Languages and Tools for Hybrid Systems Design

Languages and Tools for Hybrid Systems Design

Luca P. Carloni
Columbia University

Roberto Passerone
Cadence Berkeley Laboratories

Alessandro Pinto
University of California at Berkeley

Alberto L. Sangiovanni-Vincentelli
University of California at Berkeley

the essence of knowledge

Boston – Delft

Foundations and Trends® in Electronic Design Automation

Published, sold and distributed by:
now Publishers Inc.
PO Box 1024
Hanover, MA 02339
USA
Tel. +1-781-985-4510
www.nowpublishers.com
sales@nowpublishers.com

Outside North America:
now Publishers Inc.
PO Box 179
2600 AD Delft
The Netherlands
Tel. +31-6-51115274

A Cataloging-in-Publication record is available from the Library of Congress

The preferred citation for this publication is L.P. Carloni, R. Passerone, A. Pinto, A.L. Sangiovanni-Vincentelli, Languages and Tools for Hybrid Systems Design, Foundation and Trends® in Electronic Design Automation, vol 1, no 1/2, pp 1–193, 2006

Printed on acid-free paper

ISBN: 1-933019-28-X

Foundations and Trends® in Electronic Design Automation
Volume 1 Issue 1/2, 2006

Editorial Scope

Foundations and Trends® in Electronic Design Automation will publish survey and tutorial articles in the following topics:

- System Level Design
- Behavioral Synthesis
- Logic Design
- Verification
- Test
- Physical Design
- Circuit Level Design
- Reconfigurable Systems
- Analog Design

Information for Librarians

Foundations and Trends® in Electronic Design Automation, 2006, Volume 1, 4 issues. ISSN paper version 1551-3939. ISSN online version 1551-3947. Also available as a combined paper and online subscription.

Foundations and Trends® in
Electronic Design Automation
Vol. 1, No 1/2 (2006) 1–193

now
the essence of knowledge

Languages and Tools for Hybrid Systems Design

Luca P. Carloni[1], Roberto Passerone[2], Alessandro Pinto[3] and Alberto L. Sangiovanni-Vincentelli[4]

[1] *Department of Computer Science, Columbia University, 1214 Amsterdam Avenue, Mail Code 0401, New York, NY 10027, USA, luca@cs.columbia.edu*
[2] *Cadence Berkeley Laboratories, 1995 University Ave Suite 460, Berkeley, CA 94704, USA, robp@cadence.com*
[3] *Department of EECS, University of California at Berkeley, Berkeley, CA 94720, USA, pinto@eecs.berkeley.edu*
[4] *Department of EECS, University of California at Berkeley, Berkeley, CA 94720, USA, alberto@eecs.berkeley.edu*

Abstract

The explosive growth of embedded electronics is bringing information and control systems of increasing complexity to every aspects of our lives. The most challenging designs are safety-critical systems, such as transportation systems (e.g., airplanes, cars, and trains), industrial plants and health care monitoring. The difficulties reside in accommodating constraints both on functionality and implementation. The correct behavior must be guaranteed under diverse states of the environment and potential failures; implementation has to meet cost, size, and power consumption requirements. The design is therefore subject to extensive mathematical analysis and simulation. However, traditional

models of information systems do not interface well to the continuous evolving nature of the environment in which these devices operate. Thus, in practice, different mathematical representations have to be mixed to analyze the overall behavior of the system. *Hybrid systems* are a particular class of mixed models that focus on the combination of discrete and continuous subsystems. There is a wealth of tools and languages that have been proposed over the years to handle hybrid systems. However, each tool makes different assumptions on the environment, resulting in somewhat different notions of hybrid system. This makes it difficult to share information among tools. Thus, the community cannot maximally leverage the substantial amount of work that has been directed to this important topic. In this paper, we review and compare hybrid system tools by highlighting their differences in terms of their underlying semantics, expressive power and mathematical mechanisms. We conclude our review with a comparative summary, which suggests the need for a unifying approach to hybrid systems design. As a step in this direction, we make the case for a *semantic-aware interchange format*, which would enable the use of joint techniques, make a formal comparison between different approaches possible, and facilitate exporting and importing design representations.

