Combining Quantitative and Qualitative Analysis for Safe and Resilient Intelligent Hybrid Systems

Pauline Blohm[1], Paula Herber[1] and Anne Remke[1]

[1]University of Münster, Münster, Germany

Abstract

Model-driven development frameworks such as MATLAB Simulink are widely used in industrial design processes to conquer the increasing complexity of embedded control systems such as self-driving cars or critical infrastructures. As these systems are often safety-critical, formal methods to ensure safety, performance and resilience are highly desirable, in particular also in the presence of dynamic and uncertain environments. Formal verification has the potential to a) ensure that embedded systems function correctly for all possible system parameters and input scenarios, and b) provide statistical guarantees in the presence of uncertainty and probabilistic behavior. However, the application of existing formal verification and stochastic analysis techniques to embedded control systems is a major challenge, in particular if they are hybrid, i.e. combine discrete and continuous behavior, and include learning components to adapt to dynamic environments. To tackle this challenge, we aim at providing a quantitative analysis for intelligent Simulink models via a transformation to Stochastic Hybrid Automata (SHA) that gives us access to established analysis techniques for stochastic systems, such as reachability analysis or (statistical) model checking. To incorporate dynamic adaptations via learning, we investigate techniques to integrate domain-specific abstractions of the learning components into the SHA model. To ensure resilience of learning hybrid systems, we aim at combining the strengths of qualitative and quantitative analyses.

Keywords

Hybrid Systems, Formal Verification, Stochastic Failures, Learning, Simulink, Hybrid Automata

1. Problem

The demands on the functionality and flexibility of embedded control systems are steadily increasing. At the same time, they are more and more used in critical infrastructures, for example, controlling the supply of energy or water, and in safety-critical systems such as self-driving cars and other autonomous vehicles. With that, we increasingly use embedded control systems not only for our convenience or for profit, but also trust our lives and personal well-being to these systems. At the same time, learning components are nowadays often used to cope with dynamic environments. This makes it crucial to ensure the safety, performance, and resilience of these systems under all circumstances. Qualitative analysis techniques such as deductive verification can provide safety guarantees for hybrid systems, however, they typically only consider the worst case scenario. In contrast, quantitative analysis techniques like analytical reachability analysis or statistical model checking (SMC) can provide statistical guarantees for safety or performance properties even in the presence of uncertainty, however, they might not provide guarantees for all possible scenarios.

The integration of learning components and uncertainties further complicates formal verification of hybrid systems. Qualitative analysis techniques often rely on abstractions, such as contracts, to handle learning components or uncertainties. While these abstractions are necessary to provide safety guarantees, they usually abstract from all quantitative information, yielding imprecise and overly pessimistic results. In contrast, quantitative techniques exploit statistical information like the distribution of events or failure and repair times. However, they suffer from state-space explosion, in particular if learning components have to be verified to ensure safety under all circumstances or with high accuracy. Furthermore, quantitative analyses techniques typically do not provide us with

PhD Symposium of the 19th International Conference on Integrated Formal Methods (iFM) at the University of Manchester, UK, 12 November 2024.

pauline.blohm@uni-muenster.de (P. Blohm); paula.herber@uni-muenster.de (P. Herber); anne.remke@uni-muenster.de (A. Remke)

