Logical Analysis of Hybrid Systems

André Platzer

Logical Analysis of Hybrid Systems

Proving Theorems for Complex Dynamics

Springer

Dr. André Platzer
Carnegie Mellon University
School of Computer Science
5000 Forbes Ave.
Pittsburgh PA 15213
USA
aplatzer@cs.cmu.edu

ISBN 978-3-642-14508-7 e-ISBN 978-3-642-14509-4
DOI 10.1007/978-3-642-14509-4
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2010934645

ACM Computing Classification (1998): F.4.1, F.3, D.2.4, I.2.3, G.1.7, I.2.8

Cover design: KuenkelLopka GmbH, Heidelberg

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Foreword

Hybrid Systems are notoriously hard to analyze and verify. So far, techniques based on either explicit or implicit state reachability have failed to scale with the size of such systems. Statistical Model Checking may prove useful, but sacrifices absolute certainty about the correctness of the answer obtained. In both cases, numerical error may change the truth-value of the result from False to True or vice versa. An alternative is to use a combination of decision procedures for real arithmetic and interactive theorem proving. Andre Platzer's Ph.D. thesis explores this alternative approach in great depth. He proposes a logic called Differential Dynamic Logic for specifying properties of Hybrid Systems, investigates the meta-theory of the logic, and gives inference rules for it. He has developed an extremely impressive graphical interface for the resulting tool KeYmaera, which is based on the KeY Prover for verifying Java programs, developed at the University of Karlsruhe, Chalmers, and Koblenz. Particularly noteworthy are the use of Differential Invariants for reasoning about complex Hybrid Systems and the examples that he is able to do: "The European Train Control System" and a curved flight roundabout maneuver for aircraft collision avoidance. Both examples are beyond the scope of current Hybrid System Model Checking tools. I believe that his verification tool is unique – there is no other one like it. I heartily recommend his book and theorem prover for those who need to verify complex cyber-physical systems.

Pittsburgh, February 2010 *Edmund M. Clarke*

Preface

The design of complex systems is essential to much of engineering and science. Equally essential is the effort to fully understand these systems and to develop tools and techniques that can steer us away from unsafe or incorrect designs. In civil engineering, for example, well-understood principles like statics can be used to analyse buildings before they are built, and refined architectural models can be used to predict whether a building will be safe or whether it might collapse during an earthquake. Similarly, in auto body design, wind tunnels and corresponding computer models based on computational fluid dynamics help engineers to gain an understanding of aerodynamic forces and wind resistance for energy efficiency before constructing the actual car and its chassis. Models and their analysis also play an important role in chip design and are used extensively in the semiconductor industry to prevent expensive bugs in hardware. Modelling and model analysis is thus an integral part of science and engineering and is used very effectively in many areas to ensure high-quality system designs, saving replacement cost and preventing dangerous side effects of malfunctioning designs.

Hybrid systems is an emergent area of growing importance, emphasising a systematic understanding of systems that combine discrete (e.g., digital) and continuous (e.g., analog or physical) effects. In fact, it is foreseeable that hybrid systems and the closely related notion of cyber-physical systems will soon play a ubiquitous role in engineering. Combinations of computation and control can lead to very powerful system designs, and computational aspects are being integrated into classical physical, mechanical, and chemical process controls on a routine basis today. The number of systems where both computational and physical aspects are important for really understanding them grows exponentially with modern technological advances. Hybrid systems occur frequently in automotive industries, aviation, railway applications, factory automation, process control, medical devices, mobile robotics, and mixed analog–digital chip design.

Despite the growing relevance in complex system designs, hybrid systems is an area where analytic approaches are still in their infancy. Hybrid systems occur ubiquitously and their analysis faces inherent complexity challenges. Hence, there is probably no other area where the gap is more noticeable between the tremendous

complexity of the systems we can build and the modest size of systems that we can analyse. Mankind can build systems that are significantly more complicated than people can understand analytically. This book presents an approach with logical analysis techniques that are intended to help overcome these difficulties and bridge the gap between design demand and analysis power.