Contents

1 Introduction **1**

2 Foundations **7**

2.1 Formal definition of hybrid systems 7
2.2 Examples 11

3 Tools for Simulation **19**

3.1 Simulink and Stateflow 20
3.2 Modelica 34
3.3 HyVisual 47
3.4 Scicos 56
3.5 Shift 68
3.6 Charon 80

4 Tools for Formal Verification **93**

4.1 Introduction to verification methods 97
4.2 Hytech 100
4.3 Masaccio 109
4.4 CheckMate 115
4.5 PHAVer 125

4.6 HSolver 133
4.7 Ellipsoidal calculus for reachability 139
4.8 d/dt 142
4.9 Hysdel 147

5 Comparative Summary 161

6 The Future: Towards the Development of a Standard Interchange Format 171

6.1 Semantic-free and semantically-inclusive interchange formats in EDA 172
6.2 The hybrid system interchange format 174
6.3 Requirements for a standard interchange format 175
6.4 Metropolis-based abstract semantics for hybrid systems 177
6.5 Conclusions 179

Acknowledgements 181

References 183

1

Introduction

With the rapid advances in implementation technology, designers are given the opportunity of building systems whose complexity far exceeds the increase in rate of productivity afforded by traditional design paradigms. Design time has thus become the bottleneck for bringing new products to market. The most challenging designs are in the area of safety-critical embedded systems, such as the ones used to control the behavior of transportation systems (e.g., airplanes, cars, and trains) or industrial plants. The difficulties reside in accommodating constraints both on functionality and implementation. Functionality has to guarantee correct behavior under diverse states of the environment and potential failures; implementation has to meet cost, size, and power consumption requirements.

When designing embedded systems of this kind, it is essential to take all effects, including the interaction between environment (plant to be controlled) and design (digital controller) into consideration. This calls for methods that can deal with heterogeneous components exhibiting a variety of different behaviors. For example, digital controllers can be represented mathematically as discrete event systems, while plants are mostly represented by continuous time systems whose behavior is captured by partial or ordinary differential equations. In addition, the complexity of the plants is such that representing them at the detailed level is often impractical or even impossible. To cope with this complexity, abstraction is a very powerful method. Abstraction consists in eliminating details that do not affect the behavior of the system that we may be interested in. In both cases, different mathematical representations have to be mixed to analyze the overall behavior of the controlled system.

There are many difficulties in mixing different mathematical domains. *In primis*, the very meaning of interaction may be challenged. In fact, when heterogeneous systems are interfaced, interface variables are defined in different mathematical domains that may be incompatible. This aspect makes verification and synthesis impossible, unless a careful analysis of the interaction semantics is carried out.

In general, pragmatic solutions precede rigorous approaches to the solution of engineering problems. This case is no exception. Academic institutions and private software companies started developing computational tools for the simulation, analysis, and implementation of control systems (e.g., Simulink, Stateflow and Matlab from The Mathworks), by first deploying common sense reasoning and then trying a formalization of the basic principles. These approaches focused on a particular class of heterogeneous systems: systems featuring the combination of discrete-event and continuous-time subsystems. Recently, these systems have been the subject of intense research by the academic community because of the interesting theoretical problems arising from their design and analysis as well as of the relevance in practical applications [2, 92, 133]. These systems are called *hybrid systems* [12, 14, 17, 18, 19, 20, 63, 80, 98, 131, 132, 134, 140, 163, 168].

Hybrid systems have proven to be powerful design representations for system-level design. While Simulink, Stateflow and Matlab together provide excellent practical modeling and simulation capability for the design capture and the functional verification via simulation of embedded systems, there is a need for a more rigorous and domain-specific analysis as well as for methods to refine a high-level description into an implementation. There is a wealth of tools and languages that have been proposed over the years to handle hybrid systems. Each tool or language is based on somewhat different notions of hybrid systems and on assumptions that make a fair comparison difficult. In addition, sharing information among tools is almost impossible at this time, so that the community cannot leverage maximally the substantial amount of work that has been directed to this important topic.

In this survey, we collected data on available languages, formalism and tools that have been proposed in the past years for the design and verification of hybrid systems. We review and compare these tools by highlighting their differences in the underlying semantics, expressive power and solution mechanisms. Table 1.1 lists tools and languages reviewed in this survey with information on the institution that supports the development of each project as well as pointers to the corresponding web site[1] and to some relevant publications.