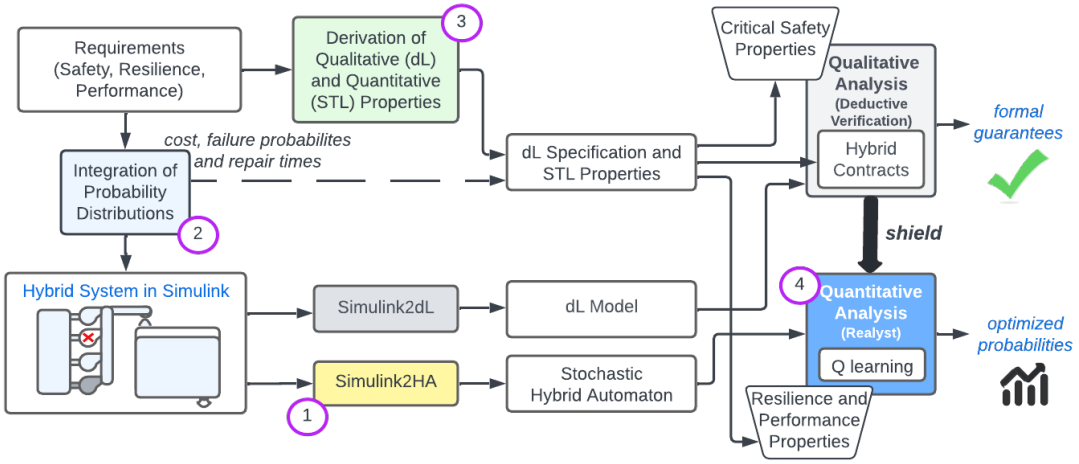

Figure 1: Combining the Strengths of Quantitative and Qualitative Analysis.

techniques for modular reasoning.

2. Proposed Solution

To ensure the safety and resilience of hybrid systems even in the presence of learning and in uncertain environments, we aim at combining the strengths of qualitative analysis with the strength of quantitative analysis. We focus on hybrid systems modelled in Simulink as it is widely adopted for embedded control systems that combine discrete and continuous behavior. Previous work of one of the co-authors has presented an approach for a qualitative analysis of Simulink models via a transformation to Differential Dynamic Logic (dL) [1, 2] which is implemented in the tool *Simulink2dL*. The resulting *dL Model* can then be analyzed using *Deductive Verification* to obtain formal guarantees that a system satisfies a given *Safety Property*. Our aim is to complement this with efficient and scalable quantitative analysis [3, 4, 5] and also to combine these techniques to provide an approach for comprehensive safety, resilience and performance analysis for intelligent hybrid systems. The concept of the thesis is shown in Fig. 1 and consists of four main parts:

1. We plan to provide an automated transformation *Simulink2SHA* from Simulink to Stochastic Hybrid Automata. With this transformation, we define a formal semantics for Simulink, and it gives us access to established *Quantitative Analysis* techniques, such as reachability analysis or (statistical) model checking.
2. We aim at investigating how to introduce stochastic components via *Probability Distributions* into the Simulink model to model uncertainties like component failure or sensor noise. With that, we can also investigate resilience and performance of a model under verification.
3. We aim at investigating how *Qualitative* and *Quantitative Properties* for the dL and SHA models can be derived from safety, resilience and performance requirements.
4. We plan to provide a technique to combine *Qualitative Analysis* results (e.g. from deductive verification) with a *Quantitative Analysis* for (potentially learning) Simulink models. In particular, we aim at investigating two different approaches: a) the integration of shielded SMC-based learning, which has been proposed by one of the co-authors in previous work [6], and b) the use of contracts or other domain-specific abstractions to safely integrate learning components, as has been proposed by two of the co-authors in [2, 7].

As a first step, we have presented an approach for a manual transformation from Simulink to SHA, which has been accepted at iFM 2024 [8]. Open research problems we plan to address in this thesis are how to model uncertainties such that the resulting models are still analyzable, how to capture qualitative and quantitative properties of intelligent systems appropriately and also how to combine

qualitative with quantitative analysis techniques for SHA with learning components. One key challenge is that, while modular reasoning would be highly desirable to handle complex systems, there exists no established concept to include quantitative and statistical information in contracts.

3. Related Work

There have been quite some efforts to enable the formal verification of systems that are modeled in Simulink [9, 10, 11, 12, 13, 14, 15]. Yet, all of these approaches, including the Simulink Design Verifier [16], are limited to discrete subsets of Simulink. Formal verification methods that support hybrid systems modeled in Simulink are, e.g., proposed in [1, 17, 18, 19, 20, 21]. However, they either focus on techniques for a special class of systems and do not provide general transformation rules for a broader set of blocks or focus on the qualitative analysis of safety properties and they neither take stochastic components nor learning into consideration.

There also has been a number of works on statistical model checking (SMC) for Simulink, for example [22, 23, 24]. Still they do not provide a stochastic model with formal semantics and thus cannot make use of more advanced quantitative analysis techniques. In [25], the authors propose a transformation from Simulink into stochastic timed automata (STA) and perform SMC with UPPAAL SMC on the resulting network of STA. However, they do not consider stochastic blocks and transform a given Simulink model into a deterministic STA model where all probabilities are one.

