.* These are attacks that observe or modify the physical processes in the system or its environment. Pure cyber-security approaches fail to model these physical processes and therefore miss these attacks. One example of physical attacks are those on sensors. Recent work has focused on investigating both threat models and countermeasures for attacks on analog sensors. A main mode of attack has been to employ electromagnetic interference (EMI) to modify the sensed signal. Two recent projects have studied EMI attacks in different applications. Foo Kune *et al.* [40] investigated EMI attacks at varying power and distances on implantable medical devices and consumer electronics. Shoukry *et al.* [41] studied the possibility of EMI attacks that spoof sensor values for certain types of automotive sensors. Countermeasures have also been developed for these attacks [40], [42], [43]. One of these countermeasures involves secure state estimation using a blend of SMT solving and convex optimization [43], pointing to the form of design automation engines that might be applicable. Another example involves *side channels attacks*, including, e.g., attacks that reveal secrets by observing physical properties of a system such as timing or power consumption. For a more detailed introductory exposition of this topic (see [1]).
2) *CPS Security/Privacy Goals:* The classes of properties considered for CPS security and privacy are similar to

those in traditional cyber-security: they involve *integrity*, *confidentiality*, *anonymity*, and *availability*. However, specific forms of these properties vary. For instance, in CPS security, one cares about ensuring control-theoretic properties such as stability under attacks. Similarly, one may consider differential privacy, but operating over data streams from reactive systems rather than tables of data stored in databases [44]. Another important aspect relates to the tradeoff between different properties. For instance, in automotive networks, one is concerned both with authentication of messages sent between electronic control units (ECUs) on the controller area network (CAN) bus (a security concern) and with real-time requirements (a timing concern). Traditional cryptographic protocols for authentication do not apply "as is," and one must design customized solutions that provide an appropriate tradeoff between those competing concerns [45].

We discuss in Section IV several specific instances of security and privacy problems in CPS. It is important to note that security and privacy have become cross-cutting concerns throughout the design process that must be considered from the very beginning of the design process; they cannot just be bolted on as an after-thought.

IV. Applications

We now discuss in more detail two application domains: 1) smart energy systems and 2) next-generation automotive systems. These domains are excellent representatives of CPS as they have a combination of all characteristics identified in Section II. For each domain, we first give a high-level motivation for the design problems in that domain, followed by a survey of some of the important problems along with proposed solution methods.

A. Smart Energy Systems

The design of smart energy systems spans across multiple layers, from developing power grids with intelligent energy generation, transmission and distribution, to constructing commercial buildings and residential homes with smart energy management schemes. It is an extremely challenging task, given the scale and heterogeneity of such systems and the stringent requirements on their performance, reliability, security and cost. Design automation methodologies and tools, such as the ones discussed in Section III, will be critical for addressing these challenges and achieving truly smart energy systems. Below we discuss some of those approaches, in particular for the design of smart buildings and homes.

1) Modeling and Design Automation: The traditional design methodology for large buildings is a top-down approach. Different building subsystems are designed in isolation by domain experts, the following design documents flown down after the bid process [46]. Such methodology, however, is not suitable for designing energy-efficient buildings, where the adoption of low energy solutions such as natural ventilation, active facade and advanced cooling control require a close interaction among architects, mechanical engineers, control engineers, and electrical and computer engineers. A new set of methodologies and tools is greatly needed to address the heterogeneous building subsystems in a holistic fashion and provide an automated design flow.

a) MBD flow and co-design: Yang *et al.* [46], [47] proposed an automated design flow for building automation and control systems. The flow leverages MBD tools such as Simulink [48] and Modelica [49] for modeling the heterogeneous subsystems, and then converts the models into a unified intermediate format and explore the design implementation.

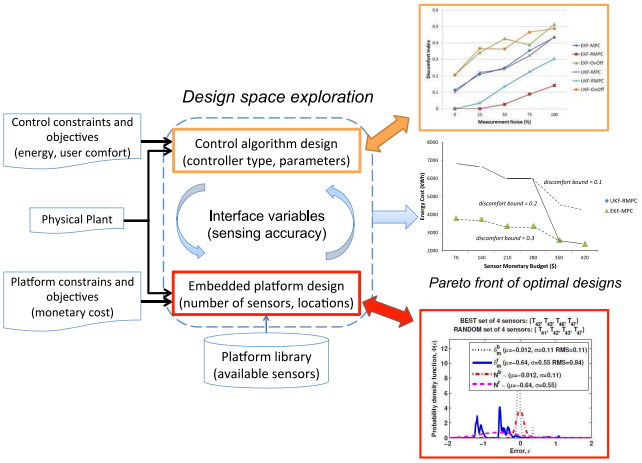


Fig. 3. Co-design of control algorithm and sensing platform for buildings.

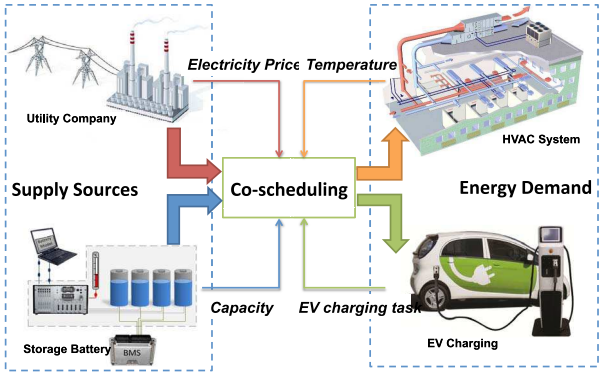


Fig. 4. Co-scheduling of energy supplies and demands for buildings.

Maasoumy *et al.* [50] presented an approach to *co-design* heating, ventilation, and air conditioning (HVAC) control algorithms and embedded sensing platforms through the concept of interface variables, as illustrated in Fig. 3 to reduce building energy consumption while meeting cost and occupancy comfort requirements. The work shows that the selection of HVAC control schemes significantly depends on the number, location and accuracy of the temperature sensors, and therefore necessitates the need for a co-design approach.

Wei *et al.* [51], [52] showed how to *co-schedule* heterogeneous energy demand types, including HVAC control and electric vehicles charging with heterogeneous energy supplies such as grid electricity and battery storage in a holistic model predictive control-based formulation, as shown in Fig. 4. The results show that such co-scheduling approach can effectively leverage the flexibility in building energy scheduling and significantly reduce energy consumption and peak demand. In these approaches, simplified RC network models are used to capture the thermal dynamics of building rooms and walls. Compared to more detailed models such as the ones used in the EnergyPlus tool from the Department of Energy [53], these simplified models provide the efficiency needed for design space exploration and runtime management.

Recently, the paradigm of contract-based design has been applied for smart buildings and their integration into the smart

grid [54], [55]. In particular, assume-guarantee contracts are formalized between the buildings and the grid to leverage the HVAC scheduling flexibility and optimize the ancillary service power flow from buildings.

b) Leveraging measurement data: There have been a number of approaches for calibrating building energy models based on real-time measurement data [56]–[60]. However, for detailed models (e.g., those in EnergyPlus or TRNSYS [61]), the calibration procedures could be quite labor-intensive and time-consuming [62]. In [63] and [64], a meta-model based approach is proposed to reduce the complexity of building energy models, which may then enable fast model calibration and efficient optimization of building design and operation.

Real-time sensor data has also been used with machine learning approaches for recognizing and predicting human activities in buildings [65]–[67]. Such information may then be leveraged for improving building energy efficiency [68]–[70], occupancy comfort, and safety and security.