The tools are covered in two main sections: one dedicated to simulation-centric tools including commercial offerings, one dedicated to formal verification-centric tools. The simulation-centric tools are the most popular among designers as they pose the least number of constraints on the systems to be analyzed. On the other hand, their semantics are too general to be amenable to formal analysis or synthesis. Tools based on restricted expressiveness of the description languages (see, for example, the synthesizable subset of RTL languages as a way of allowing tools to operate on a more formal way that may yield substantial productivity gains) do have an appeal as they may be the ones to provide the competitive edge in terms of quality of results and cost for obtaining them. The essence is to balance the gains in analysis and synthesis power versus the loss of expressive power.

We organized each section describing a tool in

(1) a brief introduction to present the tool capabilities, the organizations supporting it and how to obtain the code;

[1] George Pappas research group at the Univ. of Pennsylvania is maintaining a WikiWiki-Web site at `http://wiki.grasp.upenn.edu/ graspdoc/hst/` whose objective is to serve as a community depository for software tools that have been developed for modeling, verifying, and designing hybrid and embedded control systems. It provides an “evolving” point of reference for the research community as well as potential users of all available technology and it maintains updated links to online resources for most of the tools listed on Table 1.1.

Table 1.1 References for the various modeling approaches, toolsets.

Name	Institution	Web Page	References	Section
CHARON	Univ. of Pennsylvania	www.cis.upenn.edu/mobies/charon/	[3, 4, 8]	3.6
CHECKMATE	Carnegie Mellon Univ.	www.ece.cmu.edu/~webk/checkmate/	[151]	4.4
d/dt	Verimag	www-verimag.imag.fr/~tdang/Tool-ddt/ddt.html	[53, 21, 22]	4.8
DYMOLA	Dynasim AB	www.dynasim.se/	[67]	3.2
ELLIPSOIDAL TOOLBOX	UC Berkeley	www.eecs.berkeley.edu/~akurzhan/ellipsoids/	[113, 120, 119]	4.7
HSOLVER	Max-Planck-Institut	www.mpi-inf.mpg.de/~ratschan/hsolver/	[147]	4.6
HYSDEL	ETH Zurich	www.control.ee.ethz.ch/~hybrid/hysdel/	[166, 165]	4.9
HYTECH	Cornell, UC Berkeley	www-cad.eecs.berkeley.edu/~tah/HyTech	[11, 88, 95]	4.2
HYVISUAL	UC Berkeley	ptolemy.eecs.berkeley.edu/hyvisual	[103]	3.3
MASACCIO	UC Berkeley	www.eecs.berkeley.edu/~tah	[97]	4.3
MODELICA	Modelica Association	www.modelica.org	[71, 162, 70]	3.2
PHAVER	VERIMAG	www.cs.ru.nl/~goranf/	[69]	4.5
SCICOS	INRIA	www.scicos.org	[64, 143]	3.4
SHIFT	UC Berkeley	www.path.berkeley.edu/shift	[60, 61]	3.5
SIMULINK	The MathWorks	www.mathworks.com/products/simulink	[15, 52, 148]	3.1
STATEFLOW	The MathWorks	www.mathworks.com/products/stateflow	[15, 52, 148]	3.1
SYNDEX	INRIA	www-rocq.inria.fr/syndex	[78, 79]	3.4

(2) a section describing the syntax of the language that describes the system to be analyzed;
(3) a section describing the semantics of the language;
(4) the application of the language and tool to two examples that have been selected to expose its most interesting features;
(5) a discussion on its pros and cons.

In the last part of the survey we provide a comparative summary of the hybrid system tools that we have presented. The resulting landscape appears rather fragmented. This suggests the need for a unifying approach to hybrid systems design. As a step in this direction, we make the case for a *semantic-aware interchange format*. Today, re-modeling the system in another tool's modeling language, when (at all) possible, requires substantial manual effort and maintaining consistency between models is error-prone and difficult in the absence of tool support. The interchange format, instead, would enable the use of joint techniques, make a formal comparison between different approaches possible, and facilitate exporting and importing design representations. The popularity of Matlab, Simulink, and Stateflow implies that significant effort has already been invested in creating a large model-base in Simulink/Stateflow. It is desirable that application developers take advantage of this effort without foregoing the capabilities of their own analysis and synthesis tools. We believe that the future will be in automated semantic translators that, for instance, can interface with and translate the Simulink/Stateflow models into the models of different analysis and synthesis tools.

