Formal Methods for Semi-Autonomous Driving

Sanjit A. Seshia Dorsa Sadigh S. Shankar Sastry
UC Berkeley
{sseshia,dsadigh,sastry}@eecs.berkeley.edu

ABSTRACT

We give an overview of the main challenges in the specification, design, and verification of human cyber-physical systems, with a special focus on semi-autonomous vehicles. We identify unique characteristics of formal modeling, specification, verification and synthesis in this domain. Some initial results and design principles are presented along with directions for future work.

Categories and Subject Descriptors

B.5.2 [**Design Aids**]: Verification; I.2.2 [**Automatic Programming**]: Program Synthesis; I.2.8 [**Control Methods**]: Control Theory

General Terms

Algorithms, Verification, Learning, Design

Keywords

Formal verification, synthesis, control, cyber-physical systems, automotive systems, semi-autonomous driving

1. INTRODUCTION

Formal methods is a field of computer science and engineering concerned with the rigorous mathematical specification, design, and verification of systems [20, 5]. The essence of formal methods comes down to *proof*: (i) formulating proof obligations in terms of formal *specifications* and *models*; (ii) *verifying*, via algorithmic proof search, that a designed system meets its specifications, and (iii) algorithmically *synthesizing* all or parts of a system so as to satisfy its specification. The field has made enormous advances in the past few decades. Techniques such as model checking [3, 18, 4] and theorem proving (see, e.g. [16, 9, 7]) are used routinely in the computer-aided design of integrated circuits and have been widely applied to find bugs in software, analyze embedded systems, and find security vulnerabilities. At the heart of these advances are computational proof engines such as Boolean satisfiability solvers [14], Binary Decision Diagrams (BDDs) [2], and satisfiability modulo theories (SMT) solvers [1].

A particularly compelling application domain for formal methods is the the field of cyber-physical systems. *Cyber-physical systems* (CPS) are computational systems that are tightly integrated with the physical world. (An introduction to the area may be found in a recent textbook [11].) Depending on the characteristics of CPS that are emphasized, they are also variously termed as *embedded systems*, the *Internet of Things* (IoT), the *Internet of Everything* (IoE), or the *Industrial Internet*. 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. While CPS have existed for long, it is only recently that the area has come together as an intellectual discipline. Many CPS operate in safety-critical or mission-critical settings, and therefore it is important to gain assurance that they will operate correctly, as per specification. Thus, formal methods are essential for the design of CPS.

Several cyber-physical systems are *interactive*, i.e., they interact with one or more human beings, and the human operators' 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 cyber-physical system* (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 (e.g., [6, 10]). Correct operation of these systems depends crucially on two design aspects: (i) *interfaces* between human operator(s) and autonomous components, and (ii) *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 "driver assistance systems") 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, semi-autonomous 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 semi-autonomous driving is a fertile application area for formal methods.

In this paper, we give an overview of the main challenges associated with the principled design of h-CPS, with a special focus on semi-autonomous driving, including:

- *Modeling:* What distinguishes a model of a h-CPS from a typical CPS?
- *Specification:* How do the requirements change for a h-CPS?
- *Verification:* What new verification problems arise from the human aspect?
- *Synthesis:* What advances in controller synthesis are required for h-CPS?

We also review some of the work in this area by the authors and colleagues, including especially the papers by Li et al. [13] and Sadigh et al. [19].

DAC 2015, San Francisco, California, USA.

http://dx.doi.org/10.1145/2744769.2747927 .

2. MODELING

A first step in applying formal methods is to build mathematical models of all components of the system. In this section, we outline our view of the unique aspects of the modeling task for h-CPS. Rather than espousing any particular modeling formalism (e.g., ordinary differential equations, finite-state machines, hybrid automata, etc.), we focus on elucidating the differences with modeling fully-autonomous systems even when the underlying mathematical formalism is the same.

A model of a cyber-physical system with fully-autonomous control typically comprises three mathematical entities: the *plant* being controlled, the autonomous *controller*, and the *environment* in which they operate. The task of the controller is to ensure that the plant behaves as per specification in the operating environment.

The key difference with an h-CPS is that, in an h-CPS, we additionally have the human operator(s) with whom control must be shared. Therefore, the model must contain a representation of the human operator(s) as well as a sub-system that mediates between the human operator(s) and the autonomous controller.