2) Security and Privacy: For smart energy systems from individual buildings and homes to the entire grid, security and privacy have become a pressing concern. In below, we will discuss some of those challenges and proposed design automation solutions, in particular regarding pricing attacks and energy thefts.

a) Pricing attacks: The prevailing U.S. electricity market employs the dynamic electricity pricing scheme to guide the energy scheduling techniques. The basic idea is to set different electricity prices during different time intervals, with high prices at peak energy usage hours to discourage significant energy consumption at those times. Precisely, the predictive guideline pricing and the real-time pricing for billing customers are jointed deployed. The predictive guideline pricing provides an estimated price per time interval within the next 24 hours, while the real-time pricing computes the bill based on the recent actual energy consumption. The predictive guideline pricing is expected to match the real-time pricing, although this is often not the case in practice. Based on these pricing models, there are many automatic scheduling techniques developed in the literature. These include techniques based on dynamic programming [71], linear programming [72], mixed-integer linear programming [73], and game theoretic scheduling [74], [75].

A pricing guided scheduling framework may be vulnerable to security threats. Modern smart meters installed at homes and buildings are not merely measurement devices but also equipped with advanced operating systems that enable automatic scheduling of various appliances and devices. If the predictive guideline pricing seen at a smart meter is manipulated in a pricing attack, the smart home schedulers could make wrong scheduling decisions causing detrimental impacts. For instance, peak energy usage increase in the local community may potentially lead to blackouts [76]. Such negative impacts become quite significant when a wide range of smart meters are attacked, e.g., through malware propagation [77].

b) Energy thefts: The pricing attack hacks the inputs of smart meters. On the other hand, the outputs of smart meters, which are the measurements of energy consumption during a past time window, can also be manipulated. For example, if a smart meter only reports 10 KWh to the utility while it actually measures 100 KWh, the 90 KWh difference can be viewed as being stolen [78].

c) Detection methods using POMDP: The system level impacts of pricing attack and energy theft have been analyzed in several works such as [76] and [78]. The detections of those attacks are built upon the partially observable Markov decision process (POMDP) models. The simulation results in [76] indicate that POMDP-based detection can reduce the energy bill and peak-to-average ratio (which is a ratio indicating energy balance) by 59.3% and 62.3%, respectively, compared to a natural heuristic approach for pricing attack. Similarly, POMDP-based detection can reduce the bill increase by 78.3% while successfully detecting more than 90% energy theft [78]. Alternatively, sensors such as feeder remote terminal units can be inserted into the local power distribution network to improve the detection rate of energy theft [79], [80] when smart meters are assumed to be hacked independently. For the more general case, it would be interesting to investigate how sensor deployment can benefit the POMDP model if they are deployed in an interleaving fashion. It would be also interesting to analyze the attacks jointly performing pricing attack and energy theft.

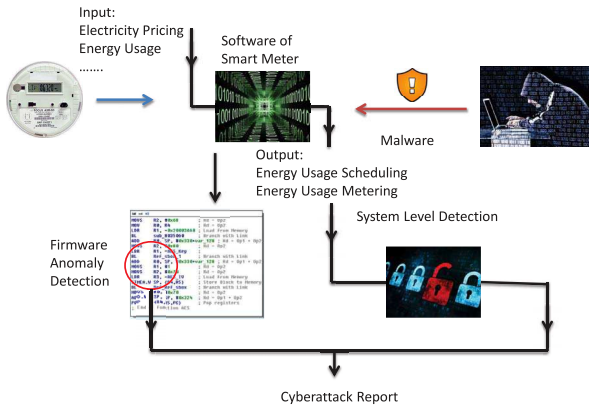


Fig. 5. Cross-layer protection against attacks.

d) New pricing frameworks: Furthermore, new pricing frameworks have been proposed to better leverage the scheduling flexibility at buildings and homes and increase the penetration of demand response. For instance, in [81], a proactive demand participation scheme calculates the building scheduling flexibility based on guideline pricing, and then captures such flexibility as demand-bid curves for grid-level optimization. As observed in [81] and [82], such scheme faces potential pricing attack on the guideline pricing and also possible manipulation on the demand-bid curves.

e) Cross-layer detection: Finally, it is worth noting that at least part of detection code for pricing attack or energy theft needs to be implemented on the smart meter, while the smart meter itself is hacked. Thus, to ensure the reliable execution of the detection code, cross-layer detection techniques would be desirable, as illustrated in Fig. 5. There has been little research in this domain, but this is certainly an interesting future research direction.

B. Next-Generation Automotive Systems

The design and implementation of automotive electronic systems have become increasingly challenging, with growing functional complexity in scale and features, as well as the adoption of more distributed and networked architectural platforms. From year 2000 to 2010, the automotive software development cost increased from 2% to 13% of a vehicle's total value [83], and the number of lines of code increased from 1 million to more than 10 million [33], [84], [85]. The number of ECUs in a standard car has gone from 20 to over 50 in the past decade [84]. The traditional *federated architecture*, where each function is deployed to one dedicated ECU,

is shifting to the *integrated architecture*, in which one function can be distributed over multiple ECUs and multiple functions can be supported by one ECU [86]. There is also the trend to deploy multicore ECUs to support growing functionality and reduce system cost (by reducing the number of ECUs in the system and their connection wires) [83]. These trends lead to significantly more sharing and contention among software functions over the architectural platform.

Moving forward, software and electronics will play a dominant role in vehicle innovation. Approximately 90% of automotive innovations in 2012 featured software and electronics, especially in active safety and infotainment systems [87], and it is predicted that this will continue to be the trend in the future given the rapid advances in autonomous driving technology. With this trend, the complexity of automotive electronic systems will continue to rise rapidly. This presents tremendous design and implementation challenges, and calls for *a new set of design automation methods and tools*.

1) Model-Based Design and Synthesis: MBD is today widely accepted as a key enabler to cope with complex system design due to its capabilities to support early design verification/validation through formal functional models [2], [88], [89]. Using these models, designers can capture complex control systems and the plant models they interact with, and conduct simulations to analyze system behavior and validate functional properties. Among many functional modeling tools, the Simulink/Stateflow toolset [90] is popular in the design of automotive electronic systems, and is based on the synchronous reactive (SR) semantics. There are other languages/tools based on SR models, such as Signal, Lustre, and Esterel [91].

One important aspect of model-based development is the capability to *synthesize correct and optimal implementations from high-level functional models*. As observed from the circuit design domain, a robust and efficient synthesis flow will greatly motivate the adoption of high-level models. For instance, the quality of logic synthesis tools propelled the adoption of RTL models while recently the advancement of high-level synthesis tools have raised the design abstraction to C/C++ in many cases.

However, synthesizing cyber-physical functional models to software and hardware implementations remains hindered by many challenges, in particular those related to system *timing* behavior. First, the *complexity of timing analysis* arises with the growing complexity and heterogeneity of automotive system functionality and architectural platform. Second, there is significant *uncertainty of timing behavior* resulting from dynamic physical environment, data input and embedded platform conditions, especially for active safety applications. Third, there are *diverse timing constraints* from different design metrics such as schedulability, control performance, extensibility and fault tolerance, some of which lead to conflicting requirements. For instance, shorter sampling periods and end-to-end latencies of control loops usually lead to better sensing and control performance [92], [93], but may be detrimental to schedulability, extensibility and security (as there is less timing slack for adding strong security techniques [45], [94]).

Current synthesis solutions and practices do not adequately address these timing challenges. Timing constraints are often set in an *ad-hoc* fashion without quantitative analysis of their impacts on multiple related metrics. Furthermore, the synthesis process is often conducted without continuous and holistic consideration of timing. For software implementation, while timing is usually considered during the mapping of software tasks onto hardware platforms, it is rarely addressed during the generation of software tasks from the initial functional models, and thereby leaving a significant gap in the synthesis process. Such issues may lead to infeasible solutions, long design cycles, and ultimately inferior and error-prone implementations.

