Efficient and Certifiable Cyber-Physical Systems — Extended Abstract —

Samarjit Chakraborty[0000−0002−0503−6235]

Department of Computer Science
The University of North Carolina at Chapel Hill, USA
samarjit@cs.unc.edu
https://cs.unc.edu/person/samarjit-chakraborty/

Abstract. Examples of cyber-physical systems (CPS) are many — autonomous and semi-autonomous cars, robots, industrial manufacturing systems, to name a few. Today, many critical functions in these systems are controlled by embedded computers and software, where the algorithmic core of the software is made up of multiple feedback controllers. The design of such control algorithms follow well-developed principles of abstraction, and rely on many simplifying assumptions on the implementation platform. However, as both—the controllers and the embedded computers implementing them—become more complex, heterogeneous, and distributed, traditional abstractions are increasingly becoming infeasible. In particular, it is becoming challenging to have efficient implementations, while ensuring certifiability. We will discuss the reasons behind this, and ongoing research in our group to address these challenges.

Keywords: Cyber-Physical Systems · Real-Time Systems · Controller Synthesis

1 Introduction

The design of cyber-physical systems (CPS) software requires an explicit modeling of the dynamics of physical systems being controlled by computers. There is a large volume of existing literature on how to design such controllers and implement them as software [65]. However, with the advent of autonomous systems—*viz.*, autonomous cars, robots, drones, *etc.*—control strategies have become more complex, perception processing increasingly relies on machine learning (ML), and the hardware platforms used to implement such controllers have also become more heterogeneous and distributed in their architecture. All of these developments have made it difficult to reconcile the models of CPS with their software implementations running on real-embedded systems. Because of this, certifying the correctness of autonomous CPS [8, 13] is posing a challenge in the adoption of technologies like autonomous vehicles and robots.

Towards addressing this challenge, work has been done on formal verification of control strategies [2, 77, 31, 9, 42], and on the verification and testing of ML algorithms used for sensor and perception processing [37, 84]. A considerable

amount of research has also been done on closing the "*model-implementation gap*" [81, 80, 10, 67] arising from the assumptions made when designing control strategies, versus the realities of modern distributed implementation platforms [11], as outlined above. As implementation platforms become more complex, timing assumptions made during controller design [41, 40] are increasingly difficult to ensure and verify [24]. In fact basic timing analysis tasks such as estimating safe *and* tight *worst case execution time* (WCET) estimates of software tasks is becoming very challenging [86, 78]. For WCET estimates to be safe, they are increasingly overestimated. Meeting all task deadlines with such overestimated WCET values leads to pessimistic or infeasible implementations. Further, in domains like automotive, in-vehicle architectures are rapidly moving away from "one function per ECU" or federated, to multiple functions sharing resources, *viz.*, "integrated" architectures [55]. The clear trend is that future architectures will be less "static" than before, as indicated by developments like AUTOSAR Adaptive [51] and service-oriented paradigms [22, 4]. As a result, the timing non-determinism experienced by control software running on such implementation platforms will only continue to increase. But in traditional design processes control engineers assume certain timing properties or deadlines that the control tasks need to satisfy for them to behave as desired, and the embedded systems engineers schedule them to meet those deadlines [50]. This ensures a clean separation of concerns, allowing the two groups of engineers to work independently.

Such a design flow where system components are designed assuming certain deterministic reliability guarantees from the other components is increasingly breaking down. This is not only the case with timing uncertainties, but is a broader design challenge. For example, when using ML components for state estimation — as it occurs when processing camera, radar or lidar data in autonomous cars — the results cannot be assumed to be perfectly correct. Similarly, 100% security [85] cannot be assumed at moderate cost during data transmission from the plant to the controller or the controller to the actuator. In all of these cases, ensuring 100% reliability comes at the expense of prohibitive cost or excessive design pessimism. Hence, an important question is: *Is it possible to certify system safety of autonomous CPS, while ensuring efficiency?* We claim that it is indeed possible to do so, and outline some of our recent work [35, 26, 88] on certifying system safety under implementation platform timing uncertainties.

In the next section we outline our work on certifying a controller implementation for safety, and briefly describe how such a scheme can also be used to synthesize task schedulers that aim not towards meeting deadline constraints, but towards satisfying system-level safety properties. In Section 3 we describe some related work, mostly focusing on prior work done by us.