In light of this growing interest in the field, the purpose of this book is to provide an introduction to hybrid systems analysis and, in particular, to present a coherent logical analysis approach for hybrid systems. One of the highly successful techniques used for analysing finite-state models in chip designs today is *model checking*, which was pioneered in 1981 by the 2007 ACM Turing Award Laureates Edmund M. Clarke, Allen Emerson, and Joseph Sifakis. Nowadays, model checking is used routinely in the semiconductor industry. Model checking is one of the inspirations for this work. Another area that is strongly related is interactive and *automated theorem proving*, which is also used in advanced industrial settings. Model checking and automated theorem proving complement each other to tackle various aspects of formal system verification. While both areas are ultimately rooted in logic, the basic operating principles are somewhat different. Model checking is based on systematically exploring the state space of a system in a clever way. Model checking searches for counterexamples, i.e., traces of a system that lead to a bug and that serve as a falsification of a correctness property. Impressive results have been demonstrated for finite-state systems where model checking is decidable. In theorem proving, in contrast, the notion of a proof is fundamental and represents a verification of a correctness property. In particular, a proof is a reason and explanation for *why* a system works. Automated theorem proving techniques that construct proofs automatically are another deep source of inspiration for the work presented here. In fact, several of the proof procedures presented in this book are inspired by theorem proving principles that have been used successfully for conventional object-oriented programs.

One important new aspect in hybrid systems is the cardinality and structure of the state space. In (sufficiently small) finite state spaces, for instance, exhaustive state exploration is still feasible, but becomes inherently impossible for the uncountable continuous state spaces of hybrid systems, especially with respect to their complicated interacting discrete and continuous dynamics. Most notably, the continuous dynamics of hybrid systems that is commonly described by differential equations poses significant new challenges compared to classical settings. Thus, verification techniques for differential equations are one very important part of hybrid systems analysis.

Outline

This book is intended as an introduction to hybrid systems and advanced analysis techniques for their dynamics. It covers basic and advanced notions of hybrid systems, specification languages for hybrid systems, verification approaches for hybrid systems, and application scenarios for hybrid systems verification. Starting from a basic background in mathematics and computer science, this book develops all

notions required for understanding and analysing hybrid systems. It also provides background material about logic and differential equations in the appendix.

This book presents a coherent logical foundation for hybrid systems analysis that will help the reader understand how behavioural properties of hybrid systems can be analysed successfully. The foundation developed here serves as a basis for advanced hybrid system analysis techniques. The hybrid systems analysis approach has also been implemented in the verification tool KeYmaera for hybrid systems, which is available for download at the book's Web page.

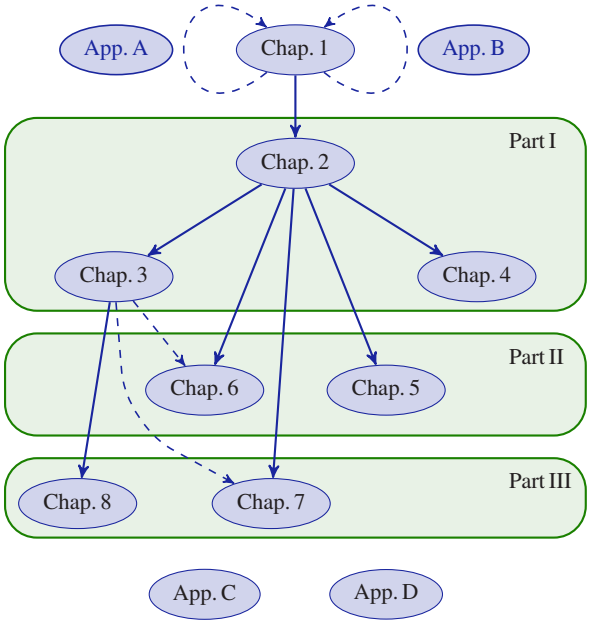
Part I describes specification and verification logics for hybrid systems that are the basis of hybrid systems analysis. It also covers constructive proof calculi that can be used to analyse and verify hybrid systems, including approaches for handling real arithmetic and differential equations. The chapters in Part I show a series of logical systems that are each presented in terms of their syntax, semantics, axiomatics, proof theory, and pragmatics. The syntax defines what can be said about hybrid system behaviour. The semantics gives meaning to the symbolic formulas and shows what we are ultimately interested in: truth, or, more precisely, what is true about the behaviour of the particular dynamics of a hybrid system. In the axiomatic parts, this book develops formal proof techniques that can be used by a human or machine to establish truth by proof. After all, truth that we do not know about is less helpful than truth that we can justify by giving a proof. The development of proof theory connects the semantic notion of truth in the real world with the syntactic device of formal proofs and shows that, in a sense of relative completeness, we can prove all true facts about hybrid systems from elementary properties of differential equations. Finally, this book shows the pragmatics of using verification procedures for analysing hybrid system scenarios. This includes both practical, algorithmic considerations of developing system analysis tools and various examples, application scenarios, and case studies that can be proven with the logics developed in Part I.