To cope with these challenges, it is critical to develop *new design automation methods and tools* that address timing holistically throughout the synthesis process, consider timing uncertainty in computation and communication, analyze timing impact on various design metrics and leverage such analysis for design space exploration. In [95], algorithms are proposed for multitask generation of finite state machines with consideration of timing extensibility and robustness. In [96], a holistic synthesis flow is proposed for automotive software development with respect to schedulability, reusability, modularity, and memory usage. The synthesis flow explores the multitask generation of dataflow functional models and the mapping of generated tasks onto multicore platforms, with *explicit timing consideration throughout the synthesis process* based on a formulation of firing and execution timing automata. Novel execution time analysis techniques based on combining machine learning and formal symbolic analysis show significant promise and have been successfully demonstrated on automotive software [19], [97]. This collection of work demonstrates promise in addressing the timing challenges, and further motivate the development of new synthesis methodologies and algorithms for next-generation automotive systems.

2) Human-in-the-Loop Automotive Systems: One of the outstanding problems in vehicle automation is the *car-to-driver handoff* problem. This is the problem where the car has to disengage from an autonomous mode and the driver is required to regain control of the vehicle. According to the Department of Motor Vehicles (DMV),[2] such disengagements are defined as deactivations of the autonomous mode in a situation where "a failure of the autonomous technology is detected and requires the driver to take immediate manual control of the vehicle." A recent report published by Google indicates that during the operation of its self-driving cars in the period from September 24, 2014 through November 30, 2015, there were 272 "immediate manual control disengagements" [98]. These correspond to situations where the autonomous technology failed to maintain safe operation of the vehicle and needed to immediately hand over the control to the driver. These situations are particularly dangerous because the driver is out of the control loop and might be performing other tasks when a handoff is required. In fact, according to a recent study, drivers usually need 5–8 s in order to safely and comfortably perform takeover [99]. This stipulates that the design of a *human-in-the-loop* control system must take into account of human factors such as delays in response time.

A foundational challenge for design automation in addressing this problem is to find appropriate mathematical models that also incorporate human factors. Li *et al.* [28] formulated a human-in-the-loop controller as a composition of three agents—an autonomous controller, a human operator, and an advisory controller which determines whether the human or autonomous controller should be in control of the plant. Fig. 6 illustrates the structure of such a human-in-the-loop controller. In a situation when disengagement from the autonomous mode is necessary, the advisory controller will send the corresponding advisory a to some user interface (e.g., audio or video

[2]DMV's final statement of reasons.

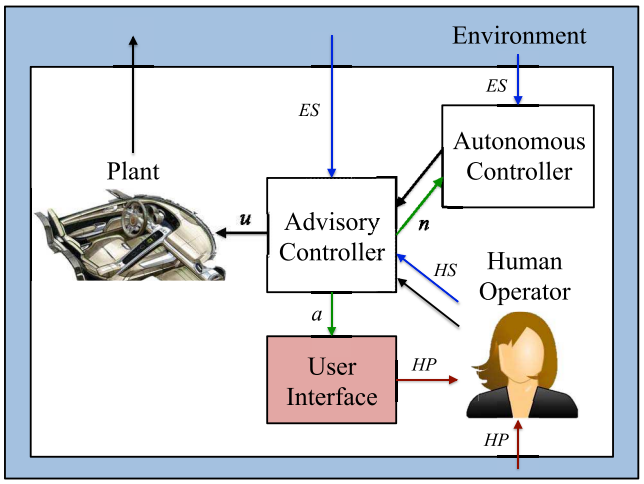


Fig. 6. Structure of a human-in-the-loop controller. *ES* denotes environment sensing. *HS* denotes human sensing. *HP* denotes human perception. *u* is the control input to the plant. *a* is the advisory issued by the advisory controller to the human operator. *n* is a notification signal from the advisory controller to the autonomous controller.

interface). Upon noticing this signal, the driver can take over control and her control inputs are passed to the vehicle. When the handoff is successful, the advisory controller notifies the autonomous part of the system by sending *n* that it is no longer controlling the plant. Between the time when the advisory is issued and the completion of the handoff, the autonomous controller is responsible for the safe operation of the vehicle.

Motivated by the definition of "limited self-driving automation" by the National Highway Traffic Safety Administration [100], four criteria are defined for this human-in-the-loop controller model corresponding to the meta-requirements described in Section III-B2. The design automation problem is then to synthesize such controllers satisfying these meta-requirements. Instead of modeling the driver explicitly, the synthesis algorithm considers specific human factors that are critical to the problem, i.e., driver response time. Li *et al.* [28] presented correct-by-construction approach to controller synthesis that follows the general theme of "temporal logic motion planning" [101]. The main idea is to use temporal logic to specify motion objectives and constraints, such as the vehicle should reach certain goal regions, and then derive a motion planner that satisfies these specifications using automata theory. A novel aspect of the synthesis algorithm [28] is that it identifies conditions when a car-to-driver handoff is necessary, uses these conditions to synthesize an advisory controller, and synthesizes an autonomous controller that ensures safe operation assuming the driver takes over within a certain response time.

Human actions can also be captured using probabilistic models. For example, Feng *et al.* [102] used Markov decision processes (MDPs) to represent human operators. Two abstractions are considered.

1) Human behaviors are assumed to be known probability distributions *a priori*.
2) Human actions are nondeterministic.

The human operator model is then composed with another MDP model of an unmanned aerial vehicle. Depending on the abstraction, operator-dependent optimal control protocols can be derived by casting the control synthesis problem into a stochastic two-player game. While the MDP formalism is a reasonable choice, assuming *a priori* knowledge of the probability distributions is unrealistic. Sadigh *et al.* [30], [103], [104] took a more data-driven approach to modeling human behavior. In early work [30], they show how experimental data collected from a driving simulator can be used to construct a discrete-time Markov chain (DTMC). Uncertainties intrinsic to the estimation of transition probabilities during the construction of the DTMC is captured by allowing the transition probabilities to lie in certain convex sets. Using an algorithm that efficiently checks properties expressed in Probabilistic Computation Tree Logic over these convex Markov models [29], the effects of different attention levels on the quality of driving are formally analyzed. In more recent work [103], [104], they model human drivers as rational agents optimizing their reward functions, learn those reward functions from data, and use the learned functions in synthesizing control for autonomous vehicles.

Dual to car-to-driver handoff is to have the autonomous controller intervene when the vehicle driven by a human driver is in trouble. An example framework is given by Vasudevan *et al.* [105], which divides this problem into two components. The first component predicts the vehicle's behaviors based on observations about the driver's pose and environment, and the second component uses this information to determine when the autonomous controller should intervene. Experimental evaluation using a car simulator shows that by incorporating information about driver pose in the construction, the semiautonomous controller outperforms one that merely treats the driver as a disturbance, including better accident prevention and not taking over control of the vehicle more often than necessary.

An important piece in the co-design with human in-the-loop is an effective communication interface between the human and the machine. Schirner *et al*. [106] outlined various kinds of interfaces and sensor technologies that can be used to augment a human's interaction with the physical world. Among these, context-aware sensing of human intent (*HS* in Fig. 6) and the design of an interface for shared governance are particularly relevant to semiautonomous systems. We envision a holistic framework that integrates human modeling, sensor technologies, human–machine interface, embedded system design, and formal reasoning for future design automation of human CPS.

3) Design for Security and Privacy: With increasing vehicle intelligence and connectivity, security and privacy have become pressing concerns for automotive systems. Koscher *et al.* [107] and Checkoway *et al.* [108] successfully compromised a production vehicle by hacking into its engine control system, brake control system, and other electronic components. The attacks are conducted through internal CAN buses using packet sniffing, targeted probing, fuzzing and reverse engineering. CAN is currently the most used protocol and, unfortunately, also the most attractive protocol for attackers [109], [110].