2 Safety Certification and Synthesis

By focusing on timing uncertainties, in this section we briefly outline how closed-loop controllers can be certified for safety even when control tasks occasionally

miss their deadlines. We then show how this could be exploited to synthesize schedulers for scheduling multiple control tasks. Such a scheduler can miss task deadlines, but will ensure that the specified system-level safety property of each controller is satisfied. With a less stringent real-time constraint (*viz.,* allowing tasks to miss deadlines), it is possible to obtain more efficient implementations of control software or allow more pessimistic WCET estimates. Such a scheduler will be able to schedule more control tasks on an implementation platform, compared to a conventional task scheduler whose goal will be to meet real-time—instead of system-level safety—constraints.

2.1 Safety Certification with Timing Uncertainties

Instead of assuming that the platform implementing control software has an ideal timing behavior, *viz.,* no deadline misses, can we certify system safety when the control task misses deadlines [59]? The underlying idea is to first define a notion of *safety*. Towards this, we construct a *safety pipe* around the system trajectory in the state space that captures the *ideal* dynamics of the closed-loop system. Such an ideal dynamics or trajectory is when the system experiences an ideal timing behavior, *e.g.*, no deadline misses. In the presence of deadline misses, the system's dynamics will deviate from the ideal trajectory, and the extent of the deviation will generally correspond to the number of deadline misses the control task experiences. But as long as the deviation is not significant, and in particular is contained within the safety pipe, the system can be certified to be safe. The diameter or shape of the safety pipe can be specified based on the desired notion of safety [35].

Towards realizing this idea, we first characterize deadline hit/miss patterns of a control task as a regular language [48]. Using approximate reachability analysis we then check whether all patterns admitted by a given language meet the above safety requirement. In other words, if the deviation of the closed-loop dynamics of the system from the dynamics when the control task always meets its deadline, is less than a specified bound.

2.2 Synthesizing schedulers for system safety

The procedure for checking system-level safety in the presence of timing uncertainties may also be extended to *synthesize* schedules for multiple control tasks [26, 88]. Here, given safety specifications for each control task, the question is: *Can these tasks be scheduled, while allowing deadline misses, such that the safety specification for each controller is satisfied?*

The synthesis involves (i) again characterizing deadline hit/miss patterns as regular languages, (ii) using approximate reachability analysis techniques to check whether all hit/miss patterns admitted by a given regular language meet specified safety properties using techniques, and finally (iii) using automata theoretic techniques to check whether these schedules satisfy the scheduling/resource constraint. An example of such a constraint could be that time is partitioned

into equal-length slots and no more than two tasks can be scheduled in each time slot. This means that tasks that are not scheduled in a slot miss their deadlines.

3 Related Work on Efficient & Certifiable CPS Design

Since the complexity of CPS hardware architectures has been rapidly growing, are they susceptible to a variety of manufacturing variabilities, transient faults and aging issues, that might impact the software execution and the results they produce [25, 17, 16, 23]. The increasing complexity of software in autonomous CPS and the associated scheduling [12, 90, 74] and management [64, 71, 73] of such software also contributes to growing safety concerns. This has led to a variety of work on testing [79, 43, 56] and compositional formal verification of real-time control software and CPS architectures [9, 83, 34, 62, 76].

Different control-theoretic methods, reachability analysis, and cross-layer design techniques have been proposed for safety verification of CPS [35, 68, 18, 44, 27]. Timing uncertainties increase in CPS architectures as they become more distributed and use heterogeneous architectures and communication protocols. As a result, control signals are subject to varying delays that can compromise the safety of the closed-loop system, even if the control strategy is functionally correct. Several papers have addressed this problem [29, 48, 6, 82]. In addition to schedule and implementation architecture synthesis, the orthogonal problem of synthesizing delay-tolerant controllers has been studied in [28, 52, 15], including the synthesis of safe controllers that exploit features of the implementation platform and its underlying operating system [21, 19, 20]. Further, the problem of *co-synthesizing* controllers and their underlying task schedules have been explored in [75, 66, 7, 49]. All of these problems also have connections to providing timing isolation to critical control software [50, 69], scheduling to meet timing constraints [61, 39, 89, 70, 46, 38] and the scheduling of mixed-criticality tasks [14, 30]. In the case of vision-based control systems, the accuracy of the (often ML-based) vision processing system [5, 3, 57] plays an important role in the safety certification of the closed-loop system. Recent work has focused on emerging CPS topics like electric vehicles [47, 45, 87, 1], autonomous vehicles [72, 36], CPS security [53, 85] and safety and certification issues arising in them. With the growing adoption of electric vehicles and drones, there has been increasing focus on the safety and reliability of battery systems [17, 58, 54]. Work on this topic also relies on battery aging models as studied in the context of mobile devices [63], but is substantially different from prior work on energy management for such devices [32, 33, 60], and is currently much less developed.