We propose an approach that models an h-CPS as a composition of five types of entities (components) [13], as illustrated in Fig. 1. The first is the *plant*, the entity being controlled. In the case of automobiles, these are the sub-systems that perform the various driving maneuvers under either manual or automatic control. The second is the *human operator* (or operators); i.e., the driver in an automobile. For simplicity, this discussion uses a single human operator, denoting her by HUMAN. The third entity is the *environment*. HUMAN perceives the environment around her and takes actions based on this perception and an underlying behavior model. We denote by HP HUMAN's perception of the environment and by HA the actions by HUMAN to control the plant. In the case of

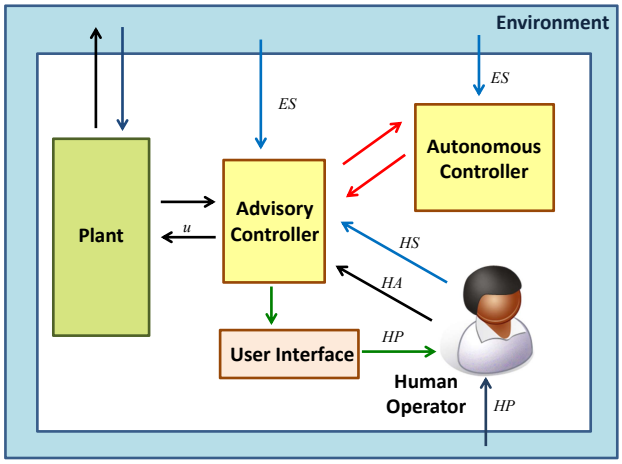

Figure 1: Structure of a Human Cyber-Physical System. ES denotes environment sensing, HS denotes human sensing, HP denotes human perception, HA denotes human actions, and u denotes the vector of control inputs to the plant.

a fully-autonomous system, the human operator is replaced by an *autonomous controller* (AUTO, for short). In practice, each specialized function of the system may be served by a separately designed autonomous controller; however, for conciseness these can be modeled as a single autonomous controller component. AUTO perceives the environment through sensors (denoted by ES, "environment sensors") and provides control input to the plant.

The distinctive aspect of h-CPS arises from its partial autonomy. We capture this by including a fifth component, the *advisory controller* (ADVISOR, for short) [13]. The function of ADVISOR is to mediate between HUMAN and AUTO. This mediation can take different forms. For instance, ADVISOR may decide when to switch from full control by HUMAN to full control by AUTO, or vice-versa. ADVISOR may also decide to combine the control inputs from HUMAN and AUTO to the plant in a systematic way that achieves design requirements. This is indicated in Fig. 1 by the yellow box adjoining plant, between it and the AUTO and HUMAN components. We note that for legal and policy reasons, it may not be possible in many applications (including driving) for ADVISOR to always take decisions that override HUMAN. It is for this reason that we use the term ADVISOR, indicating that the form of control exercised by ADVISOR may, in some situations, only provide suggestions to HUMAN as to the best course of action.

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). For this reason, formal methods has, for the most part, steered clear of problems that involve human modeling, with a common criticism being that such models can "never be precise." On the other hand, as George Box wrote, "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. In the case of driving, such data can be collected from field tests or from hardware-based car simulators (e.g., the Force Dynamics 401CR) that give the human driver a realistic feel for the self-driving features. In both cases, the set up must be instrumented with a variety of sensors to capture human action and perception.

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

- The use of *data-driven* human modeling;
- The inclusion of relevant aspects of the *human-machine interface*, and
- The presence of the *advisory controller*.

3. SPECIFICATION

Human CPS (h-CPS) have certain unique requirements which need to be formalized as formal specifications for verification and control. We focus here on the case of semi-autonomous driving, but the concepts are more generally applicable.

