

# Preface

Our modern life increasingly depends on embedded systems. How to develop complex embedded systems correctly is a grand challenge for computer science and control theory. The model-based method is thought to be an effective approach to the design of complex embedded systems. Using this approach at the very beginning, an abstract model of the system to be developed is defined. Extensive analysis and verification on the abstract model are then committed so that errors can be identified and corrected at the very early stage. Then the higher-level abstract model is refined to a lower-level abstract model, even to source code, step by step, using model transformation techniques.

Model-based design is supported by Simulink/Stateflow (S/S) and has been widely adopted in the industry. Simulink is an environment for the model-based analysis and design of embedded control systems, which offers an intuitive graphical modelling language reminiscent of circuit diagrams and thus appealing to the practising engineer. Stateflow is a toolbox adding facilities for modelling and simulating reactive systems by means of hierarchical statecharts, extending Simulink's scope to event-driven and hybrid forms of embedded control. Modelling, analysis, and design using S/S have become a de facto standard in the embedded systems industry.

Simulink/Stateflow relies on extensive simulation based on unverified numerical computation to validate system requirements, which is prone to incomplete coverage of open systems and possible unsoundness of analysis results due to numerical errors. As a result, existing errors in the model might not have been discovered through simulation. Once such incorrectly developed systems are deployed, catastrophe can be caused by system failure, especially in safety-critical fields.

Formal verification techniques help to remedy the problems of simulation. For example, industrial standards such as DO-178C, IEC 61508, EN 50128, ISO 26262, etc. all advocate the use of formal or semiformal methods as complement to conventional methods, to develop embedded systems with high safety levels. Actually, Simulink does support formal analysis of S/S models, e.g. using the tool *Simulink Design Verifier*, which adopts formal methods to identify hidden design errors in models without extensive simulation runs. However, currently, it can only detect blocks in the model that result in low-level errors such as integer overflow,

dead logic, array access violations, division by zero, and so on, but not system-level properties of the complete model with the physical and environmental aspects taken into account. On the other hand, existing formal methods in the literature that deal with continuous-time S/S models lack either expressiveness or scalability. The rigorous safety requirements of safety-critical embedded systems demands more effective formal verification techniques for S/S.

This book presents a state-of-the-art formal approach to the safety verification of continuous-time S/S diagrams. This approach is called *deductive* as it relies on a process-like modelling language and a deductive proof system built upon the language. Formal verification of S/S models is achieved by building an interface, namely, bidirectional translations, between S/S models and the formal models in Hybrid CSP (HCSP), a formal modelling language for hybrid discrete-continuous systems. Through such an interface, S/S models are firstly translated into HCSP, and the correctness of such translations can be justified by formal proofs or by inverse translation from HCSP to S/S and co-simulation. This provides a gateway to the mechanized verification of S/S models, since formal analysis of HCSP models is supported by a deductive proof system called Hybrid Hoare Logic (HHL), which is implemented using the interactive proof assistant Isabelle/HOL.

The presented approach has the following prominent features: first, the HCSP language has a rich set of composition primitives, facilitating the modelling of numerous concurrencies and communications of S/S diagrams; second, the HHL-based deductive reasoning of HCSP models is compositional and is incorporated with powerful invariant generation techniques for dealing with continuous dynamics and thus avoids exhaustive exploration of the state space and is more scalable; third, the usefulness of the approach is demonstrated by impressive real-world case studies originating from the railway and aerospace industries.

This book is intended for researchers, graduate students, and engineering practitioners in the fields of formal methods and embedded systems. From this book, the readers will learn the HCSP/HHL-based deductive method and the use of corresponding tools for formal verification of S/S diagrams. Moreover, for those who are not familiar with formal methods, they will gain some general knowledge about the fundamental elements and common techniques in developing a deductive formal verification approach, especially for embedded systems. Finally, by investigating the successful case studies, the readers will realize how formal methods can contribute to real industry, and hopefully will be inspired to start to use the proposed approach or even develop their own formal methods in their future work.

## How to Read the Book

This book has grown out of our research in formal verification of hybrid/embedded systems over the past few years. Lots of the presented materials originate from published conference or journal papers and tutorial lectures given at three international

conferences, but have been reorganized in a systematic and consistent way. Besides, much improvement has been made upon the published results, e.g. a revised proof system of Hybrid Hoare Logic which can deal with general recursion in Chap. 7; the formal proof of correctness of the two-way translation between S/S and HCSP models in Sects. 10.5, 11.2, and 12.2; and so on, are our latest work and have not appeared elsewhere.

The main body of this book excluding Introduction consists of Chaps. 2–14, which can be roughly divided into five parts.

- Chapters 2 and 3 briefly introduce some basic theories fundamental to the understanding of many proofs in this book, including dynamical and control systems, algebraic geometry, first-order theory of reals, Unifying Theories of Programming (UTP), etc. Those who are familiar with these theories can skip the two chapters.
- Chapters 4 and 5 give a brief introduction to the S/S modelling environment for self-containedness. For those who are familiar with or expert in S/S, he/she can go quickly through or just skip this part.
- Chapters 6, 7, 8, and 9 present the HCSP modelling language, the HHL proof system and its implementation, and differential invariant generation techniques, which together form the cornerstone of the formal verification of S/S models in this book. This is the most theoretical part of this book and basic knowledge in first-order logic, duration calculus, ordinary differential equations, etc. are prerequisites. In particular, Chap. 9 on differential invariant generation is quite involved and somehow independent of Chaps. 6, 7, and 8, so one can just get some general ideas about this chapter without going into details if he/she is not working on this specific topic.
- Chapters 10, 11, and 12 establish the translations between S/S and HCSP models step by step. Correctness of such translations is justified by formal proofs based on UTP, and thus guarantees the scientific rigor of our approach. Nevertheless, skipping the proof details will not affect the understanding of the approach much.
- Chapters 13 and 14 illustrate the implementation and use of the toolkit MARS, as well as its application in real-world case studies. It is highly recommended that the readers investigate the case studies by playing with the toolkit using the online resources at the following addresses:
  - The MARS toolkit together with the case study on the descent guidance control of a lunar lander can be accessed at [http://lcs.ios.ac.cn/~znj/tools/MARS\\_v1.1.zip](http://lcs.ios.ac.cn/~znj/tools/MARS_v1.1.zip).
  - The individual HHL prover (see Chap. 8) that implements both deep and shallow embeddings can be accessed at <http://lcs.ios.ac.cn/~znj/tools/hhlprover.zip>.
  - The case study on the operation scenarios of Chinese High-Speed Train Control System at Level 3 (CTCS-3) can be accessed at <http://lcs.ios.ac.cn/~znj/tools/CTCS-3.zip>.