It is worth noting that conventional controller design approaches focus on ensuring stability or on optimizing control performance metrics like peak overshoot. During such a design process, computer science or algorithmic efficiency metrics pertaining to computation, communication, or memory consumption are not accounted for. These only come into picture at the *implementation* stage, by when the control strategy has already been frozen. Some of our work has also attempted to incorporate these metrics at the controller design stage [19,

15, 21], which result in more efficient controller implementations, while ensuring stability, control performance, and also certifiability.

4 Concluding Remarks

Current workflows for CPS design attempt to design each component in isolation, and aim to optimize a "local" metric, like meet all task deadlines. While this enables separation of concerns and design modularity, in the process of ensuring system safety and certification, it also results in undue pessimism. We argue that by relaxing the notion of safety and allowing each of the system components to admit some error or failure, it would be possible to ensure *both* — efficiency and certifiability. We have recently shown this in the case of timing and its impact on system-level safety. Our future research agenda is to determine how this strategy would also extend to other system components, such as those responsible for security and ML.

References

1. Aalund, R., Diao, W., Kong, L., Pecht, M.G.: Understanding the non-collision related battery safety risks in electric vehicles a case study in electric vehicle recalls and the LG chem battery. IEEE Access **9**, 89527–89532 (2021)
2. Alegre, F., Feron, E., Pande, S.: Using ellipsoidal domains to analyze control systems software. CoRR **abs/0909.1977** (2009), `http://arxiv.org/abs/0909.1977`
3. Amert, T., Balszun, M., Geier, M., Smith, F.D., Anderson, J.H., Chakraborty, S.: Timing-predictable vision processing for autonomous systems. In: Design, Automation & Test in Europe Conference (DATE) (2021)
4. Arestova, A., Martin, M., Hielscher, K.S.J., German, R.: A service-oriented real-time communication scheme for AUTOSAR adaptive using OPC UA and time-sensitive networking. Sensors **21**(7) (2021)
5. Balszun, M., Geier, M., Chakraborty, S.: Predictable vision for autonomous systems. In: IEEE 23rd International Symposium on Real-Time Distributed Computing (ISORC) (2020)
6. Balszun, M., et al.: Effectively utilizing elastic resources in networked control systems. In: 23rd IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA) (2017)
7. Bhatia, L., Tomic, I., Fu, A., Breza, M., McCann, J.A.: Control communication co-design for wide area cyber-physical systems. ACM Trans. Cyber Phys. Syst. **5**(2), 18:1–18:27 (2021)
8. Bordoloi, U.D., et al.: Autonomy-driven emerging directions in software-defined vehicles. In: Design, Automation & Test in Europe Conference (DATE) (2023)
9. Broy, M., et al.: Cross-layer analysis, testing and verification of automotive control software. In: 11th International Conference on Embedded Software (EMSOFT) (2011)
10. Caspi, P., Scaife, N., Sofronis, C., Tripakis, S.: Semantics-preserving multitask implementation of synchronous programs. ACM Trans. Embedded Comput. Syst. **7**(2), 15:1–15:40 (2008)
11. Chakraborty, S., et al.: Cross-layer interactions in CPS for performance and certification. In: DATE (2019)