Recognizing both the safety issues and the potential benefits of vehicle automation, in 2013 the U.S. National Highway Traffic Safety Administration (NHTSA) published a statement that provides descriptions and guidelines for the continual development of these technologies [15]. Particularly, the statement defines five levels of automation ranging from vehicles without any control systems automated (Level 0) to vehicles with full automation (Level 4). We focus on Level 3 which describes a mode of automation that requires only limited driver control:

> *"Level 3 - Limited Self-Driving Automation: Vehicles at this level of automation enable the driver to cede full control of all safety-critical functions under certain traffic or environmental conditions and in those conditions to rely heavily on the vehicle to monitor for changes in those conditions requiring transition back to driver control. The driver is expected to be available for occasional control, but with sufficiently comfortable transition time. The vehicle is designed to ensure safe operation during the automated driving mode."* [15]

Essentially, this mode of automation stipulates that the human driver can act as a fail-safe mechanism and requires the driver to take over control should something go wrong. The challenge, however, lies in identifying the complete set of conditions under which the human driver has to be notified ahead of time. Based on the NHTSA statement, we have identified [13] four important criteria required for a human-in-the-loop controller to achieve this level of automation.

1. *Effective Monitoring*. The advisory controller should be able to monitor all information about the h-CPS and its environment needed to determine if human intervention is needed. This is a requirement on the types of sensors required and their quality and performance.
2. *Conditional Correctness*. When the autonomous controller is in control (and not the human operator) the system must satisfy a given formal specification (e.g., provided in temporal logic). This is therefore a traditional correctness requirement that is conditional in the sense that it applied only when the autonomous controller is in charge.
3. *Prescience*. The advisory controller must determine if the above formal specification may be violated ahead of time, and issues an advisory to the human operator in such a way that she has sufficient time to respond. This requirement must be based on a model of human response time.
4. *Minimal Intervention*. The advisory controller should only invoke the human operator when it is necessary, and does so in a minimally-intervening manner (minimizing a given cost function capturing the cost of asking the human to intervene).

Li et al. [13] show how these criteria can be made mathematically precise for the special case when *linear temporal logic* (LTL) is used to express correctness requirements on the system. LTL and its derivatives have proved a very effective specification language for electronic design automation, and it has been useful in some robotics and CPS applications as well. Other formal specification languages can also be employed. Moreover, specialized requirements, e.g., related to security and privacy, may also apply in certain settings. We believe the above four criteria will apply to all h-CPS design problems.

4. VERIFICATION

We see h-CPS as more than just "yet another application area" for formal verification: h-CPS, in general, and semi-autonomous driving, in particular, are also giving rise to interesting new classes of verification problems. We highlight here two main directions:

- *Quantitative Verification:* Traditionally, formal verification tools have Boolean outputs, i.e., either the specification is satisfied or it is not. However, due to the uncertainty inherent in modeling semi-autonomous systems, probabilistic modeling and verification gains importance. Additionally, as stated in the preceding section, some requirements are inherently quantitative, such as the minimal need for human intervention — such requirements are not hard constraints and hence the corresponding verification questions are best captured with a quantitative formulation.
- *Verification with Data-Driven Models:* Models of human agents as well as of the environment are expected to be generated from empirically observed data. Since empirical data is inherently incomplete, the inferred models must represent this incompleteness. For instance, if transition probabilities of a Markov chain model are inferred from empirical data, the estimation error in those probabilities must be represented in the model. Verifying such models requires an extension to existing algorithmic methods such as model checking.

In the rest of this section, we illustrate the above directions with some recent work by the authors and colleagues [19] on probabilistic modeling and verification of human driver behavior. Here the aim is to infer a Markov Decision Process (MDP) model for the whole closed-loop system (including human, controller, plant, and environment) from experimental data obtained from a industrial-scale car simulator.

The driver behavior is dependent on particular modes or scenarios, which are determined by the future external environment, e.g., a turn in the road, and by the driver state, e.g., attentive or distracted. In order to recover these unknown modes, the data is clustered using the k-means algorithm [8], which allows for flexibility in determining the modes in an unsupervised manner. In this modeling framework the modes will be the states of the MDP. These modes are created based on data collected about:

1. *Driver Pose:* This contains the past two seconds of skeleton data, specifically the positions of the wrist, elbow, and shoulder joints.
2. *Environment Estimation:* This contains a feature vector for the future four seconds of the outside environment, including road bounds and curvature, obstacle locations, and the car's deviation from the lane center.