Survey organization. In Section 2, we lay the foundation for the analysis. In particular, we review the formal mathematical definition of hybrid systems (Section 2.1) and we define two examples (Section 2.2), a system of three point masses and a full wave rectifier, which will be used to compare and explain the tools and languages presented in this survey. In Section 3 we introduce and discuss the most relevant tools for simulation and design of hybrid and embedded systems. With respect to the industrial offering, we present the Simulink/Stateflow design environment, the Modelica language, and the modeling and simulation tool Dymola based on it. Among the academic tools, we summarize the essential features of Scicos, Shift, HyVisual and Charon, a tool that is the bridge between the simulation tools and the formal verification tools as it supports both (although the verification component of Charon is not publicly available). In Section 4, we focus on tools for formal verification of hybrid systems. In particular, we discuss HyTech, PHAVer, HSolver, Masaccio, CheckMate, d/dt and Hysdel. The last two can also be used to synthesize a controller that governs the behavior of the system to follow desired patterns. We also summarize briefly tools based on the ellipsoidal calculus like Ellipsoidal Toolbox. In Section 5 we give a comparative summary of the design approaches, languages, and tools presented throughout this survey. To end in Section 6, we offer a discussion and a plan on the issues surrounding the construction of the interchange format.

References

[1] R. Alur, C. Courcoubetis, and D. Dill, "Model checking in dense real time," *Information and Computation*, vol. 104, pp. 2–34, May 1993.

[2] R. Alur, C. Courcoubetis, N. Halbwachs, T. A. Henzinger, P. H, X. Nicollin, J. Sifakis, and S. Yovine, "The algorithmic analysis of hybrid systems," *Theoretical Computer Science*, vol. 138, pp. 3–34, February 1995.

[3] R. Alur, T. Dang, J. Esposito, R. Fierro, Y. Hur, F. Ivancic, V. Kumar, I. Lee, P. Mishra, G. Pappas, , and O. Sokolsky, "Hierarchical hybrid modeling of embedded systems," in *EMSOFT 2001: Embedded Software, First International Workshop*, (T. A. Henzinger and C. M. Kirsch, eds.), (Tahoe City, CA, USA), pp. 14–31, Springer-Verlag, 2001.

[4] R. Alur, T. Dang, J. Esposito, Y. Hur, F. Ivancic, V. Kumar, I. Lee, P. Mishra, G. J. Pappas, and O. Sokolsky, "Hierarchical modeling and analysis of embedded systems," *Proceedings of the IEEE*, vol. 91, pp. 11–28, January 2003.

[5] R. Alur, A. Das, J. Esposito, R. Fierro, Y. Hur, G. Grudic, V. Kumar, I. Lee, J. P. Ostrowski, G. Pappas, J. Southall, J. Spletzer, and C. J. Taylor, "A framework and architecture for multirobot coordination," in *Proc. ISER00, 7^{th} Intl. Symp. on Experimental Robotics*, pp. 289–299, 2000.

[6] R. Alur and R. Grosu, "Modular refinement of hierarchic reactive machines," in *Principles of Programming Languages*, pp. 390–402, ACM Press, 2000.

[7] R. Alur, R. Grosu, Y. Hur, V. Kumar, and I. Lee, "Modular refinement of hierarchic reactive machines," in *Proc. of the 27th Annual ACM Symp. on Principles of Programming Languages*, pp. 390–402, 2000.

[8] R. Alur, R. Grosu, Y. Hur, V. Kumar, and I. Lee, "Modular specification of hybrid systems in Charon," in *Proc. of the Third Intl. Work. on Hybrid*

Systems: Computation and Control, (N. A. Lynch and B. H. Krogh, eds.), pp. 6–19, Springer-Verlag, 2000.

[9] R. Alur and T. A. Henzinger, "Modularity for timed and hybrid systems," in *CONCUR '97: Eight International Conference on Concurrency Theory*, pp. 74–88, Springer-Verlag, 1997.

[10] R. Alur, T. A. Henzinger, and P. H. Ho, "Automatic symbolic verification of embedded systems," in *Proc. of the 14th Annual Real-time Systems Symp.*, pp. 2–11, 1993.