Several approaches have been proposed to add message authentication codes in CAN data frames to provide message authentication [111]–[115]. However, the limitations on CAN bus bandwidths and message lengths make it very challenging to embed security mechanisms without hindering safety and control applications, especially when the initial designs did not consider security [111]. Recently, time-triggered communication protocols such as FlexRay and TTEthernet are proposed to provide more predictable timing and higher bandwidth than CAN for automotive systems. In [116] and [117], low cost and flexible multicast authentication methods are proposed for time-triggered systems. In [118], authentication methods

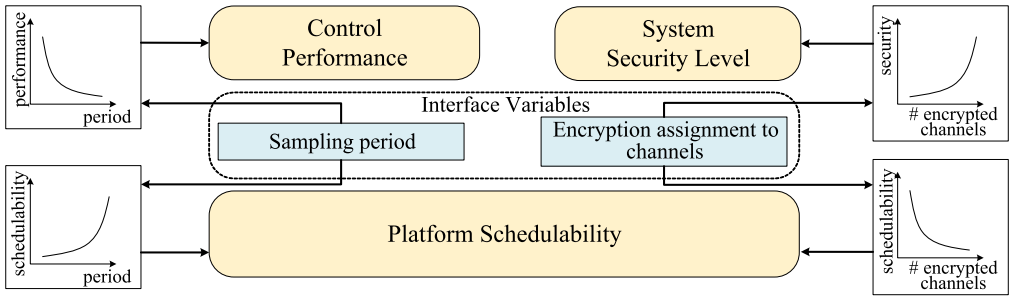


Fig. 7. Control and platform codesign for secure CPSs.

are proposed for time-triggered systems using time-delayed release of keys, based on a variant of the TESLA protocol [119], [120]. In [121], algorithms are proposed to optimize task allocation, priority assignment and network scheduling for time-triggered systems with time-delayed release of keys authentication. While these new protocols have more bandwidth and higher speed, adding security updates into existing designs still remains challenging and has complex impacts on various design metrics.

To cope with these challenges, it is critical to *quantitively address security from the beginning of the design process and together with other design objectives*. In [45], a set of algorithms is presented to address automotive security from the level of software tasks, i.e., by assuming a task graph is given and optimizing task allocation and scheduling with respect to security and schedulability. The results demonstrate the importance of considering security *during* the design process rather than trying to add security measurement as an afterthought. However, as stated before in Section IV-B1, the task graph abstraction does not contain important functional information that directly affect system security, control performance and other metrics. To effectively address the automotive security issue, the consideration has to start at the functional level.

In [94], a cross-layer design framework is proposed to combine control-theoretic methods at the functional layer and cybersecurity techniques at the embedded platform layer, and addresses security together with other design metrics such as control performance under resource and real-time constraints. As shown in Fig. 7, control performance and system security level are measured at the functional layer, while schedulability is analyzed at the embedded platform layer. To bridge these metrics, a set of *interface variables* are introduced, specifically the sampling period of every control task and the selection of messages to be encrypted. Intuitively, when the sampling period of a control task increases, its control performance decreases while platform schedulability increases with less frequent activation of the control task. On the other hand, when the number of messages being encrypted increases, the system security level increases while platform schedulability decreases because of the increased overhead. Furthermore, the sampling periods may have to increase for schedulability concern thereby worsening the control performance. These relations are quantitatively modeled in the codesign formulation in [94].

V. Conclusion

This paper has presented a view of the challenges and opportunities for design automation of CPSs. We repeat some of the key points here. In our opinion, the design challenges for today's CPS stem from the following combination of characteristics: *hybrid, heterogeneous, distributed, large-scale, dynamic, adaptive, and human-in-the-loop*. To design dependable and secure systems with these characteristics, we believe that we need design automation tools to have the following combination of features: *cross-domain, component-based, learning-based, time-aware, trust-aware, and human-centric*. We presented a sampling of recent efforts and opportunities, including combining MBD with data-driven learning, design automation for human CPS, component-based design methodologies, and design for CPS security and privacy. Motivating applications from the automotive, smart grid, and smart buildings domains illustrate these topics.

Will a durable design methodology, such as the RTL design flow, emerge for CPSs? It is hard to tell for sure, given the heterogeneity of CPS. However, the surest trend, at the moment, is the confluence of data-driven and MBD methods. It is our opinion that this trend holds the beginnings of an exciting future for the design automation of CPS.

Acknowledgment

The authors thank the anonymous reviewers for their comments. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA or the U.S. Government.

References

[1] E. A. Lee and S. A. Seshia, *Introduction to Embedded Systems: A Cyber-Physical Systems Approach*, 2nd ed. LeeSeshia.org, 2015. [Online]. Available: http://leeseshia.org

[2] A. Sangiovanni-Vincentelli, W. Damm, and R. Passerone, "Taming Dr. Frankenstein: Contract-based design for cyber-physical systems," *Eur. J. Control*, vol. 18, no. 3, pp. 217–238, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0947358012709433

[3] E. A. Lee, "CPS foundations," in *Proc. 47th Design Autom. Conf. (DAC)*, 2010, Anaheim, CA, USA, Jul. 2010, pp. 737–742.

[4] J. Sztipanovits *et al.*, "Toward a science of cyber–physical system integration," *Proc. IEEE*, vol. 100, no. 1, pp. 29–44, Jan. 2012.

[5] J. Sztipanovits, T. Bapty, S. Neema, X. D. Koutsoukos, and E. K. Jackson, "Design tool chain for cyber-physical systems: Lessons learned," in *Proc. 52nd Annu. Design Autom. Conf.*, San Francisco, CA, USA, Jun. 2015, pp. 1–6.

[6] S. Skogestad and I. Postlethwaite, *Multivariable Feedback Control: Analysis and Design*. Chichester, U.K.: Wiley, 2007.

[7] G. Nicolescu and P. J. Mosterman, *Model-Based Design for Embedded Systems*. Boca Raton, FL, USA: CRC Press, 2009.

[8] S. Sastry and M. Bodson, *Adaptive Control: Stability, Convergence and Robustness*. Dover Publications, 2011.

[9] K. J. Åström and B. Wittenmark, *Adaptive Control*. Dover Publications, 2008.

[10] N. Chambers, *Hands-Off Training: Google's Self-Driving Car Holds Tantalizing Promise, but Major Roadblocks Remain*, Sci. Amer., May 2011. [Online]. Available: http://www.scientificamerican.com/article/google-driverless-robot-car/

[11] T. M. Mitchell, *Machine Learning*. New York, NY, USA: McGraw-Hill, 1997.

[12] D. Angluin and C. H. Smith, "Inductive inference: Theory and methods," *ACM Comput. Surveys*, vol. 15, no. 3, pp. 237–269, Sep. 1983.

[13] S. Jha and S. A. Seshia, "A theory of formal synthesis via inductive learning," *ArXiv e-prints*, May 2015. [Online]. Available: https://arxiv.org/abs/1505.03953

[14] X. Jin, A. Donzé, J. V. Deshmukh, and S. A. Seshia, "Mining requirements from closed-loop control models," in *Proc. Int. Conf. Hybrid Syst. Comput. Control (HSCC)*, Philadelphia, PA, USA, Apr. 2013, pp. 43–52.

[15] W. Li, L. Dworkin, and S. A. Seshia, "Mining assumptions for synthesis," in *Proc. 9th ACM/IEEE Int. Conf. Formal Methods Models Codesign (MEMOCODE)*, Jul. 2011, pp. 43–50.