12. Chakraborty, S., Erlebach, T., Thiele, L.: On the complexity of scheduling conditional real-time code. In: 7th International Workshop on Algorithms and Data Structures (WADS) (2001)
13. Chakraborty, S., Faruque, M.A.A., Chang, W., Goswami, D., Wolf, M., Zhu, Q.: Automotive cyber-physical systems: A tutorial introduction. IEEE Design & Test **33**(4), 92–108 (2016)
14. Chakraborty, S., et al.: Embedded systems and software challenges in electric vehicles. In: Design, Automation & Test in Europe Conference & Exhibition (DATE) (2012)
15. Chang, W., Chakraborty, S.: Resource-aware automotive control systems design: A cyber-physical systems approach. Foundations and Trends in Electronic Design Automation **10**(4), 249–369 (2016)
16. Chang, W., et al.: Battery- and aging-aware embedded control systems for electric vehicles. In: 35th IEEE Real-Time Systems Symposium (RTSS) (2014)
17. Chang, W., et al.: Reliable CPS design for mitigating semiconductor and battery aging in electric vehicles. In: 3rd IEEE International Conference on Cyber-Physical Systems, Networks, and Applications (CPSNA) (2015)
18. Chang, W., et al.: Model-based design of resource-efficient automotive control software. In: 35th International Conference on Computer-Aided Design (ICCAD) (2016)
19. Chang, W., et al.: Memory-aware embedded control systems design. IEEE Trans. on CAD of Integrated Circuits and Systems **36**(4), 586–599 (2017)
20. Chang, W., et al.: Cache-aware task scheduling for maximizing control performance. In: Design, Automation & Test in Europe (DATE) (2018)
21. Chang, W., et al.: OS-aware automotive controller design using non-uniform sampling. ACM Transactions on Cyber-Physical Systems TCPS **2**(4), 26:1–26:22 (2018)
22. Fraccaroli, E., Joshi, P., Xu, S., Shazzad, K., Jochim, M., Chakraborty, S.: Timing predictability for SOME/IP-based service-oriented automotive in-vehicle networks. In: Design, Automation & Test in Europe Conference (DATE) (2023)
23. Gandoman, F.H., et al.: Status and future perspectives of reliability assessment for electric vehicles. Reliab. Eng. Syst. Saf. **183**, 1–16 (2019)
24. Geier, M., Burghart, T., Hackl, M., Chakraborty, S.: In situ latency monitoring for heterogeneous real-time systems. In: 32nd International Conference on VLSI Design (VLSID) (2019)
25. Georgakos, G., et al.: Reliability challenges for electric vehicles: from devices to architecture and systems software. In: 50th Annual Design Automation Conference (DAC) (2013)
26. Ghosh, B., et al.: Statistical hypothesis testing of controller implementations under timing uncertainties. In: 28th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA) (2022)
27. Goswami, D., Schneider, R., Chakraborty, S.: Co-design of cyber-physical systems via controllers with flexible delay constraints. In: 16th Asia South Pacific Design Automation Conference (ASP-DAC) (2011)
28. Goswami, D., Schneider, R., Chakraborty, S.: Re-engineering cyber-physical control applications for hybrid communication protocols. In: Design, Automation and Test in Europe (DATE) (2011)
29. Goswami, D., Schneider, R., Chakraborty, S.: Relaxing signal delay constraints in distributed embedded controllers. IEEE Trans. Contr. Sys. Techn. **22**(6), 2337–2345 (2014)

30. Goswami, D., et al.: Time-triggered implementations of mixed-criticality automotive software. In: Design, Automation & Test in Europe Conference & Exhibition (DATE) (2012)
31. Goswami, D., et al.: Model-based development and verification of control software for electric vehicles. In: The 50th Annual Design Automation Conference (DAC) (2013)
32. Gu, Y., Chakraborty, S.: A hybrid DVS scheme for interactive 3d games. In: 14th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS) (2008)
33. Gu, Y., Chakraborty, S.: Power management of interactive 3d games using frame structures. In: 21st International Conference on VLSI Design (2008)
34. Guo, L., Zhu, Q., Nuzzo, P., Passerone, R., Sangiovanni-Vincentelli, A.L., Lee, E.A.: Metronomy: A function-architecture co-simulation framework for timing verification of cyber-physical systems. In: International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS) (2014)
35. Hobbs, C., Ghosh, B., Xu, S., Duggirala, P.S., Chakraborty, S.: Safety analysis of embedded controllers under implementation platform timing uncertainties. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **41**(11), 4016–4027 (2022)
36. Hobbs, C., et al.: Perception computing-aware controller synthesis for autonomous systems. In: Design, Automation & Test in Europe Conference & Exhibition (DATE) (2021)
37. Hsieh, C., Li, Y., Sun, D., Joshi, K., Misailovic, S., Mitra, S.: Verifying controllers with vision-based perception using safe approximate abstractions. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **41**(11), 4205–4216 (2022)
38. Jacob, R., et al.: TTW: A time-triggered wireless design for CPS. In: Design, Automation & Test in Europe Conference (DATE) (2018)
39. Ju, L., Chakraborty, S., Roychoudhury, A.: Accounting for cache-related preemption delay in dynamic priority schedulability analysis. In: Design, Automation and Test in Europe (DATE) (2007)
40. Ju, L., Huynh, B.K., Roychoudhury, A., Chakraborty, S.: Timing analysis of Esterel programs on general-purpose multiprocessors. In: 47th Design Automation Conference (DAC) (2010)
41. Ju, L., et al.: Context-sensitive timing analysis of Esterel programs. In: 46th Design Automation Conference (DAC) (2009)
42. Kauer, M., et al.: Fault-tolerant control synthesis and verification of distributed embedded systems. In: Design, Automation & Test in Europe Conference (DATE) (2014)
43. Kramer, S., Ziegenbein, D., Hamann, A.: Real world automotive benchmark for free. In: International Workshop on Analysis Tools and Methodologies for Embedded and Real-Time Systems (2015)
44. Kumar, P., et al.: A hybrid approach to cyber-physical systems verification. In: Design Automation Conference (DAC) (2012)
45. Lukasiewycz, M., et al.: Cyber-physical systems design for electric vehicles. In: 15th Euromicro Conference on Digital System Design (DSD) (2012)
46. Lukasiewycz, M., et al.: Modular scheduling of distributed heterogeneous time-triggered automotive systems. In: 17th Asia and South Pacific Design Automation Conference (ASP-DAC) (2012)
47. Lukasiewycz, M., et al.: System architecture and software design for electric vehicles. In: 50th Annual Design Automation Conference (DAC) (2013)