Since the model is inferred from an empirical data set that is incomplete, the model has estimation errors, e.g., in the transition probabilities, that depend on the level of confidence in the data set. Therefore, in this case we infer a generalization of an MDP called a Convex-MDP (CMDP) [17], where the uncertainty in the values of transition probabilities is captured in the form of convex uncertainty regions, a first-class component of the model. Puggelli et al. [17] show how one can extend algorithms for model checking properties expressed in probabilistic computation tree logic (PCTL) to the CMDP model. Sadigh et al. [19] use that model to infer desired properties about human driver behavior, such as a quantitative evaluation of distracted driving.

For instance, Fig. 2 shows verification results for computing the PCTL expression P_{max} [Attention $\mathbf{U}$ $Unsafe$], which is "maximum probability of eventually reaching an unsafe state, if the state of the driver's attention remains constant (one of two states, either attentive or distracted)", for values of confidence level ranging from 60% to 99% for one driver. The probability of reaching an unsafe state is always lower in the case of attentive driving. The probability also decreases as the confidence level is increased, but, more significantly, the difference between the maximumum probabilities for attentive and distracted driver states grows as confidence in the data increases.

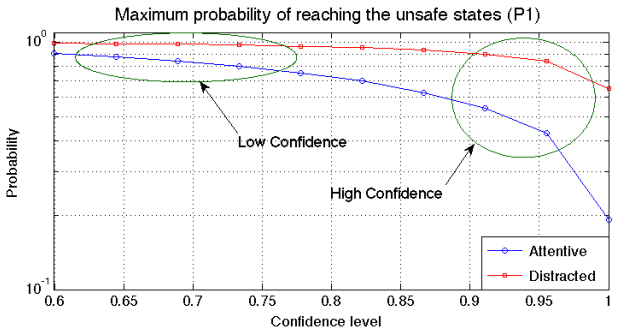

Figure 2: Comparison of distracted and attentive driving for different values of confidence level C_L for Property P1 (reproduced from [19]).

To summarize, we see formal verification of h-CPS as generating a range of new technical questions relating especially to quan-

titative verification and verification with data-driven models.

5. SYNTHESIS AND CONTROL

The design and synthesis of control strategies for h-CPS have important differences with those for fully-autonomous systems. First, as noted in Sec. 2, one must synthesize both the autonomous controller and the advisory controller. Further, as described in Sec. 3, h-CPS have special requirements not present in the fully-autonomous setting. Control algorithms must be modified to address these additional requirements. Moreover, just as in the case of verification, the state of the art in controller synthesis must be extended to handle quantitative requirements and incorporate data-driven techniques. While the specific modifications vary from problem to problem, some general principles have emerged in recent years.

One such principle is based on the common formulation of synthesis as *game solving*. Here the synthesis problem is encoded as a game between the controller and its environment, and the winning strategy of the controller, if one exists, forms the desired control to be synthesized. In zero-sum games, if a winning strategy does not exist for the controller, then one exists for the environment. The latter strategy for the environment is called a *counterstrategy*. Li et al. [12, 13] describe this approach of *counterstrategy-guided synthesis*. They show how one can extract, from the counterstrategy, assumptions about the environment that are sufficient to guarantee correct operation by a fully-autonomous controller. These assumptions, when suitably restricted to be efficiently monitorable at run time, form the basis for an advisory controller. For semi-autonomous driving, such an advisory controller can continually monitor the environment of a vehicle, alerting the human when encountering a situation that the self-driving feature cannot correctly handle.

Another principle involves the *co-design* of human-machine interfaces and control algorithms. The information displayed to human operators must be informed by the state of the controller and its environment. Similarly, the functioning of the controller depends on human input and sensor data capturing information about the state of the human operator and the environment. However, there has been little work in the formal methods and design automation community on these co-design problems.

The identification of these principles is but an initial step. Control for h-CPS with provable guarantees is still an open field with several interesting technical problems yet to be solved.

6. CONCLUSION

In summary, the field of human cyber-physical systems, in general, and semi-autonomous driving, in particular, is a fertile ground for formal methods. 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.

Acknowledgments

This work was funded in part by NSF grant CCF-1116993 and an NDSEG Fellowship. Discussions with collaborators including Ruzena Bajcsy, Bjoern Hartmann, Wenchao Li, Richard Murray, and Claire Tomlin have helped shape some of the ideas in this paper, and we gratefully acknowledge them.