[16] V. Raman, A. Donzé, D. Sadigh, R. M. Murray, and S. A. Seshia, "Reactive synthesis from signal temporal logic specifications," in *Proc. 8th Int. Conf. Hybrid Syst. Comput. Control (HSCC)*, Seattle, WA, USA, Apr. 2015, pp. 239–248.

[17] S. Jha, S. A. Seshia, and A. Tiwari, "Synthesis of optimal switching logic for hybrid systems," in *Proc. Int. Conf. Embedded Softw. (EMSOFT)*, Taipei, Taiwan, Oct. 2011, pp. 107–116.

[18] S. A. Seshia and A. Rakhlin, "Game-theoretic timing analysis," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, San Jose, CA, USA, 2008, pp. 575–582.

[19] S. A. Seshia and A. Rakhlin, "Quantitative analysis of systems using game-theoretic learning," *ACM Trans. Embedded Comput. Syst.*, vol. 11, no. S2, 2012, Art. no. 55.

[20] J. Kapinski, J. V. Deshmukh, S. Sankaranarayanan, and N. Arechiga, "Simulation-guided Lyapunov analysis for hybrid dynamical systems," in *Proc. 17th Int. Conf. Hybrid Syst. Comput. Control (HSCC)*, Berlin, Germany, 2014, pp. 133–142.

[21] S. A. Seshia, "Sciduction: Combining induction, deduction, and structure for verification and synthesis," in *Proc. Design Autom. Conf. (DAC)*, San Francisco, CA, USA, Jun. 2012, pp. 356–365.

[22] S. A. Seshia, "Combining induction, deduction, and structure for verification and synthesis," *Proc. IEEE*, vol. 103, no. 11, pp. 2036–2051, Nov. 2015.

[23] D. Amodei *et al.*, "Concrete problems in AI safety," *ArXiv e-prints*, Jun. 2016. [Online]. Available: https://arxiv.org/abs/1606.06565

[24] S. A. Seshia, D. Sadigh, and S. S. Sastry, "Towards verified artificial intelligence," *ArXiv e-prints*, Jun. 2016. [Online]. Available: https://arxiv.org/abs/1606.08514

[25] *The Interfaces Between Flight Crews and Modern Flight Systems*, Federal Aviation Admin. (FAA), Washington, DC, USA, 1995. [Online]. Available: http://www.faa.gov/avr/afs/interfac.pdf

[26] L. T. Kohn, J. M. Corrigan, and M. S. Donaldson, Eds., "To err is human: Building a safer health system," Rep. Committee Qual. Health Care America, Inst. Med., Nat. Acad. Press, Washington, DC, USA, Tech. Rep., 2000.

[27] S. A. Seshia, D. Sadigh, and S. S. Sastry, "Formal methods for semi-autonomous driving," in *Proc. 52nd Annu. Design Autom. Conf. (DAC)*, San Francisco, CA, USA, 2015, pp. 1–5.

[28] W. Li, D. Sadigh, S. Sastry, and S. A. Seshia, "Synthesis of human-in-the-loop control systems," in *Proc. 20th Int. Conf. Tools Algorithms Construct. Anal. Syst. (TACAS)*, Apr. 2014.

[29] A. Puggelli, W. Li, A. Sangiovanni-Vincentelli, and S. A. Seshia, "Polynomial-time verification of PCTL properties of MDPs with convex uncertainties," in *Proc. 25th Int. Conf. Comput.-Aided Verification (CAV)*, St. Petersburg, Russia, Jul. 2013, pp. 527–542.

[30] D. Sadigh *et al.*, "Data-driven probabilistic modeling and verification of human driver behavior," in *Proc. Formal Verificat. Model. Human Mach. Syst. AAAI Spring Symp.*, Mar. 2014.

[31] K. Keutzer, A. R. Newton, J. M. Rabaey, and A. L. Sangiovanni-Vincentelli, "System-level design: Orthogonalization of concerns and platform-based design," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 19, no. 12, pp. 1523–1543, Dec. 2000.

[32] A. L. Sangiovanni-Vincentelli, L. P. Carloni, F. D. Bernardinis, and M. Sgroi, "Benefits and challenges for platform-based design," in *Proc. 41th Design Autom. Conf. (DAC)*, San Diego, CA, USA, Jun. 2004, pp. 409–414.

[33] A. Sangiovanni-Vincentelli, "Quo vadis, SLD? Reasoning about the trends and challenges of system level design," *Proc. IEEE*, vol. 95, no. 3, pp. 467–506, Mar. 2007.

[34] P. Nuzzo, A. L. Sangiovanni-Vincentelli, D. Bresolin, L. Geretti, and T. Villa, "A platform-based design methodology with contracts and related tools for the design of cyber-physical systems," *Proc. IEEE*, vol. 103, no. 11, pp. 2104–2132, Nov. 2015.

[35] I. Saha, R. Ramaithitima, V. Kumar, G. J. Pappas, and S. A. Seshia, "Automated composition of motion primitives for multi-robot systems from safe LTL specifications," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Chicago, IL, USA, Sep. 2014, pp. 1525–1532.

[36] A. Pnueli, "The temporal logic of programs," in *Proc. 18th Annu. Symp. Found. Comput. Sci. (FOCS)*, Washington, DC, USA, 1977, pp. 46–57.

[37] *KMel Robotics*. Accessed on Nov. 2016. [Online]. Available: http://kmelrobotics.com/

[38] A. Biere, A. Cimatti, E. M. Clarke, O. Strichman, and Y. Zhu, "Bounded model checking," *Adv. Comput.*, vol. 58, pp. 117–148, 2003.

[39] C. Barrett, R. Sebastiani, S. A. Seshia, and C. Tinelli, "Satisfiability modulo theories," in *Handbook of Satisfiability*, vol. 4, A. Biere, H. van Maaren, and T. Walsh, Eds. Amsterdam, The Netherlands: IOS Press, 2009, ch. 8.

[40] D. F. Kune *et al.*, "Ghost talk: Mitigating EMI signal injection attacks against analog sensors," in *Proc. 34th Annu. IEEE Symp. Security Privacy*, San Francisco, CA, USA, May 2013, pp. 145–159.

[41] Y. Shoukry, P. D. Martin, P. Tabuada, and M. B. Srivastava, "Non-invasive spoofing attacks for anti-lock braking systems," in *Proc. 15th Int. Workshop Cryptograph. Hardw. Embedded Syst. (CHES)*, Santa Barbara, CA, USA, 2013, pp. 55–72.

[42] Y. Shoukry *et al.*, "Secure state estimation under sensor attacks: A satisfiability modulo theory approach," in *Proc. Amer. Control Conf. (ACC)*, Chicago, IL, USA, Jul. 2015.

[43] Y. Shoukry *et al.*, "SMT-based observer design for cyber-physical systems under sensor attacks," in *Proc. Int. Conf. Cyber Phys. Syst. (ICCPS)*, Vienna, Austria, Apr. 2016, pp. 1–10.

[44] J. L. Ny and G. J. Pappas, "Differentially private filtering," *IEEE Trans. Autom. Control*, vol. 59, no. 2, pp. 341–354, Feb. 2014.

[45] C.-W. Lin, B. Zheng, Q. Zhu, and A. Sangiovanni-Vincentelli, "Security-aware design methodology and optimization for automotive systems," *ACM Trans. Design Autom. Electron. Syst.*, vol. 21, no. 1, pp. 1–18, Nov. 2015. [Online]. Available: http://doi.acm.org/10.1145/2803174

[46] Y. Yang, Q. Zhu, M. Maasoumy, and A. Sangiovanni-Vincentelli, "Development of building automation and control systems," *IEEE Des. Test Comput.*, vol. 29, no. 4, pp. 45–55, Aug. 2012.