There also exists a broad variety of approaches to formally ensure safety of learning components using shielding or runtime monitoring [26, 27, 28]. These approaches do not directly support Simulink though, and they do not consider formal analysis techniques that take stochastic failures and learning into account.

Uppaal Stratego [29] uses priced timed automata to model stochastic behavior, and provides tooling for statistical model checking [30], timed games [31] and learning-based strategy synthesis [32]. While Uppaal Stratego comes with a graphical interface and is designed for usability, the underlying formalisms are less expressive than stochastic hybrid automata, in particular w.r.t. continuous system behavior governed by differential equations and controlled by continuous and stochastic variables.

Finally, there has been some work on combining rigorous formal and statistical methods. In [33], the authors incorporate statistical hypothesis testing to compute promising configurations of program verifiers automatically. However, they do not support hybrid systems, and they do not consider both safety and performance properties. In [34], the authors present a formal framework for an integrated qualitative and quantitative model-based safety analysis. However, they do not support hybrid systems and do not consider deductive verification methods.

4. Progress and Current State

The first step towards safe and resilient hybrid system is enabling a quantitative analysis for (stochastic) Simulink models. As Simulink does not offer elaborate quantitative analysis, such as reachability analysis or statistical model checking, a transformation into a formal model is desired. To tackle this problem, we are currently working on a modular approach to transform Simulink models into SHA. In [8], we present an approach that enables us to transform a subset of Simulink models to SHA and analyze the SHA using the tool RealySt [3] to obtain reachability probabilities for critical safety properties. This is an important first step towards ensuring safety and resilience of hybrid systems in the presence of uncertainties. However, it still has some limitations, e.g. we only provide transformation rules for a subset of Simulink blocks and the parallel composition has to be performed manually. To tackle these limitations, we are currently working on a tool to automatically transform a given Simulink model to an SHA using the transformation rules provided in [8]. Additionally, we plan to define transformation rules for a larger subset of Simulink blocks and provide better support for the integration of stochasticity into Simulink models, e.g. by providing parameterized subsystems that model specific stochastic behaviour.

As next steps, we plan to address the research challenges defined above, namely the integration of stochastic and learning components in Simulink, the derivation of qualitative and quantitative properties that are important for safety, resilience and performance of intelligent Simulink models in uncertain and dynamic environments, and the development of combined quantitative and quantitative analysis techniques that enable us to formally analyze these properties.

References

[1] T. Liebrenz, P. Herber, S. Glesner, Deductive verification of hybrid control systems modeled in Simulink with KeYmaera X, in: Int. Conference on Formal Engineering Methods, volume 11232 of *LNCS*, Springer, 2018, pp. 89–105. doi:10.1007/978-3-030-02450-5_6.

[2] J. Adelt, T. Liebrenz, P. Herber, Formal Verification of Intelligent Hybrid Systems that are modeled with Simulink and the Reinforcement Learning Toolbox, in: Formal Methods, volume 13047 of *LNCS*, Springer, 2021, pp. 349–366. doi:10.1007/978-3-030-90870-6_19.

[3] J. Delicaris, J. Stübbe, S. Schupp, A. Remke, Realyst: A C++ tool for optimizing reachability probabilities in stochastic hybrid systems, in: 16th EAI Int. Conference on Performance Evaluation Methodologies and Tools, volume 539 of *LNICS*, Springer, 2023, pp. 170–182. doi:10.1007/978-3-031-48885-6_11.

[4] C. Da Silva, S. Schupp, A. Remke, Optimizing reachability probabilities for a restricted class of stochastic hybrid automata via flowpipe construction, ACM Trans. Model. Comput. Simul. 33 (2023). doi:10.1145/3607197.

[5] M. Niehage, A. Remke, The best of both worlds: Analytically-guided simulation of hpngs for optimal reachability, in: Performance Evaluation Methodologies and Tools, Springer Nature, 2024, pp. 61–81. doi:10.1007/978-3-031-48885-6_5.