[11] R. Alur, T. A. Henzinger, and P. H. Ho, "Automatic symbolic verification of embedded systems," *IEEE Transactions on Software Engineering*, vol. 22, pp. 181–201, March 1996.

[12] R. Alur, T. A. Henzinger, and E. D. Sontag, eds., *Hybrid Systems III: Verification and control, Proceedings of the DIMACS/SYCON Workshop, October 22-25, 1995, Rutgers University, New Brunswick, NJ, USA*, Springer, 1996.

[13] R. Alur, S. Kannan, and S. La Torre, "Polyhedral flows in hybrid automata," in *HSCC '99: Proceedings of the Second International Workshop on Hybrid Systems*, (F. W. Vaandrager and J. H. van Schuppen, eds.), (London, UK), pp. 5–18, Springer-Verlag, 1999.

[14] R. Alur and G. J. Pappas, eds., *Hybrid Systems: Computation and Control, 7th International Workshop, HSCC 2004, Philadelphia, PA, USA, March 25–27., 2002*, Springer, 2004.

[15] A. Angermann, M. Beuschel, M. Rau, and U. Wohlfarth, *MATLAB, Simulink, Stateflow: grundlagen, toolboxen, beispiele (MATLAB, Simulink, Stateflow: fundamentals, toolboxes, examples).* Oldenbourg-Verlag, 2003.

[16] M. Antoniotti and A. Gollu, "SHIFT and Smart AHS: A language for hybrid system engineering modeling and simulation," in *Proceedings of the Conference on Domain-Specific Languages*, (Santa Barbara, CA, USA), Oct. 15-17 1997.

[17] P. J. Antsaklis, "Special issue on hybrid systems: Theory and applications," *Proc. of the IEEE*, vol. 88, July 2000.

[18] P. J. Antsaklis, W. Kohn, A. Nerode, and S. Sastry, eds., *Hybrid Systems II*, Springer, 1995.

[19] P. J. Antsaklis, W. Kohn, A. Nerode, and S. Sastry, eds., *Hybrid Systems IV*, Springer, 1997.

[20] P. J. Antsaklis and A. Nerode, "Hybrid control systems, special issue," *IEEE Transactions on Automatic Control*, vol. 43, April 1998.

[21] E. Asarin, O. Bournez, T. Dang, , and O. Maler, "Approximate reachability analysis of piecewise linear dynamical systems," in *HSCC 00: Hybrid Systems—Computation and Control*, (B. Krogh and N. Lynch, eds.), pp. 20–31, Springer-Verlag, 2000.

[22] E. Asarin, O. Bournez, T. Dang, O. Maler, and A. Pnueli, "Effective synthesis of switching controllers for linear systems," *Proceedings of the IEEE*, vol. 88, pp. 1011–1025, July 2000.

[23] E. Asarin, T. Dang, and O. Maler, "d/dt: A verification tool for hybrid systems," in *Proc. of the 40th IEEE Conf. on Decision and Control*, pp. 2893–2898, 2001.

[24] E. Asarin, T. Dang, and O. Maler, "The d/dt tool for verification of hybrid systems," in *Proc. of the 14th Intl. Conf. on Computer-Aided Verification*, pp. 365–370, 2002.

[25] R. Bagnara, E. Ricci, E. Zaffanella, and P. M. Hill, "Possibly not closed convex polyhedra and the Parma Polyhedra Library," 2002.

[26] F. Balarin, H. Hsieh, L. Lavagno, C. Passerone, A. Sangiovanni-Vincentelli, and Y. Watanabe, "Metropolis: An integrated design environment for electronic system design," *IEEE Micro*, vol. 36, pp. 45–52, April 2003.

[27] F. Balarin, L. Lavagno, C. Passerone, A. Sangiovanni-Vincentelli, M. Sgroi, and Y. Watanabe, "Modeling and designing heterogeneous systems," Tech. Rep. 2002/01, Cadence Berkeley Laboratories, January 2002.

[28] A. Balluchi, A. Casagrande, P. Collins, A. Ferrari, T. Villa, and A. L. Sangiovanni-Vincentelli, "Ariadne: A framework for reachability analysis of hybrid automata," in *17th International Symposium on Mathematical Theory of Networks and Systems (MTNS)*, July 2006.

[29] A. Bemporad, F. Borrelli, and M. Morari, "Piecewise linear optimal controllers for hybrid systems," in *American Control Conference*, (Chicago, USA), pp. 1190–1194, 2000.