[47] Y. Yang, A. Pinto, A. Sangiovanni-Vincentelli, and Q. Zhu, "A design flow for building automation and control systems," in *Proc. 31st IEEE Int. Real Time Syst. Symp. (RTSS)*, San Diego, CA, USA, 2010, pp. 105–116.

[48] *Simulink*. [Online]. Available: http://www.mathworks.com/products/simulink

[49] *Modelica*. [Online]. Available: https://www.modelica.org

[50] M. Maasoumy, Q. Zhu, C. Li, F. Meggers, and A. Sanqiovanni- Vincentelli, "Co-design of control algorithm and embedded platform for building HVAC systems," in *Proc. ACM/IEEE Int. Conf. Cyber Phys. Syst. (ICCPS)*, Philadelphia, PA, USA, Apr. 2013, pp. 61–70.

[51] T. Wei, Q. Zhu, and M. Maasoumy, "Co-scheduling of HVAC control, EV charging and battery usage for building energy efficiency," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, San Jose, CA, USA, Nov. 2014, pp. 191–196.

[52] T. Wei *et al.*, "Battery management and application for energy-efficient buildings," in *Proc. 51st ACM/EDAC/IEEE Design Autom. Conf. (DAC)*, San Francisco, CA, USA, Jun. 2014, pp. 1–6.

[53] *EnergyPlus: A Whole Building Energy Simulation Program*. [Online]. Available: https://energyplus.net

[54] B. Jin, P. Nuzzo, M. Maasoumy, Y. Zhou, and A. Sangiovanni-Vincentelli, "A contract-based framework for integrated demand response management in smart grids," in *Proc. 2nd ACM Int. Conf. Embedded Syst. Energy Efficient Built Environ.*, Seoul, South Korea, 2015, pp. 167–176.

[55] M. Maasoumy, P. Nuzzo, and A. Sangiovanni-Vincentelli, "Smart buildings in the smart grid: Contract-based design of an integrated energy management system," in *Cyber Physical Systems Approach to Smart Electric Power Grid*. Heidelberg, Germany: Springer, 2015, pp. 103–132.

[56] T. A. Reddy, "Literature review on calibration of building energy simulation programs: Uses, problems, procedures, uncertainty, and tools," *ASHRAE Trans.*, vol. 112, no. 2, pp. 226–240, 2006.

[57] G. Liu and M. Liu, "A rapid calibration procedure and case study for simplified simulation models of commonly used HVAC systems," *Build. Environ.*, vol. 46, no. 2, pp. 409–420, 2011.

[58] P. Raftery, M. Keane, and J. O'Donnell, "Calibrating whole building energy models: An evidence-based methodology," *Energy Build.*, vol. 43, no. 9, pp. 2356–2364, 2011.

[59] P. Raftery, M. Keane, and A. Costa, "Calibrating whole building energy models: Detailed case study using hourly measured data," *Energy Build.*, vol. 43, no. 12, pp. 3666–3679, 2011.

[60] J. Yoon, E.-J. Lee, and D. E. Claridge, "Calibration procedure for energy performance simulation of a commercial building," *J. Sol. Energy Eng.*, vol. 125, no. 3, pp. 251–257, 2003.

[61] *TRNSYS: An Energy Simulation Software Package*. [Online]. Available: http://sel.me.wisc.edu/trnsys

[62] Z. O'Neill and B. Eisenhower, "Leveraging the analysis of parametric uncertainty for building energy model calibration," *Build. Simulat.*, vol. 6, no. 4, pp. 365–377, 2013.

[63] B. Eisenhower, Z. O'Neill, S. Narayanan, V. A. Fonoberov, and I. Mezić, "A methodology for meta-model based optimization in building energy models," *Energy Build.*, vol. 47, pp. 292–301, Apr. 2012.

[64] B. Eisenhower, Z. O'Neill, V. A. Fonoberov, and I. Mezić, "Uncertainty and sensitivity decomposition of building energy models," *J. Build. Perform. Simulat.*, vol. 5, no. 3, pp. 171–184, 2012.

[65] A. Ridi, N. Zarkadis, C. Gisler, and J. Hennebert, "Duration models for activity recognition and prediction in buildings using hidden Markov models," in *Proc. IEEE Int. Conf. Data Sci. Adv. Anal. (DSAA)*, Paris, France, Oct. 2015, pp. 1–10.

[66] A. Khan *et al.*, "Occupancy monitoring using environmental & context sensors and a hierarchical analysis framework," in *Proc. 1st ACM Conf. Embedded Syst. Energy Efficient Build.*, Memphis, TN, USA, 2014, pp. 90–99.

[67] C. Sandels, J. Widén, and L. Nordström, "Simulating occupancy in office buildings with non-homogeneous Markov chains for demand response analysis," in *Proc. Power Energy Soc. Gen. Meeting*, Denver, CO, USA, Jul. 2015, pp. 1–5.

[68] J. R. Dobbs and B. M. Hencey, "Predictive HVAC control using a Markov occupancy model," in *Proc. Amer. Control Conf. (ACC)*, Portland, OR, USA, Jun. 2014, pp. 1057–1062.

[69] J. L. G. Ortega, L. Han, N. Whittacker, and N. Bowring, "A machine-learning based approach to model user occupancy and activity patterns for energy saving in buildings," in *Proc. Sci. Inf. Conf. (SAI)*, London, U.K., Jul. 2015, pp. 474–482.

[70] M. Behl and R. Mangharam, "Sometimes, money does grow on trees: Data-driven demand response with DR-advisor," in *Proc. 2nd ACM Int. Conf. Embedded Syst. Energy Efficient Built Environ. (BuildSys)*, Seoul, South Korea, Nov. 2015, pp. 137–146.

[71] L. Liu, Y. Liu, L. Wang, A. Zomaya, and S. Hu, "Economical and balanced energy usage in the smart home infrastructure: A tutorial and new results," *IEEE Trans. Emerg. Topics Comput.*, vol. 3, no. 4, pp. 556–570, Dec. 2015.

[72] X. Chen, T. Wei, and S. Hu, "Uncertainty-aware household appliance scheduling considering dynamic electricity pricing in smart home," *IEEE Trans. Smart Grid*, vol. 4, no. 2, pp. 932–941, Jun. 2013.

[73] S.-J. Kim and G. B. Giannakis, "Scalable and robust demand response with mixed-integer constraints," *IEEE Trans. Smart Grid*, vol. 4, no. 4, pp. 2089–2099, Dec. 2013.

[74] A.-H. Mohsenian-Rad, V. W. S. Wong, J. Jatskevich, R. Schober, and A. Leon-Garcia, "Autonomous demand-side management based on game-theoretic energy consumption scheduling for the future smart grid," *IEEE Trans. Smart Grid*, vol. 1, no. 3, pp. 320–331, Dec. 2010.

[75] Y. Liu *et al.*, "Game-theoretic market-driven smart home scheduling considering energy balancing," *IEEE Syst. J.*, to be published.

[76] Y. Liu, S. Hu, and T.-Y. Ho, "Leveraging strategic detection techniques for smart home pricing cyberattacks," *IEEE Trans. Depend. Secure Comput.*, vol. 13, no. 2, pp. 220–235, Mar./Apr. 2016.

[77] Y. Guo, C.-W. Ten, S. Hu, and W. W. Weaver, "Preventive maintenance for advanced metering infrastructure against malware propagation," *IEEE Trans. Smart Grid*, vol. 7, no. 3, pp. 1314–1328, May 2016.

[78] Y. Liu and S. Hu, "Cyberthreat analysis and detection for energy theft in social networking of smart homes," *IEEE Trans. Comput. Soc. Syst.*, vol. 2, no. 4, pp. 148–158, Dec. 2015.