[6] M. Niehage, A. Hartmanns, A. Remke, Learning optimal decisions for stochastic hybrid systems, in: ACM-IEEE Int. Conference on Formal Methods and Models for System Design, ACM, 2021, pp. 44–55. doi:10.1145/3487212.3487339.

[7] P. Blohm, J. Adelt, P. Herber, Safe Integration of Learning in SystemC using Timed Contracts and Model Checking, in: R. von Hanxleden, S. A. Edwards, J. Brandt, Q. Zhu (Eds.), 21st ACM-IEEE International Symposium on Formal Methods and Models for System Design, ACM / IEEE, 2023, pp. 12–22. doi:10.1145/3610579.3611078.

[8] P. Blohm, P. Herber, A. Remke, Towards Quantitative Analysis of Simulink Models using Stochastic Hybrid Automata, in: International Conference on Integrated Formal Methods, accepted for publication, 2024.

[9] D. Araiza-Illan, K. Eder, A. Richards, Formal verification of control systems' properties with theorem proving, in: UKACC Int. Conference on Control, IEEE, 2014, pp. 244–249. doi:10.1109/CONTROL.2014.6915147.

[10] L. De Moura, N. Bjørner, Z3: An efficient SMT solver, in: International conference on Tools and Algorithms for the Construction and Analysis of Systems, Springer, 2008, pp. 337–340. doi:10.1007/978-3-540-78800-3_24.

[11] J.-C. Filliâtre, A. Paskevich, Why3 — where programs meet provers, in: European Symposium on Programming, Springer, 2013, pp. 125–128. doi:10.1007/978-3-642-37036-6_8.

[12] P. Herber, R. Reicherdt, P. Bittner, Bit-precise formal verification of discrete-time MATLAB/Simulink models using SMT solving, in: Int. Conference on Embedded Software, IEEE, 2013, pp. 1–10. doi:10.1109/EMSOFT.2013.6658586.

[13] S. K. Lahiri, S. A. Seshia, The UCLID decision procedure, in: Int. Conference on Computer Aided Verification, Springer, 2004, pp. 475–478. doi:10.1007/978-3-540-27813-9_40.

[14] R. Reicherdt, S. Glesner, Formal verification of discrete-time MATLAB/Simulink models using Boogie, in: Int. Conference on Software Engineering and Formal Methods, volume 8702 of *LNCS*, Springer, 2014, pp. 190–204. doi:10.1007/978-3-319-10431-7_14.

[15] M. Barnett, B.-Y. E. Chang, R. DeLine, B. Jacobs, K. R. M. Leino, Boogie: A modular reusable

verifier for object-oriented programs, in: Int. Symposium on Formal Methods for Components and Objects, Springer, 2005, pp. 364–387. doi:10.1007/11804192_17.

[16] The MathWorks, Simulink Design Verifier, https://de.mathworks.com/products/simulink-design-verifier.html, 2024.

[17] A. Chutinan, B. H. Krogh, Computational techniques for hybrid system verification, in: IEEE Trans. on Automatic Control, volume 48(1), IEEE, 2003, pp. 64–75. doi:10.1109/TAC.2002.806655.

[18] S. Minopoli, G. Frehse, SL2SX translator: from Simulink to SpaceEx models, in: Int. Conf. on Hybrid Systems: Computation and Control, ACM, 2016, pp. 93–98. doi:10.1145/2883817.2883826.

[19] L. Zou, N. Zhan, S. Wang, M. Fränzle, Formal Verification of Simulink/Stateflow Diagrams, in: Int. Symposium on Automated Technology for Verification and Analysis (ATVA), volume 9364 of *LNCS*, Springer, 2015, pp. 464–481. doi:10.1007/978-3-319-47016-0.

[20] M. Chen, X. Han, T. Tang, S. Wang, M. Yang, N. Zhan, H. Zhao, L. Zou, MARS: A toolchain for modelling, analysis and verification of hybrid systems, in: Provably Correct Systems, Springer, 2017, pp. 39–58. doi:10.1007/978-3-319-48628-4_3.

