# Logic & Proofs for Cyber-Physical Systems*

André Platzer

Computer Science Department, Carnegie Mellon University, Pittsburgh, USA
aplatzer@cs.cmu.edu

**Abstract.** *Cyber-physical systems* (CPS) combine cyber aspects such as communication and computer control with physical aspects such as movement in space, which arise frequently in many safety-critical application domains, including aviation, automotive, railway, and robotics. But how can we ensure that these systems are guaranteed to meet their design goals, e.g., that an aircraft will not crash into another one?
This paper highlights some of the most fascinating aspects of cyber-physical systems and their dynamical systems models, such as hybrid systems that combine discrete transitions and continuous evolution along differential equations. Because of the impact that they can have on the real world, CPSs deserve proof as safety evidence.
*Multi-dynamical systems* understand complex systems as a combination of multiple elementary dynamical aspects, which makes them natural mathematical models for CPS, since they tame their complexity by compositionality. The family of *differential dynamic logics* achieves this compositionality by providing compositional logics, programming languages, and reasoning principles for CPS. Differential dynamic logics, as implemented in the theorem prover KeYmaera X, have been instrumental in verifying many applications, including the Airborne Collision Avoidance System ACAS X, the European Train Control System ETCS, automotive systems, mobile robot navigation, and a surgical robot system for skull-base surgery. This combination of strong theoretical foundations with practical theorem proving challenges and relevant applications makes *Logic for CPS* an ideal area for compelling and rewarding research.

## Logical Foundations of Cyber-Physical Systems

Can we trust a computer to control physical processes? That depends on how it has been programmed and what will happen if it malfunctions. When a lot is at stake, computers need to be *guaranteed* to interact correctly with the physical world. So, we need ways of analyzing, designing, and guaranteeing the behavior of such systems. Providing these ways is an *intellectual grand challenge* with substantial scientific, economical, societal, and educational impact. Its solution is the key to enabling computer assistance that we can bet our lives on.

* This paper focuses on illustrating important principles of cyber-physical systems here. Technical surveys can be found in the literature, e.g., [2, 7, 8, 12, 20, 32, 41, 42]. This material is based upon work supported by the National Science Foundation under NSF CAREER Award CNS-1054246.


N. Olivetti and A. Tiwari (Eds.): IJCAR 2016, LNAI 9706, pp. 15–21, 2016.
DOI: 10.1007/978-3-319-40229-1_3

**Cyber-Physical Systems.** Computer control has been suggested to remedy inefficiencies, reliability issues, or defects for virtually all physical systems. But computer control only helps our society if we can ensure that it works correctly. As has been argued on numerous occasions [1–8, 11, 12, 17, 18, 20, 21, 23–27, 38, 41–44], we must, thus, *verify* the correctness of these systems, as testing may miss bugs. This problem is confounded, because the behavior of the system under one circumstance can radically differ from the behavior under another, especially when complex computer decisions for different objectives interact. It is crucial to prove the absence of bugs so that we are confident to bet our lives on the system functioning correctly, since that is what we do every time we get into an airplane or car.

Systems like these are called *cyber-physical systems* (*CPS*). They combine cyber capabilities (communication, computation and control) with physical capabilities (sensing and actuation) to solve problems *that neither part could solve alone*. While CPS are widely appreciated for their broad range of application domains (e.g., automotive, aerospace, medical, transportation, civil engineering, materials, chemistry, energy), the goal of the *Logical Foundations of CPS* is to identify the *common foundational core* that constitutes the true essence of CPS and their proof principles to serve as the simultaneous mathematical basis for all those applications. The foundations of digital computer science have revolutionized how systems are designed and our whole society works. We need even stronger foundations when software reaches out into our physical world.

**Multi-dynamical Systems.** The first crucial insight for CPS foundations is the multi-dynamical systems principle [32] of understanding complex systems as a combination of multiple elementary dynamical aspects. Mathematically, CPS are *multi-dynamical systems* [32], i.e. systems characterized by multiple facets of dynamical systems, schematically summarized in Fig. 1. CPS involve computer control decisions and are, thus, *discrete*. CPS are *continuous*, because they evolve along differential equations of motion or other physical processes. CPS are *uncertain*, because their behavior is subject to choices coming from environmental variability or intentional uncertainties that simplify their model. This uncertainty can manifest in different ways. Uncertainties make CPS *stochastic* when good information about the distribution of choices is available. Uncertainties make CPS *nondeterministic* when no commitment about the resolution of choices is made. Uncertainties make CPS *adversarial* when they involve multiple agents with potentially conflicting goals or even active competition in a game. Verifying that CPS work correctly requires dealing with all of these dynamical features—and sometimes even more—at the same time.