[79] C. Liao, C.-W. Chen, and S. Hu, "Strategic FRTU deployment considering cybersecurity in secondary distribution network," *IEEE Trans. Smart Grid*, vol. 4, no. 3, pp. 1264–1274, Sep. 2013.

[80] Y. Zhou, X. Chen, A. Y. Zomaya, L. Wang, and S. Hu, "A dynamic programming algorithm for leveraging probabilistic detection of energy theft in smart home," *IEEE Trans. Emerg. Topics Comput.*, vol. 3, no. 4, pp. 502–513, Dec. 2015.

[81] T. Wei, Q. Zhu, and N. Yu, "Proactive demand participation of smart buildings in smart grid," *IEEE Trans. Comput.*, vol. 65, no. 5, pp. 1392–1406, May 2016.

[82] T. Wei, B. Zheng, Q. Zhu, and S. Hu, "Security analysis of proactive participation of smart buildings in smart grid," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, Austin, TX, USA, Nov. 2015, pp. 465–472.

[83] Q. Zhu and P. Deng, "Design synthesis and optimization for automotive embedded systems," in *Proc. Int. Symp. Phys. Design (ISPD)*, Petaluma, CA, USA, 2014, pp. 141–148. [Online]. Available: http://doi.acm.org/10.1145/2560519.2565873

[84] R. N. Charette, "This car runs on code," *IEEE Spectr.*, vol. 46, no. 3, p. 3, Feb. 2009.

[85] J. P. MacDuffie and T. Fujimoto, "Why dinosaurs will keep ruling the auto industry," *Harvard Bus. Rev.*, vol. 88, no. 6, pp. 23–25, 2010.

[86] M. Di Natale and A. L. Sangiovanni-Vincentelli, "Moving from federated to integrated architectures in automotive: The role of standards, methods and tools," *Proc. IEEE*, vol. 98, no. 4, pp. 603–620, Apr. 2010.

[87] *The Road to 2020 and Beyond: What's Driving the Global Automotive Industry*? McKinsey&Company, New York, NY, USA, Sep. 2013.

[88] *General Motors Developed Two-Mode Hybrid Powertrain With Mathworks Model-Based Design; Cut 24 Months Off Expected Dev Time*. Accessed on Nov. 2016. [Online]. Available: http://www.greencarcongress.com

[89] *Automakers Opting for Model-Based Design*. Accessed on Nov. 2016. [Online]. Available: http://www.designnews.com

[90] M. Stigge, P. Ekberg, N. Guan, and W. Yi, "The digraph real-time task model," in *Proc. 17th IEEE Real Time Embedded Technol. Appl. Symp. (RTAS)*, Chicago, IL, USA, Apr. 2011, pp. 71–80.

[91] A. Benveniste *et al.*, "The synchronous languages 12 years later," *Proc. IEEE*, vol. 91, no. 1, pp. 64–83, Jan. 2003.

[92] D. Seto, J. P. Lehoczky, L. Sha, and K. G. Shin, "On task schedulability in real-time control systems," in *Proc. 17th IEEE Real Time Syst. Symp.*, Washington, DC, USA, Dec. 1996, pp. 13–21.

[93] E. Bini and A. Cervin, "Delay-aware period assignment in control systems," in *Proc. Real Time Syst. Symp.*, Barcelona, Spain, Nov./Dec. 2008, pp. 291–300.

[94] B. Zheng, P. Deng, R. Anguluri, Q. Zhu, and F. Pasqualetti, "Cross-layer codesign for secure cyber-physical systems," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 35, no. 5, pp. 699–711, May 2016.

[95] Q. Zhu, P. Deng, M. D. Natale, and H. Zeng, "Robust and extensible task implementations of synchronous finite state machines," in *Proc. 16th IEEE/ACM Conf. Design Autom. Test Europe (DATE)*, Grenoble, France, 2013, pp. 1319–1324.

[96] P. Deng, F. Cremona, Q. Zhu, M. D. Natale, and H. Zeng, "A model-based synthesis flow for automotive CPS," in *Proc. ACM/IEEE Int. Conf. Cyber Phys. Syst. (ICCPS)*, Seattle, WA, USA, Apr. 2015, pp. 198–207.

[97] J. P. Kotker, "The internals of gametime: Implementation and evaluation of a timing analyzer for embedded software," Dept. Elect. Eng. Comput. Sci., Univ. California at Berkeley, Berkeley, CA, USA, Tech. Rep. UCB/EECS-2013-116, May 2013.

[98] (Dec. 2015). *Google Self-Driving Car Testing Report on Disengagements of Autonomous Mode*. https://static.googleusercontent.com/media/www.google.com/en//selfdrivingcar/files/reports/report-annual-15.pdf

[99] B. K.-J. Mok *et al.*, "Timing of unstructured transitions of control in automated driving," in *Proc. IEEE Intell. Veh. Symp. (IV)*, Jun./Jul. 2015, pp. 1167–1172.

[100] *Preliminary Statement of Policy Concerning Automated Vehicles*, Nat. Highway Traffic Safety Admin., Washington, DC, USA, 2013. [Online]. Available: http://www.nhtsa.gov/staticfiles/rulemaking/pdf/Automated_Vehicles_Policy.pdf

[101] G. E. Fainekos, H. Kress-Gazit, and G. J. Pappas, "Temporal logic motion planning for mobile robots," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Barcelona, Spain, 2005, pp. 2020–2025.

[102] L. Feng, C. Wiltsche, L. Humphrey, and U. Topcu, "Controller synthesis for autonomous systems interacting with human operators," in *Proc. ACM/IEEE 6th Int. Conf. Cyber-Phys. Syst. (ICCPS)*, Seattle, WA, USA, 2015, pp. 70–79.

[103] D. Sadigh, S. Sastry, S. A. Seshia, and A. D. Dragan, "Planning for autonomous cars that leverages effects on human actions," in *Proc. Robot. Sci. Syst. Conf. (RSS)*, Ann Arbor, MI, USA, Jun. 2016.

[104] D. Sadigh, S. S. Sastry, S. A. Seshia, and A. D. Dragan, "Information gathering actions over human internal state," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Daejeon, South Korea, Oct. 2016.

[105] R. Vasudevan *et al.*, "Safe semi-autonomous control with enhanced driver modeling," in *Proc. Amer. Control Conf. (ACC)*, Montreal, QC, Canada, Jun. 2012, pp. 2896–2903.

[106] G. Schirner, D. Erdogmus, K. Chowdhury, and T. Padir, "The future of human-in-the-loop cyber-physical systems," *Computer*, vol. 46, no. 1, pp. 36–45, Jan. 2013.

[107] K. Koscher *et al.*, "Experimental security analysis of a modern automobile," in *Proc. IEEE Symp. Security Privacy (S&P)*, Berleley, CA, USA, 2010, pp. 447–462.

[108] S. Checkoway *et al.*, "Comprehensive experimental analyses of automotive attack surfaces," in *Proc. USENIX Security Symp.*, San Francisco, CA, USA, 2011.

[109] F. Koushanfar, A.-R. Sadeghi, and H. Seudie, "EDA for secure and dependable cybercars: Challenges and opportunities," in *Proc. 49th ACM/EDAC/IEEE Design Autom. Conf. (DAC)*, San Francisco, CA, USA, 2012, pp. 220–228.

[110] T. Hoppe, S. Kiltz, and J. Dittmann, "Security threats to automotive CAN networks—Practical examples and selected short-term countermeasures," in *Proc. 27th Int. Conf. Comput. Safety Rel. Security (SAFECOMP)*, Newcastle upon Tyne, U.K., 2008, pp. 235–248. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-87698-4_21

[111] C.-W. Lin, Q. Zhu, C. Phung, and A. L. Sangiovanni-Vincentelli, "Security-aware mapping for CAN-based real-time distributed automotive systems," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, San Jose, CA, USA, 2013, pp. 115–121.