Part II focuses on the practical and algorithmic questions of how to turn the theoretical foundation from Part I into automated theorem proving procedures. This part also shows techniques for generating invariants and differential invariants of hybrid systems that are crucial for proving correctness, and shows how to overcome complexity challenges in real arithmetic verification. Part III shows how safety-critical properties of more advanced applications of hybrid systems in railway and aircraft control can be proven with the approach presented in Parts I and II. This part includes a study of collision avoidance in the European Train Control System (ETCS) and roundabout collision avoidance manoeuvres in air traffic control. Numerous examples, illustrations, and proofs throughout the text will also help the reader develop an intuition about hybrid systems behaviour and master the intricacies of the more subtle aspects in hybrid systems analysis.

How to Read This Book

The basic suggested reading sequence is linear (with additional consultation of the appendices for background information as needed). Except for the foundation of this

work that is laid out in Part I, however, the chapters are mostly kept self-contained so that they can also be studied independently. The following figure shows the reading order dependencies among the chapters (solid lines) and the partial dependencies of suggested reading sequences that hold for the advanced material of the respective chapters (dashed lines).



For background on classical first-order logic, we recommend you review App. A as needed. Depending on your interest, field of study, and preference, we recommend you either study the background information in App. A on first-order logic before reading Part I or use the material in App. A as a background reference book on demand while reading the main part of this book. Similarly, we recommend you review the background on ordinary differential equations in App. B either before or during the study of the main part. An intuitive approach to understanding differential equations and formal definitions of their semantics will be given throughout the text. Logic itself is also explained and illustrated intuitively during the main chapters, but some readers may also find it helpful to refresh, update, or learn about the basics of first-order logic from App. A before proceeding to the main part.

While there is a lot of flexibility in the reading sequence of the chapters, we strongly recommend you study the logical foundations of hybrid systems analysis in Chap. 2 of Part I before reading any other chapter of Parts I–III. Some more advanced sections in the applications in Part III also depend on the theory of differential invariants that is developed together with other extensions in Chap. 3.

Appendix C shows a formal relation of hybrid automata with hybrid programs. Appendix D gives more detail on the implementation of the approach put forth in this book in the verification tool KeYmaera. It also presents a survey of computational techniques for handling real arithmetic. Both App. C and D can be read as needed, after studying the introductory material and notions in Chap. 2. The most important formation rules for the logic and proof rules for the calculi are summarised at the end of the book.

Online Material for This Book

The Web page for this book provides online material, including the verification tool KeYmaera that implements our logical analysis approach for hybrid systems. We also provide slide material for parts of this book, an online tutorial for KeYmaera, and several KeYmaera problem files for examples from this book, including train and air traffic control studies. The book Web page is at the following URL:

`http://symbolaris.com/lahs/`

Acknowledgements

This book is based on my Ph.D. thesis and would not have been possible without the support of the PIs and collaborators on the projects that I have been working on. My sincere thanks go to Prof. Ernst-Rüdiger Olderog for his excellent advice and support, and for giving me the opportunity to work in one of the most fascinating areas of science in a group with a friendly and productive atmosphere. My advisor, Prof. Olderog, and the Director of AVACS, Prof. Werner Damm, both deserve my highest gratitude, not only for their continuous support and for their faith, but also for allowing me the freedom to pursue my own research ambitions in the stimulating context of the AVACS project ("Automatic Verification and Analysis of Complex Systems"). Ultimately, this made it possible for me to develop the logic and verification approach presented in this book.

I want to thank the external referees of my Ph.D. thesis, Prof. Tobias Nipkow from the Technical University of Munich and Prof. George J. Pappas from the University of Pennsylvania. It is an honour for me that they were willing to invest their valuable time and effort in the careful reviewing of my thesis. In fact, I am thankful to all members of my Ph.D. committee, Werner Damm, Ernst-Rüdiger Olderog, George J. Pappas, Tobias Nipkow, and Hardi Hungar for fruitful discussions and for the highest support they offered for my work.

I am especially grateful to Prof. Edmund M. Clarke, who invited me to Carnegie Mellon University several times, for his support, interest, and collaboration, and for sharing with me parts of his huge knowledge in all areas of formal methods. I further want to acknowledge the help by Prof. Peter H. Schmitt from the University of

Karlsruhe (TH), Profs. Bernhard Beckert and Ulrich Furbach from the University of Koblenz-Landau, Prof. Reiner Hähnle from the Chalmers University of Technology, Gothenburg, Sweden, Profs. Edmund M. Clarke and Frank Pfenning from Carnegie Mellon University, and Prof. Rajeev Goré from the Australian National University, Canberra, at various stages of my career.

I want to thank the program committee of the TABLEAUX 2007 conference for selecting my first paper on differential dynamic logic for the Best Paper Award, the first award at any TABLEAUX conference. This recognition has encouraged me to continue pursuing my research direction, which ultimately led to the results described in this book. I also thank the program committee of the FM 2009 conference for selecting my paper on formal verification of curved flight collision avoidance maneuvers for the Best Paper Award. I am very grateful to the ACM Doctoral Dissertation Award committee for honoring my Ph.D. thesis with the 2009 ACM Doctoral Dissertation Honorable Mention Award.