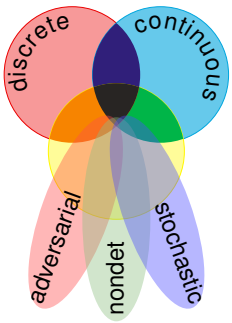


**Fig. 1.** Dynamical aspects of CPS

**CPS Proofs.** Multi-dynamical systems study complex CPS as a combination of multiple elementary dynamical aspects. This approach helps to tame the com-

plexity of CPS by understanding that their complexity just comes from combining lots of simple dynamical effects with one another. The overall system is quite complex, but each of its pieces is better-behaved, since it only has one dynamics. What miracle translates this *descriptive simplification* of a CPS in terms of a combination of multiple dynamical aspects into an *analytic simplification* in terms of multiple dynamical systems that can be considered side-by-side?

*The key to this mystery is to integrate the CPS dynamics all within a single, compositional logic* [32]. Since compositionality is an intrinsic feature starting from the very semantics of logic [9, 10, 13, 14, 37, 39, 40], logics naturally reason compositionally, too. With suitable generalizations of logics to embrace multi-dynamical systems [27–31, 34, 35], this compositionality generalizes to CPS. Verification works by constructing a proof in such a logic. The whole proof verifies a complex CPS. Yet, each proof step only reasons separately about one dynamical aspect at a time using, e.g., local dynamics of differential equations, the theory of real-closed fields, symbolic logic, differential form computations [35], fixpoint theory [34], and so on, each captured in a separate, modular axiom or proof rule.

**Theory.** This logical view on CPS has already made it possible to develop rich theories of *hybrid systems* that combine discrete change and continuous differential equations [27, 31, 35], theories of *distributed hybrid systems* that combine distributed systems with hybrid systems [30], theories of *hybrid games* that combine discrete, continuous, and adversarial dynamics [34], all of which are sound and relatively complete, but was also used for *stochastic hybrid systems* [29]. The approach was instrumental in formulating and proving the first [27] and second [31] *completeness theorem* for hybrid systems, which characterize and align the discrete and continuous challenges of hybrid systems, and reveal their fundamental symmetry. The theory of hybrid systems forms a *proof-theoretical bridge* aligning the theory of continuous systems with the theory of discrete systems. Proof theory was essential in the study of *provability of properties of differential equations* and *differential cut elimination* [33], which turn out to generalize ideas from Lie's results on Lie groups [19] but also relate to Gentzen's cut elimination theorem in classical logic [10]. Logic was equally crucial for the development of *differential ghosts* that create extra dimensions [33] as proof-theoretical analogues of dark matter, whose existence was speculated to balance out energy invariants in astrophysics [16].

As a logical rendition of Lie's ideas, *differential invariants* [28, 33, 35] enable induction principles for differential equations characterizing the rate of change of truth of a formula in the direction of the dynamics; see Fig. 2. Intuitively, $F$ always remains true after following the differential equation $x' = f(x)$

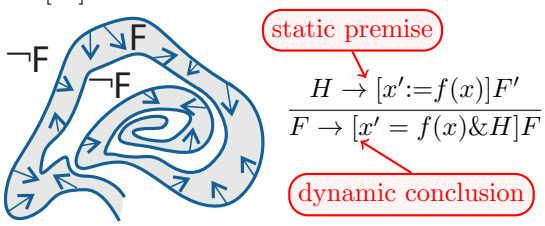


**Fig. 2. (left)** Differential invariant $F$ **(right)** Proof rule for invariance of $F$ along differential equation $x' = f(x)$ in evolution domain $H$

within the domain $H$ (conclusion), if $F$ started out true initially (conclusion's assumption), and if, within $H$, the differential $F'$ of $F$ (which characterizes the infinitesimal change of $F$ as a function of $x'$) holds after assigning the right-hand side $f(x)$ of the differential equation to its left-hand side $x'$ (premise). Differential invariants lift the high descriptive power of differential equations to a high analytic power, so that their properties can be proved even if the equations cannot be solved. Solutions ruin the descriptive power even *if* the differential equations can be solved, so that differential invariants are advantageous regardless.

**Applications.** Logical Foundations of CPS play an increasingly important role in practical applications by way of their implementations in the theorem prover KeYmaera and its clean-slate successor[1] KeYmaera X. This includes finding and fixing [36] flaws in an air traffic conflict resolution maneuver, verifying and identifying issues in the *Next-generation Airborne Collision Avoidance System* ACAS X [15], verifying the *European train control system* ETCS, car control systems, mobile ground robot navigation, and finding and fixing bugs in a skull-base surgical robot system. Logic also identified a way of correctly relating proof in a model to truth in reality [22], which is an inevitable challenge for CPS.