[112] D. K. Nilsson, U. Larson, and E. Jonsson, "Efficient in-vehicle delayed data authentication based on compound message authentication codes," in *Proc. IEEE 68th Veh. Technol. Conf. (VTC-Fall)*, Calgary, AB, Canada, 2008, pp. 1–5.

[113] B. Groza, P.-S. Murvay, A. van Herrewege, and I. Verbauwhede, "LiBrA-CAN: A lightweight broadcast authentication protocol for controller area networks," in *Proc. 11th Int. Conf. Cryptol. Netw. Security (CANS)*, Darmstadt, Germany, 2012, pp. 185–200.

[114] C. J. Szilagyi, "Low cost multicast network authentication for embedded control systems," Ph.D. dissertation, Dept. Elect. Comput. Eng., Carnegie Mellon Univ., Pittsburgh, PA, USA, 2012.

[115] A. V. Herrewege, D. Singelee, and I. Verbauwhede, "CANAuth—A simple, backward compatible broadcast authentication protocol for CAN bus," in *Proc. Workshop Embedded Security Cars*, 2011.

[116] C. Szilagyi and P. Koopman, "Low cost multicast authentication via validity voting in time-triggered embedded control networks," in *Proc. 5th Workshop Embedded Syst. Security (WESS)*, Scottsdale, AZ, USA, 2010, Art. no. 10. [Online]. Available: http://doi.acm.org/10.1145/1873548.1873558

[117] C. Szilagyi and P. Koopman, "Flexible multicast authentication for time-triggered embedded control network applications," in *Proc. IEEE/IFIP Int. Conf. Depend. Syst. Netw. (DSN)*, Lisbon, Portugal, Jul. 2009, pp. 165–174.

[118] A. Wasicek, C. El-Salloum, and H. Kopetz, "Authentication in time-triggered systems using time-delayed release of keys," in *Proc. 14th IEEE Int. Symp. Object/Compon./Service Oriented Real Time Distrib. Comput. (ISORC)*, Newport Beach, CA, USA, 2011, pp. 31–39.

[119] A. Perrig, R. Canetti, J. D. Tygar, and D. Song, "Efficient authentication and signing of multicast streams over lossy channels," in *Proc. IEEE Symp. Security Privacy (S&P)*, 2000, pp. 56–73.

[120] A. P. Ran, R. Canetti, D. Song, and J. D. Tygar, "Efficient and secure source authentication for multicast," in *Proc. Netw. Distrib. Syst. Security Symp. (NDSS)*, San Diego, CA, USA, 2001, pp. 35–46.

[121] C.-W. Lin, Q. Zhu, and A. Sangiovanni-Vincentelli, "Security-aware mapping for TDMA-based real-time distributed systems," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, San Jose, CA, USA, Nov. 2014, pp. 24–31.

Sanjit A. Seshia (S'99–M'05–SM'11) received the B.Tech. degree in computer science and engineering from the Indian Institute of Technology Bombay, Mumbai, India, in 1998, and the M.S. and Ph.D. degrees in computer science from Carnegie Mellon University, Pittsburgh, PA, USA, in 2000 and 2005, respectively.

He is currently a Professor with the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, Berkeley, CA, USA. His Ph.D. thesis work on the UCLID verifier and decision procedure helped pioneer the area of satisfiability modulo theories (SMTs) and SMT-based verification. He has co-authored a widely used textbook on embedded systems. He led the offering of a massive open online course on cyber-physical systems for which his group developed novel virtual laboratory auto-grading technology based on formal methods. His current research interests include dependable computing and computational logic, with a current focus on applying automated formal methods to problems in embedded and cyber-physical systems, electronic design automation, and computer security.

Prof. Seshia has received the Presidential Early Career Award for Scientists and Engineers from the White House, the Alfred P. Sloan Research Fellowship, the Prof. R. Narasimhan Lecture Award, the Frederick Emmons Terman Award for contributions to electrical engineering and computer science education, and the School of Computer Science Distinguished Dissertation Award at Carnegie Mellon University. He has served as an Associate Editor of the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS, and the Co-Chair of the Program Committee of the International Conference on Computer-Aided Verification in 2012.

Shiyan Hu (SM'10) received the Ph.D. degree in computer engineering from Texas A&M University, College Station, TX, USA, in 2008.

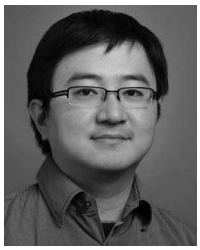
He is an Associate Professor with Michigan Technological University, Houghton, MI, USA, where he is the Director of the Center for Cyber-Physical Systems and the Associate Director of the Institute of Computer and Cybersystems. He has been a Visiting Professor with IBM Research, Austin, TX, USA, in 2010, and a Visiting Associate Professor with Stanford University, Stanford, CA, USA, from 2015 to 2016. His current research interests include cyber-physical systems, cybersecurity, computer-aided design of very large-scale integration circuits, and embedded systems. He has published over 100 refereed papers in the above areas.

Dr. Hu has received the National Science Foundation CAREER Award, the ACM SIGDA Richard Newton DAC Scholarship (as the faculty advisor), and the JSPS Faculty Invitation Fellowship. He is the Editor-in-Chief of *IET Cyber-Physical Systems: Theory and Applications*. He serves as an Associate Editor for the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS, the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, and the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS. He is also a Guest Editor of seven IEEE/ACM TRANSACTIONS, such as the IEEE TRANSACTIONS ON COMPUTERS and the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS. He is an ACM Distinguished Speaker, an IEEE Computer Society Distinguished Visitor, and an invited participant for the U.S. National Academy of Engineering Frontiers of Engineering Symposium. He is the Chair of the IEEE Technical Committee on Cyber-Physical Systems. He has served as the General Chair, the TPC Chair, the TPC Track Chair, and a TPC Member for numerous conferences. He is a fellow of IET.

Wenchao Li (S'08–M'15) received the M.S. and Ph.D. degrees in electrical engineering and computer sciences from the University of California at Berkeley (UC Berkeley), Berkeley, CA, USA, in 2013.

He is currently an Assistant Professor of Electrical and Computer Engineering with Boston University, Boston, MA, USA. Prior to joining Boston University, he was with the Computer Science Laboratory, SRI International, Menlo Park, CA, USA, from 2013 to 2016. His current research interests include human cyber-physical systems, formal methods, design automation, and machine learning.

Dr. Li has received the ACM SIGDA Outstanding Ph.D. Dissertation Award in 2015 and the Leon O. Chua Award from the Electrical Engineering and Computer Sciences Department, UC Berkeley, in 2013. He currently serves on the Program Committees of DATE and NASA Formal Methods conference.



Qi Zhu (M'12) received the B.E. degree in computer science from Tsinghua University, Beijing, China, in 2003, and the Ph.D. degree in electrical engineering and computer sciences from the University of California at Berkeley, Berkeley, CA, USA, in 2008.

He is an Assistant Professor of Electrical and Computer Engineering with the University of California at Riverside (UCR), Riverside, CA, USA. Prior to joining UCR, he was a Research Scientist with the Strategic CAD Laboratories, Intel, Santa Clara, CA, USA, from 2008 to 2011. His current research interests include model-based design and software synthesis for cyber-physical systems, CPS security, energy-efficient buildings and infrastructures, and system-on-chip design.

Dr. Zhu has received the 2016 CAREER Award from the National Science Foundation, and the Best Paper Awards of *ACM Transactions on Design Automation of Electronic Systems* in 2016, the 2013 International Conference on Cyber-Physical Systems 2013, and the 2006 and 2007 Design Automation Conference.