[21] T. Liebrenz, P. Herber, S. Glesner, A service-oriented approach for decomposing and verifying hybrid system models, in: Int. Conference on Formal Aspects of Component Software, volume 12018 of *LNCS*, Springer, 2019, pp. 127–146. doi:10.1007/978-3-030-40914-2_7.

[22] P. Zuliani, A. Platzer, E. M. Clarke, Bayesian statistical model checking with application to stateflow/simulink verification, Formal Methods in System Design 43 (2013) 338–367. doi:10.1007/s10703-013-0195-3.

[23] A. Legay, L.-M. Traonouez, Statistical model checking of simulink models with Plasma Lab, in: Formal Techniques for Safety-Critical Systems: 4th International Workshop, Springer, 2016, pp. 259–264. doi:10.1007/978-3-319-29510-7_15.

[24] B. Boyer, K. Corre, A. Legay, S. Sedwards, PLASMA-lab: A flexible, distributable statistical model checking library, in: Quantitative Evaluation of Systems: 10th International Conference, Springer, 2013, pp. 160–164. doi:10.1007/978-3-642-40196-1_12.

[25] P. Filipovikj, N. Mahmud, R. Marinescu, C. Seceleanu, O. Ljungkrantz, H. Lönn, Simulink to uppaal statistical model checker: Analyzing automotive industrial systems, in: International Symposium on Formal Methods, Springer, 2016, pp. 748–756. doi:10.1007/978-3-319-48989-6_46.

[26] M. Alshiekh, R. Bloem, R. Ehlers, B. Könighofer, S. Niekum, U. Topcu, Safe Reinforcement Learning via Shielding, Proceedings of the AAAI Conference on Artificial Intelligence 32 (2018). doi:10.5555/3504035.3504361.

[27] B. Könighofer, F. Lorber, N. Jansen, R. Bloem, Shield synthesis for reinforcement learning, in: International Symposium on Leveraging Applications of Formal Methods, Springer, 2020, pp. 290–306. doi:10.1007/978-3-030-61362-4_16.

[28] D. Phan, J. Yang, M. Clark, R. Grosu, J. Schierman, S. Smolka, S. Stoller, A Component-Based Simplex Architecture for High-Assurance Cyber-Physical Systems, in: 2017 17th International Conference on Application of Concurrency to System Design (ACSD), IEEE, 2017, pp. 49–58. doi:10.1109/ACSD.2017.23.

[29] A. David, P. G. Jensen, K. G. Larsen, M. Mikučionis, J. H. Taankvist, Uppaal stratego, in: Int. Conf. on Tools and Algorithms for the Construction and Analysis of Systems, Springer, 2015, pp. 206–211. doi:10.1007/978-3-662-46681-0_16.

[30] A. David, K. G. Larsen, A. Legay, M. Mikučionis, D. B. Poulsen, Uppaal smc tutorial, International Journal on Software Tools for Technology Transfer 17 (2015) 397–415.

[31] F. Cassez, A. David, E. Fleury, K. G. Larsen, D. Lime, Efficient on-the-fly algorithms for the analysis of timed games, in: M. Abadi, L. de Alfaro (Eds.), Concurrency Theory, Springer, Berlin, Heidelberg, 2005, pp. 66–80. doi:10.1007/11539452_9.

[32] P. Ashok, J. Křetínskỳ, K. G. Larsen, A. Le Coënt, J. H. Taankvist, M. Weininger, SOS: Safe, optimal and small strategies for hybrid markov decision processes, in: International Conference on Quantitative Evaluation of Systems, Springer, 2019, pp. 147–164. doi:10.1007/978-3-030-30281-8_9.

[33] A. Knüppel, T. Thüm, I. Schaefer, GUIDO: Automated Guidance for the Configuration of Deductive Program Verifiers, in: IEEE/ACM Int. Conference on Formal Methods in Software Engineering

(FormaliSE), IEEE, 2021, pp. 124–129. doi:10.1109/FormaliSE52586.2021.00018.

[34] M. Gudemann, F. Ortmeier, A framework for qualitative and quantitative formal model-based safety analysis, in: IEEE Int. Symposium on High Assurance Systems Engineering, IEEE, 2010, pp. 132–141. doi:10.1109/HASE.2010.24.