Finally, multi-dynamical systems impact education in the *Foundations of Cyber-Physical Systems* course that is breaking with the myth that cyber-physical systems are too challenging to be taught at the undergraduate level. The compositionality principles of logic and multi-dynamical systems considerably tame the educational complexity of CPS by making it possible to focus on one aspect at a time without losing the ability to combine the understanding attained for each aspect. The rich variety of systems that the students verified for their final course projects[2] indicates that this approach effectively conveys the principles for a successful separation of concerns for CPS.

**Summary.** Logical foundations make a big difference for cyber-physical systems, certainly in understanding the basic principles of CPS, but also in real applications like the Next-generation Airborne Collision Avoidance System. Lessons from centuries of logic and foundations research can have a huge impact on advancing CPS. Yet, conversely, the questions that CPS pose can have an equally significant impact on advancing logic. Cyber-physical systems serve as a catalytic integrator for other sciences, because they benefit from combining numerous exciting areas of logic, mathematics, computer science, and control theory that previously seemed unrelated. The mix of enabling strong analytic foundations with the need for practical advances of rigorous reasoning and the significance of its applications, as well as its fruitful interactions with many other sciences, make cyber-physical systems an ideal field for compelling and rewarding research that has only just begun. Numerous wonders remain yet to be discovered.

[1] `http://www.keymaeraX.org/`

[2] The students' self-defined 3-week course projects and their presentations to a panel of experts from industry in the CPS V&V Grand Prix are available from the course web pages `http://lfcps.org/course/fcps.html`

## References


1. Alur, R.: Formal verification of hybrid systems. In: Chakraborty, S., Jerraya, A., Baruah, S.K., Fischmeister, S. (eds.) EMSOFT. pp. 273–278. ACM (2011)
2. Alur, R.: Principles of Cyber-Physical Systems. MIT Press (2015)
3. Alur, R., Courcoubetis, C., Halbwachs, N., Henzinger, T.A., Ho, P.H., Nicollin, X., Olivero, A., Sifakis, J., Yovine, S.: The algorithmic analysis of hybrid systems. Theor. Comput. Sci. 138(1), 3–34 (1995)
4. Alur, R., Henzinger, T., Lafferriere, G., Pappas, G.J.: Discrete abstractions of hybrid systems. Proc. IEEE 88(7), 971–984 (2000)
5. Branicky, M.S.: General hybrid dynamical systems: Modeling, analysis, and control. In: Alur, R., Henzinger, T.A., Sontag, E.D. (eds.) Hybrid Systems. LNCS, vol. 1066, pp. 186–200. Springer (1995)
6. Clarke, E.M., Emerson, E.A., Sifakis, J.: Model checking: algorithmic verification and debugging. Commun. ACM 52(11), 74–84 (2009)
7. Davoren, J.M., Nerode, A.: Logics for hybrid systems. IEEE 88(7), 985–1010 (2000)
8. Doyen, L., Frehse, G., Pappas, G.J., Platzer, A.: Verification of hybrid systems. In: Clarke, E.M., Henzinger, T.A., Veith, H. (eds.) Handbook of Model Checking, chap. 28. Springer (2017)
9. Frege, G.: Begriffsschrift, eine der arithmetischen nachgebildete Formelsprache des reinen Denkens. Verlag von Louis Nebert (1879)
10. Gentzen, G.: Untersuchungen über das logische Schließen. I. Math. Zeit. 39(2), 176–210 (1935)
11. Henzinger, T.A., Sifakis, J.: The discipline of embedded systems design. Computer 40(10), 32–40 (Oct 2007)
12. Henzinger, T.A.: The theory of hybrid automata. In: LICS. pp. 278–292. IEEE Computer Society, Los Alamitos (1996)
13. Hilbert, D.: Die Grundlagen der Mathematik. Abhandlungen aus dem Seminar der Hamburgischen Universität 6(1), 65–85 (1928)
14. Hoare, C.A.R.: An axiomatic basis for computer programming. Commun. ACM 12(10), 576–580 (1969)
15. Jeannin, J., Ghorbal, K., Kouskoulas, Y., Gardner, R., Schmidt, A., Zawadzki, E., Platzer, A.: A formally verified hybrid system for the next-generation airborne collision avoidance system. In: Baier, C., Tinelli, C. (eds.) TACAS. LNCS, vol. 9035, pp. 21–36. Springer (2015)
16. Kapteyn, J.C.: First attempt at a theory of the arrangement and motion of the sidereal system. Astrophysical Journal 55, 302 (May 1922)
17. Larsen, K.G.: Verification and performance analysis for embedded systems. In: Chin, W., Qin, S. (eds.) TASE 2009, Third IEEE International Symposium on Theoretical Aspects of Software Engineering, 29-31 July 2009, Tianjin, China. pp. 3–4. IEEE Computer Society (2009)
18. Lee, E.A., Seshia, S.A.: Introduction to Embedded Systems — A Cyber-Physical Systems Approach. Lulu.com (2013)
19. Lie, S.: Vorlesungen über continuierliche Gruppen mit geometrischen und anderen Anwendungen. Teubner, Leipzig (1893)
20. Lunze, J., Lamnabhi-Lagarrigue, F. (eds.): Handbook of Hybrid Systems Control: Theory, Tools, Applications. Cambridge Univ. Press (2009)
21. Maler, O.: Control from computer science. Annual Reviews in Control 26(2), 175–187 (2002)