[30] A. Bemporad, F. Borrelli, and M. Morari, "On the optimal control law for linear discrete time hybrid systems," in *International Workshop on Hybrid Systems: Computation and Control*, (Stanford, California, USA), pp. 105–119, 2002.

[31] A. Bemporad and M. Morari, "Control of systems integrating logic, dynamics, and constraints," *Automatica*, vol. 35, pp. 407–427, March 1999.

[32] J. Bengtsson, K. G. Larsen, F. Larsson, P. Pettersson, and W. Yi, "UPPAAL—a tool suite for automatic verification of real-time systems," in *Hybrid Systems III*, pp. 208–219, Springer-Verlag, 1996.

[33] A. Benveniste, "Compositional and uniform modelling of hybrid systems," *IEEE Transactions on Automatic Control*, vol. 43, pp. 579–584, April 1998.

[34] A. Benveniste and P. Le Guernic, "Hybrid dynamical systems theory and the signal language," *IEEE Transactions on Automatic Control*, vol. 35, pp. 535–546, May 1990.

[35] V. Bertin, E. Closse, M. Poize, J. Pulou, J. Sifakis, P. Venier, D. Weil, and S. Yovine, "Taxis = Esterel + Kronos: A tool for verifying real-time properties of embedded systems," in *Proc. of the 40th IEEE Conf. on Decision and Control*, Springer-Verlag, 2001.

[36] D. Bobrow, G. Kiczales, and J. Rivieres, *The Art of the Metaobject Protocol*. MIT Press, 1991.

[37] G. Bobrow, L. Demichiel, R. Gabriel, S. Keene, G. Kiczales, and D. Moon, "Common Lisp object specification," *Lisp and Symbolic Computation*, vol. 1, January 1989.

[38] G. Booch, I. Jacobson, and J. Rumbaugh, *Unified Modeling Language User Guide*. Addison Wesley, 1997.

[39] O. Botchkarev and S. Tripakis, "Verification of hybrid systems with linear differential inclusions using ellipsoidal approximations," in *HSCC*, pp. 73–88, 2000.

[40] O. Bournez, O. Maler, and A. Pnueli, "Orthogonal polyhedra: representation and computation," in *HSCC '99: Proceedings of the Second International Workshop on Hybrid Systems*, (F. W. Vaandrager and J. H. van Schuppen, eds.), (London, UK), pp. 46–60, Springer-Verlag, 1999.

[41] M. Bozga, C. Daws, O. Maler, A. Olivero, S. Tripakis, and S. Yovine, "Kronos: A model-checking tool for real-time systems," in *Proc. of the 10th Intl. Conf. on Computer-Aided Verification*, (A. J. Hu and M. Y. Vardi, eds.), pp. 546–550, Springer-Verlag, 1998.

[42] J. R. Burch, E. M. Clarke, K. L. McMillan, and D. L. Dill, "Symbolic model checking: 10^{20} states and beyond," *Information and Computation*, vol. 98, no. 2, pp. 142–170, 1992.

[43] L. P. Carloni, M. Di Benedetto, R. Passerone, A. Pinto, and A. Sangiovanni-Vincentelli, "Modeling techniques, programming languages and design toolsets for hybrid systems," Tech. Rep., IST - Columbus Project, 2004. available at `www.columbus.gr/documents/public/WPHS/Columbus_DHS4_0.2.pdf`.

[44] A. Casagrande, A. Balluchi, L. Benvenuti, A. Policriti, T. Villa, and A. L. Sangiovanni-Vincentelli, "Improving reachability analysis of hybrid automata for engine control," in *Proc. of CDC 2004, 44th IEEE Conference on Decision and Control*, (Atlantis, Paradise Island, Bahamas), pp. 2322–2327, December 2004.

[45] P. Caspi, A. Curic, A. Maignan, C. Sofronis, and S. Tripakis, "Translating discrete-time Simulink to Lustre," in *Proc. of the Third Intl. Conf. on Embedded Software (EMSOFT). Philadelphia, PA*, (R. Alur and I. Lee, eds.), (Berlin), pp. 84–99, Springer Verlag, October 2003.

[46] A. Chutinan, *Hybrid system verification using discrete model approximations*. PhD thesis, Carnegie Mellon University, 1999.