The following table demonstrates the subject of each of Chaps. 6–14 and its dependency on published source materials, the full list of which is appended to the end of this preface. By consulting these source publications, the readers can learn some additional materials that are not selected when writing this book, either because they are obsolete due to subsequent updates or because they are not closely related to the topic of this book, so that can get a general idea about the development history of our approach.

Chapters' dependency on source publications		
Chapter no.	Main content	Source publications
Chap. 6	The modelling language HCSP	[5, 8]
Chap. 7	The HHL proof system	[4, 5, 9]
Chap. 8	Implementation of the HHL prover	[10]
Chap. 9	Differential invariant generation	[6, 7]
Chap. 10	Translation of Simulink models to HCSP	[14]
Chap. 11	Translation of Stateflow models to HCSP	[16]
Chap. 12	Translation of HCSP to Simulink models	[2]
Chap. 13	The integrated toolchain MARS	[1, 10]
Chap. 14	Case studies	[3, 13, 15]

Besides, at the end of most chapters, a review of literature on related work, over which we have made improvements, or those parallel to our work, is provided for further reading.

## Acknowledgements

Our work on formal verification of hybrid/embedded systems was initiated, and the adoption of the HCSP-based deductive approach was advocated, by Prof. Chaochen Zhou and Prof. Naijun Zhan at the Institute of Software, Chinese Academy of Sciences (ISCAS), in 2009. Ever since, many of our colleagues and collaborators have contributed to the development of the proposed approach.

Our initial version of HHL was mainly attributed to Prof. Chaochen Zhou [5]. Prof. Dimitar P. Guelev from the Institute of Mathematics and Informatics, Bulgarian Academy of Sciences, helped to develop improved versions of HHL [4, 9]. Our work on differential invariant generation was mainly done with Dr. Jiang Liu when he was a postdoc at ISCAS and later a research professor at the Chongqing Institute of Green and Intelligent Technology, Chinese Academy of Sciences [6, 7]. The work on translating S/S diagrams into HCSP was mainly attributed to Dr. Liang Zou, and Prof. Martin Fränzle from the University of Oldenburg, Germany, and Prof. Shengchao Qin from Teesside University, UK, both gave great advice on this work [14, 16]. The work on inverse translation from HCSP to Simulink

is a joint work with Mr. Mingshuai Chen from ISCAS, Prof. Anders P. Ravn from Aalborg University, Denmark, and Prof. Mengfei Yang from the Chinese Academy of Space Technology [2]. Mr. Mingshuai Chen and Mr. Xiao Han from Beijing Jiaotong University (BJTU) also contributed a lot to the development of the toolchain MARS [1].

The formal modelling and verification of Chinese High-Speed Train Control System at Level 3 (CTCS-3) are the fruit of our long-term collaboration with a research team led by Prof. Tao Tang from BJTU [3, 15]. Actually, to formally verify CTCS-3 is the initial inspiration and intention of launching our research work on hybrid system verification in 2009. Over the years, Drs. Jidong Lv, Lei Yuan, Datian Zhou, etc. from BJTU, as well as Dr. Liang Zou, Mr. Zhao Quan, Ms. Danqing Guo, etc. from ISCAS, have been intensively involved in this work. The formal verification of the descent guidance control program of a Chinese lunar lander is the outcome of a 4-year (2012–2015) NSFC joint project [13], which is achieved under the guidance of project members Prof. Mengfei Yang from the Chinese Academy of Space Technology, and Prof. Bin Gu and Drs. Yao Chen, Yanxia Qi, Zheng Wang, etc. from Beijing Institute of Control Engineering, China.

We thank Profs. Lu Yang, Bican Xia, and Deepak Kapur; Drs. Ehsan Ahmad, Yangjia Li, Ming Xu, and Jiaqi Zhu; Mr. Yang Gao and Yu Peng; and many others for their valuable comments and helpful discussions on the materials of this book.

We thank the editors of this monograph for their patience in allowing our request for extending the manuscript submission deadline for several times.

We would like to point out that it is difficult to list all the contributors to this book or those who have helped us in any form to complete such a work, and we sincerely apologize for any omitted name.

The work in this monograph has been supported partly by “973 Program” under grant No. 2014CB340701, by NSFC under Grants 91418204 and 91118007, by CDZ project CAP (GZ 1023), and by the CAS/SAFEA International Partnership Program for Creative Research Teams.

Beijing, China  
Beijing, China  
Chongqing, China  
August 2016

Naijun Zhan  
Shuling Wang  
Hengjun Zhao

Formal Verification of Simulink/Stateflow Diagrams

A Deductive Approach

Zhan, N.; Shuling, W.; Zhao, H.

2017, XV, 258 p. 74 illus., 60 illus. in color., Hardcover

ISBN: 978-3-319-47014-6