48. Maggio, M., Hamann, A., Mayer-John, E., Ziegenbein, D.: Control-system stability under consecutive deadline misses constraints. In: 32nd Euromicro Conference on Real-Time Systems (ECRTS) (2020)
49. Mahfouzi, R., Aminifar, A., Samii, S., Rezine, A., Eles, P., Peng, Z.: Breaking silos to guarantee control stability with communication over ethernet TSN. IEEE Des. Test **38**(5), 48–56 (2021)
50. Masrur, A., et al.: VM-based real-time services for automotive control applications. In: 16th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA) (2010)
51. Menard, C., Goens, A., Lohstroh, M., Castrillón, J.: Achieving determinism in adaptive AUTOSAR. In: Design, Automation and Test in Europe Conference (DATE) (2020)
52. Mohamed, S., Goswami, D., Nathan, V., Rajappa, R., Basten, T.: A scenario- and platform-aware design flow for image-based control systems. Microprocess. Microsystems **75**, 103037 (2020)
53. Mundhenk, P., et al.: Security analysis of automotive architectures using probabilistic model checking. In: 52nd Annual Design Automation Conference (DAC) (2015)
54. Narayanaswamy, S., Kauer, M., Steinhorst, S., Lukasiewycz, M., Chakraborty, S.: Modular active charge balancing for scalable battery packs. IEEE Trans. Very Large Scale Integr. Syst. **25**(3), 974–987 (2017)
55. Obermaisser, R., Salloum, C.E., Huber, B., Kopetz, H.: From a federated to an integrated automotive architecture. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **28**(7), 956–965 (2009)
56. Oetjens et al., J.: Safety evaluation of automotive electronics using virtual prototypes: State of the art and research challenges. In: 51st Design Automation Conference (DAC) (2014)
57. Otterness, N., et al.: An evaluation of the NVIDIA TX1 for supporting real-time computer-vision workloads. In: Parmer, G. (ed.) IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS) (2017)
58. Park, S., Zhang, L., Chakraborty, S.: Battery assignment and scheduling for drone delivery businesses. In: International Symposium on Low Power Electronics and Design (ISLPED) (2017)
59. Pazzaglia, P., et al.: DMAC: deadline-miss-aware control. In: 31st Euromicro Conference on Real-Time Systems (ECRTS) (2019)
60. Peters, N., et al.: Web browser workload characterization for power management on HMP platforms. In: 11th International Conference on Hardware/Software Codesign and System Synthesis (CODES) (2016)
61. Phan, L.T.X., Chakraborty, S., Thiagarajan, P.S.: A multi-mode real-time calculus. In: 29th IEEE Real-Time Systems Symposium (RTSS) (2008)
62. Phan, L.T.X., et al.: Composing functional and state-based performance models for analyzing heterogeneous real-time systems. In: 28th IEEE Real-Time Systems Symposium ((RTSS) (2007)
63. Pröbstl, A., Islam, B., Nirjon, S., Chang, N., Chakraborty, S.: Intelligent chargers will make mobile devices live longer. IEEE Des. Test **37**(5), 42–49 (2020)
64. Ramesh, S., et al.: Specification, verification and design of evolving automotive software. In: 54th Annual Design Automation Conference (DAC) (2017)
65. Åström, K.J., Wittenmark, B.: Computer-Controlled Systems (3rd Ed.). Prentice-Hall, Inc., USA (1997)