[47] A. Chutinan and B. H. Krogh, "Computing polyhedral approximations to flow pipes for dynamic systems," in *37th IEEE Conf. on Decision and Control: Session on Synthesis and Verification of Hybrid Control Laws (TM-01)*, 1998.

[48] E. Clarke and E. Emerson, "Design and synthesis of synchronization skeletons using branching-time temporal logic," in *Workshop on Logic of Programs*, Springer-Verlag, 1981.

[49] E. M. Clarke, O. Grumberg, and D. A. Peled, *Model Checking*. MIT Press, 2000.

[50] E. Closse, M. Poize, J. Pulou, J. Sifakis, P. Venier, D. Weil, and S. Yovine, "Taxys: A tool for the development and verification real-time embedded systems," in *Proc. of the 13th Intl. Conf. on Computer-Aided Verification*, Springer-Verlag, 2001.

[51] P. Collins, "Continuity and computability of reachable sets," *Theoretical Computer Science*, vol. 341, pp. 162–195, 2005.

[52] J. B. Dabney and T. L. Harman, *Mastering Simulink*. Prentice Hall, 2003.

[53] T. Dang, *Verification and synthesis of hybrid systems*. PhD thesis, INPG, 2000.

[54] T. Dang and O. Maler, "Reachability analysis via face lifting," in *HSCC 98: Hybrid Systems—Computation and Control*, (T. A. Henzinger and S. Sastry, eds.), pp. 96–109, Springer-Verlag, 1998.

[55] A. David, G. Behrmann, K. G. Larsen, and W. Yi, "A Tool architecture for the next generation of UPPAAL," in *Proceedings of UNU/IIST 10th Anniversary Colloquium: Formal Methods at the Crossroads: from Panacea to Foundational Support*, (B. K. Aichernig and T. Maibaum, eds.), pp. 208–219, Springer-Verlag, 2002.

[56] J. Davis, M. Goel, C. Hylands, B. Kienhuis, E. A. Lee, J. Liu, X. Liu, L. Muliadi, S. Neuendorffer, J. Reekie, N. Smyth, J. Tsay, and Y. Xiong, "Overview of the Ptolemy project," Tech. Rep. UCB/ERL M99/37, Univ. of California at Berkeley, 1999.

[57] C. Daws, A. Olivero, S. Tripakis, and S. Yovine, "The tool Kronos," in *Hybrid Systems III*, pp. 208–219, Springer-Verlag, 1996.

[58] A. Deshpande, D. Godbole, A. Gollu, L. Semenzato, R. Sengupta, D. Swaroop, and P. Varaiya, "Automated highway system tool interface format," Tech. Rep., California PATH Technical Report, January 1996.

[59] A. Deshpande, D. Godbole, A. Gollu, and P. Varaiya, "Design and evaluation tools for automated highway systems," in *Hybrid Systems III*, Springer-Verlag, 1996.

[60] A. Deshpande, A. Gollu, and P. Varaiya, "Shift: A formalism and a programming language for dynamic networks of hybrid automata," in *Hybrid Systems IV*, pp. 113–134, Springer-Verlag, 1997.

[61] A. Deshpande, A. Gollu, and P. Varaiya, "The SHIFT programming language for dynamic networks of hybrid automata," *IEEE Transactions on Automatic Control*, vol. 43, pp. 584–7, April 1998.

[62] A. Deshpande and P. Varaiya, "Viable control of hybrid systems," in *Hybrid Systems II*, Springer-Verlag, 1995.

[63] M. D. Di Benedetto and A. L. Sangiovanni-Vincentelli, eds., *Hybrid Systems: Computation and Control, 4th International Workshop, HSCC 2001, Rome, Italy, March 28-30, 2001*, Springer, 2001.

[64] R. Djenidi, C. Lavarenne, R. Nikoukhah, Y. Sorel, and S. Steer, "From hybrid simulation to real-time implementation," in *ESS'99 11th European Simulation Symposium and Exhibition*, pp. 74–78, October 1999.

[65] S. Edwards, L. Lavagno, E. Lee, and A. Sangiovanni-Vincentelli, "Design of embedded systems: Formal models, validation and synthesis," *Proc. of the IEEE*, vol. 85, pp. 366–390, March 1997.