22. Mitsch, S., Platzer, A.: ModelPlex: Verified runtime validation of verified cyber-physical system models. Form. Methods Syst. Des. (2016), special issue of selected papers from RV'14
23. Nerode, A.: Logic and control. In: Cooper, S.B., Löwe, B., Sorbi, A. (eds.) CiE. LNCS, vol. 4497, pp. 585–597. Springer (2007)
24. Nerode, A., Kohn, W.: Models for hybrid systems: Automata, topologies, controllability, observability. In: Grossman, R.L., Nerode, A., Ravn, A.P., Rischel, H. (eds.) Hybrid Systems. LNCS, vol. 736, pp. 317–356. Springer (1992)
25. NITRD CPS Senior Steering Group: CPS vision statement. NITRD (2012)
26. Pappas, G.J.: Wireless control networks: modeling, synthesis, robustness, security. In: Caccamo, M., Frazzoli, E., Grosu, R. (eds.) Proceedings of the 14th ACM International Conference on Hybrid Systems: Computation and Control, HSCC 2011, Chicago, IL, USA, April 12-14, 2011. pp. 1–2. ACM (2011)
27. Platzer, A.: Differential dynamic logic for hybrid systems. J. Autom. Reas. 41(2), 143–189 (2008)
28. Platzer, A.: Differential-algebraic dynamic logic for differential-algebraic programs. J. Log. Comput. 20(1), 309–352 (2010)
29. Platzer, A.: Stochastic differential dynamic logic for stochastic hybrid programs. In: Bjørner, N., Sofronie-Stokkermans, V. (eds.) CADE. LNCS, vol. 6803, pp. 431–445. Springer (2011)
30. Platzer, A.: A complete axiomatization of quantified differential dynamic logic for distributed hybrid systems. Log. Meth. Comput. Sci. 8(4), 1–44 (2012), special issue for selected papers from CSL'10
31. Platzer, A.: The complete proof theory of hybrid systems. In: LICS. pp. 541–550. IEEE (2012)
32. Platzer, A.: Logics of dynamical systems. In: LICS. pp. 13–24. IEEE (2012)
33. Platzer, A.: The structure of differential invariants and differential cut elimination. Log. Meth. Comput. Sci. 8(4), 1–38 (2012)
34. Platzer, A.: Differential game logic. ACM Trans. Comput. Log. 17(1), 1:1–1:51 (2015)
35. Platzer, A.: A uniform substitution calculus for differential dynamic logic. In: Felty, A., Middeldorp, A. (eds.) CADE. LNCS, vol. 9195, pp. 467–481. Springer (2015)
36. Platzer, A., Clarke, E.M.: Formal verification of curved flight collision avoidance maneuvers: A case study. In: Cavalcanti, A., Dams, D. (eds.) FM. LNCS, vol. 5850, pp. 547–562. Springer (2009)
37. Pratt, V.R.: Semantical considerations on Floyd-Hoare logic. In: FOCS. pp. 109–121. IEEE (1976)
38. President's Council of Advisors on Science and Technology: Leadership under challenge: Information technology R&D in a competitive world. An Assessment of the Federal Networking and Information Technology R&D Program (Aug 2007)
39. Scott, D., Strachey, C.: Toward a mathematical semantics for computer languages? Tech. Rep. PRG-6, Oxford Programming Research Group (1971)
40. Smullyan, R.M.: First-Order Logic. Dover (1968)
41. Tabuada, P.: Verification and Control of Hybrid Systems: A Symbolic Approach. Springer (2009)
42. Tiwari, A.: Abstractions for hybrid systems. Form. Methods Syst. Des. 32(1), 57–83 (2008)
43. Tiwari, A.: Logic in software, dynamical and biological systems. In: LICS. pp. 9–10. IEEE Computer Society (2011)
44. Wing, J.M.: Five deep questions in computing. Commun. ACM 51(1), 58–60 (2008)