I am truly thankful to my colleagues at Carnegie Mellon University for their encouraging feedback about my work and for the friendly and constructive atmosphere at CMU. For many fruitful discussions I thank my colleagues and friends from Oldenburg, Ingo Brückner, Henning Dierks, Johannes Faber, Sibylle Fröschle, Jochen Hoenicke, Stephanie Kemper, Roland Meyer, Michael Möller, Jan-David Quesel, Tim Strazny, and especially my office mate Andreas Schäfer. Ernst-Rüdiger Olderog, Johannes Faber, Ingo Brückner, Roland Meyer, Henning Dierks, Silke Wagner, Nicole Betz, Alex Donzé, and especially Andreas Schäfer also deserve credit for proofreading some of my earlier papers, which formed the basis for this book. I also acknowledge Andreas Schäfer's helpful feedback from proofreading parts of this book. I appreciate the feedback of my students on this book.

Furthermore, I thank Jan-David Quesel for writing a Master's thesis under my supervision and for his invaluable support with the implementation of the verification tool KeYmaera based on the techniques that I present in this book and in prior publications. I also thank him for help with the experiments and ETCS. I am also thankful for indispensable and reliable help from Richard Bubel and Philipp Rümmer with the implementation internals of the KeY basis. I thank the whole KeY team for providing the impressive Java verification tool KeY as a basis for our implementation of KeYmaera.

For help with the book process, I thank Ronan Nugent from Springer.

Especially, I thank my parents, Rudolf and Brigitte Platzer, and my sister, Julia, for their continuous support and encouragement, and I thank my wife, Nicole, for her true faith in me. She also deserves credit for her invaluable help with some of the illustrations in this book.

Funding

This research was partly supported by the German Research Council (DFG) under grant SFB/TR 14 AVACS ("Automatic Verification and Analysis of Complex Systems", see `http://www.avacs.org`); a Transregional Collaborative Research

Center of the Max Planck Institute and the Universities of Oldenburg, Saarbrücken, and Freiburg in Germany, with associated cooperations with the University of Pennsylvania, ETH Zürich, and the Academy of Sciences of the Czech Republic. It was further supported partly by a research fellowship of the German Academic Exchange Service (DAAD) and by a research award of the Floyd und Lili Biava Stiftung. Some part of this work was also supported by the National Science Foundation under grant nos. CNS-0931985 and CNS-0926181, including the NSF Expedition on Computational Modeling and Analysis of Complex Systems (CMACS); see `http://cmacs.cs.cmu.edu` for more information.

The views and conclusions contained in this book are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of any sponsoring institution or government.

Further Sources

This book is based on several sources, most notably the author's Ph.D. thesis [236]. Chapter 2 is an extended version of an article in the Journal of Automated Reasoning [235] and also covers some material from previous work at TABLEAUX [231] and HSCC [232]. Chapter 3 is an extended version of an article in the Journal of Logic and Computation [237], to which we now add a relative completeness argument and prove that DAL is a conservative extension of the sublogic $\mathsf{d}\mathscr{L}$. We further combine the solution-based techniques from Chap. 2 with differential induction-based techniques from Chap. 3 by introducing the new extension of differential monotonicity relaxations. Chapter 4 is a substantially extended version of a previous paper at LFCS [233], to which we now add a complete and more elegant calculus and provide a modular relative completeness proof.

In Chap. 5, we extend a previous paper at VERIFY [230] with more details on iterative background closure strategies, including experimental evaluation, and complement this proof technique with a new iterative inflation strategy. Chapter 6 is based on joint work with Edmund M. Clarke at CAV [239] and in Formal Methods in System Design [240].

Chapter 7 is a substantially revised and improved version of joint work with Jan-David Quesel at HSCC [243] with extensions from follow-up work [244]. Chapter 8 is a significantly improved and detailed case study developed on the basis of joint work with Edmund M. Clarke at HSCC [238] and CAV [239] with subsequent extensions at FM [241].

Appendix B summarises classical results from the theory of differential equations from the literature [297]. Finally, App. D uses a few excerpts from joint work with Jan-David Quesel at IJCAR [242], adding an overall discussion of the KeYmaera verification tool that implements the approach presented in this book. Appendix D also adds a thorough description of computational back-ends for real arithmetic, with extensions from joint work with Philipp Rümmer and Jan-David Quesel [246].

Pittsburgh, February 2010 *André Platzer*

Contents

List of Figures

List of Tables

List of Theorems