7. REFERENCES

[1] C. Barrett, R. Sebastiani, S. A. Seshia, and C. Tinelli. Satisfiability modulo theories. In A. Biere, H. van Maaren, and T. Walsh, editors, *Handbook of Satisfiability*, volume 4, chapter 8. IOS Press, 2009.

[2] R. E. Bryant. Graph-based algorithms for Boolean function manipulation. *IEEE Transactions on Computers*, C-35(8):677–691, August 1986.

[3] E. M. Clarke and E. A. Emerson. Design and synthesis of synchronization skeletons using branching-time temporal logic. In *Logic of Programs*, pages 52–71, 1981.

[4] E. M. Clarke, O. Grumberg, and D. A. Peled. *Model Checking*. MIT Press, 2000.

[5] E. M. Clarke and J. M. Wing. Formal methods: State of the art and future directions. *ACM Computing Surveys (CSUR)*, 28(4):626–643, 1996.

[6] Federal Aviation Administration (FAA). The interfaces between flight crews and modern flight systems. `http://www.faa.gov/avr/afs/interfac.pdf`, 1995.

[7] M. J. C. Gordon and T. F. Melham. *Introduction to HOL: A Theorem Proving Environment for Higher-Order Logic*. Cambridge University Press, 1993.

[8] J. A. Hartigan et al. Algorithm AS 136: A K-Means Clustering Algorithm. *Journal of the Royal Statistical Society.*, 28(1):pp. 100–108, 1979.

[9] M. Kaufmann, P. Manolios, and J. S. Moore. *Computer-Aided Reasoning: An Approach*. Kluwer Academic Publishers, 2000.

[10] L. T. Kohn and J. M. Corrigan and M. S. Donaldson, editors. To err is human: Building a safer health system. Technical report, A report of the Committee on Quality of Health Care in America, Institute of Medicine, Washington, DC, 2000. National Academy Press.

[11] E. A. Lee and S. A. Seshia. *Introduction to Embedded Systems: A Cyber-Physical Systems Approach*. http://leeseshia.org, first edition edition, 2011.

[12] W. Li, L. Dworkin, and S. A. Seshia. Mining assumptions for synthesis. In *Proceedings of the Ninth ACM/IEEE International Conference on Formal Methods and Models for Codesign (MEMOCODE)*, pages 43–50, July 2011.

[13] W. Li, D. Sadigh, S. Sastry, and S. A. Seshia. Synthesis of human-in-the-loop control systems. In *Proceedings of the 20th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, April 2014.

[14] S. Malik and L. Zhang. Boolean satisfiability: From theoretical hardness to practical success. *Communications of the ACM (CACM)*, 52(8):76–82, 2009.

[15] National Highway Traffic Safety Administration. Preliminary statement of policy concerning automated vehicles, May 2013.

[16] S. Owre, J. M. Rushby, and N. Shankar. PVS: A prototype verification system. In D. Kapur, editor, *11th International Conference on Automated Deduction (CADE)*, volume 607 of *Lecture Notes in Artificial Intelligence*, pages 748–752. Springer-Verlag, June 1992.

[17] A. Puggelli, W. Li, A. Sangiovanni-Vincentelli, and S. A. Seshia. Polynomial-time verification of PCTL properties of MDPs with convex uncertainties. In *Proceedings of the 25th International Conference on Computer-Aided Verification (CAV)*, July 2013.

[18] J.-P. Queille and J. Sifakis. Specification and verification of concurrent systems in CESAR. In *Symposium on Programming*, number 137 in LNCS, pages 337–351, 1982.

[19] D. Sadigh, K. Driggs-Campbell, A. Puggelli, W. Li, V. Shia,

R. Bajcsy, A. L. Sangiovanni-Vincentelli, S. S. Sastry, and S. A. Seshia. Data-driven probabilistic modeling and verification of human driver behavior. In *Formal Verification and Modeling in Human-Machine Systems, AAAI Spring Symposium*, March 2014.

[20] J. M. Wing. A specifier's introduction to formal methods. *IEEE Computer*, 23(9):8–24, September 1990.