66. Roy, D., et al.: Multi-objective co-optimization of FlexRay-based distributed control systems. In: 22nd IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS) (2016)
67. Roy, D., et al.: Semantics-preserving cosynthesis of cyber-physical systems. Proceedings of the IEEE **106**(1), 171–200 (2018)
68. Roy, D., et al.: Automated synthesis of cyber-physical systems from joint controller/architecture specifications. In: Forum on Specification and Design Languages (FDL) (2016)
69. Sagstetter, F., Lukasiewycz, M., Chakraborty, S.: Generalized asynchronous time-triggered scheduling for FlexRay. IEEE Trans. on CAD of Integrated Circuits and Systems **36**(2), 214–226 (2017)
70. Sagstetter, F., et al.: Schedule integration framework for time-triggered automotive architectures. In: 51st Annual Design Automation Conference (DAC) (2014)
71. Sagstetter, F., et al.: Multischedule synthesis for variant management in automotive time-triggered systems. IEEE Trans. on CAD of Integrated Circuits and Systems **35**(4), 637–650 (2016)
72. Samal, K., Wolf, M., Mukhopadhyay, S.: Closed-loop approach to perception in autonomous system. In: Design, Automation & Test in Europe Conference & Exhibition (DATE) (2021)
73. Samii, S., et al.: Dynamic scheduling and control-quality optimization of self-triggered control applications. In: 31st IEEE Real-Time Systems Symposium (RTSS) (2010)
74. Schneider, R., et al.: Multi-layered scheduling of mixed-criticality cyber-physical systems. Journal of Systems Architecture - Embedded Systems Design **59**(10-D), 1215–1230 (2013)
75. Schneider, R., et al.: Constraint-driven synthesis and tool-support for flexray-based automotive control systems. In: 9th International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS) (2011)
76. Schneider, R., et al.: Compositional analysis of switched Ethernet topologies. In: Design, Automation and Test in Europe (DATE) (2013)
77. Tabuada, P.: Verification and Control of Hybrid Systems - A Symbolic Approach. Springer (2009)
78. Thiele, L., Wilhelm, R.: Design for timing predictability. Real-Time Systems **28**(2-3), 157–177 (2004)
79. Tibba, G., et al.: Testing automotive embedded systems under X-in-the-loop setups. In: 35th International Conference on Computer-Aided Design (ICCAD) (2016)
80. Tripakis, S.: Bridging the semantic gap between heterogeneous modeling formalisms and FMI. In: International Conference on Embedded Computer Systems: Architectures, Modeling, and Simulation (SAMOS) (2015)
81. Tripakis, S., Limaye, R., Ravindran, K., Wang, G.: On tokens and signals: Bridging the semantic gap between dataflow models and hardware implementations. In: Intl. Conf. on Embedded Computer Systems: Architectures, Modeling, and Simulation (SAMOS) (2014)
82. Voit, H., et al.: Optimizing hierarchical schedules for improved control performance. In: 5th IEEE International Symposium on Industrial Embedded Systems (SIES) (2010)
83. Wang, Z., Liang, H., Huang, C., Zhu, Q.: Cross-layer design of automotive systems. IEEE Des. Test **38**(5), 8–16 (2021)

84. Wang, Z., et al.: Bounding perception neural network uncertainty for safe control of autonomous systems. In: Design, Automation & Test in Europe Conference (DATE) (2021)
85. Waszecki, P., et al.: Automotive electrical and electronic architecture security via distributed in-vehicle traffic monitoring. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **36**(11), 1790–1803 (2017)
86. Wilhelm, R.: Determining reliable and precise execution time bounds of real-time software. IT Professional **22**(3), 64–69 (2020). https://doi.org/\tt 10.1109/MITP.2020.2972138
87. Xu, G., Xu, K., Zheng, C., Zhang, X., Zahid, T.: Fully electrified regenerative braking control for deep energy recovery and maintaining safety of electric vehicles. IEEE Trans. Veh. Technol. **65**(3), 1186–1198 (2016)
88. Xu, S., et al.: Safety-aware flexible schedule synthesis for cyber-physical systems using weakly-hard constraints. In: 28th Asia and South Pacific Design Automation Conference (ASPDAC) (2023)
89. Zhang, L., et al.: Task- and network-level schedule co-synthesis of ethernet-based time-triggered systems. In: 19th Asia and South Pacific Design Automation Conference (ASP-DAC) (2014)
90. Zhang, L., et al.: Schedule management framework for cloud-based future automotive software systems. In: 22nd IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA) (2016)