[66] J. Eker, J. W. Janneck, E. A. Lee, J. Liu, J. Ludwig, S. Neuendorffer, S. Sachs, and Y. Xiong, "Taming heterogeneity—The Ptolemy approach," *Proceedings of the IEEE*, vol. 91, pp. 127–144, January 2003.

[67] H. Elmqvist, "Dymola - user's manual," Tech. Rep., DynaSim AB, Research Park Ideon, Lund, Sweden, 1993.

[68] F. Eskafi, D. Khorramabadi, and P. Varaiya, "Design and evaluation tools for automated highway systems," *Transpn. Res. - C*, vol. 3, no. 1, pp. 1–17, 1995.

[69] G. Frehse, "PHAVer: Algorithmic verification of hybrid systems past HyTech," in *HSCC*, pp. 258–273, 2005.

[70] P. Fritzson, *Principles of Object-oriented Modeling and Simulation with Modelica 2.1.* J. Wiley & Sons, 2004.

[71] P. Fritzson and V. Engelson, "Modelica - a unified object-oriented language for system modeling and simulation," in *ECCOP '98: Proc. of the 12th Eur. Conf. on Object-Oriented Programming*, (London, UK), pp. 67–90, Springer-Verlag, 1998.

[72] A. Girard and G. J. Pappas, "Approximate bisimulations for constrained linear systems," in *Proc. of the 44th IEEE Conf. on Decision and Control*, December 2005.

[73] A. Girard and G. J. Pappas, "Approximate bisimulations for nonlinear dynamical systems," in *Proc. of the 44th IEEE Conf. on Decision and Control*, December 2005.

[74] D. Godbole, J. Lygeros, E. Singh, A. Deshpande, and E. Lindsey, "Design and verification of communication protocols for degraded modes of operation of AHS," in *Conference on Decision and Control*, IEEE, 1995.

[75] A. Gollu, *Object management systems*. PhD thesis, UC Berkeley, 1995.

[76] A. Gollu and P. Varaiya, "Smart AHS: A simulation framework for automated vehicles and highway systems," *Mathematical and Computer Modeling*, vol. 27, pp. 103–28, May-June 1998.

[77] C. Gomez, *Engineering and Scientific Computing with Scilab*. Birkhauser Verlag, 1999.

[78] T. Grandpierre, C. Lavarenne, and Y. Sorel, "Optimized rapid prototyping for real-time embedded heterogeneous multiprocessors," in *Proc. of CODES'99*, pp. 74–78, 1999.

[79] T. Grandpierre and Y. Sorel, "From algorithm and architecture specifications to automatic generation of distributed real-time executives: A seamless flow of graphs transformations," in *MEMOCODE2003, Formal Methods and Models for Codesign Conference*, p. 123, June 2003.

[80] R. L. Grossman, A. Nerode, A. P. Ravn, and H. Rischel, eds., *Hybrid systems*, Springer, 1993.

[81] J. Haddon, D. Godbole, A. Deshpande, and J. Lygeros, "Verification of hybrid systems: monotonicity in the AHS control system," in *Proceedings of the DIMACS/SYCON workshop on Hybrid systems III : verification and control*, (Secaucus, NJ, USA), pp. 161–172, Springer-Verlag New York, Inc., 1996.

[82] N. Halbwachs, *Synchronous Programming of Reactive Systems*. Kluwer Academic Publishers, 1993.

[83] N. Halbwachs, P. Raymond, and Y.-E. Proy, "Verification of linear hybrid systems by means of convex approximation," in *SAS 94: Static Analysis Symposium*, (B. LeCharlier, ed.), pp. 233–237, Springer-Verlag, 1994.

[84] D. Harel, "Statecharts: A visual formalism for complex systems," *Science of Computer Programming*, vol. 8, p. 231:274, July 1987.

[85] W. P. M. H. Heemels, B. De Schutter, and A. Bemporad, "Equivalence of hybrid dynamical models," *Automatica*, vol. 37, pp. 1085–1091, July 2001.

[86] T. Henzinger and P. H. Ho, "A note on abstract-interpretation strategies for hybrid automata," in *Hybrid Systems II*, (P. Antsaklis, A. Nerode, W. Kohn, and S. Sastry, eds.), pp. 252–264, Springer-Verlag, 1995.

[87] T. Henzinger and P. H. Ho, "HyTeCh: The Cornell hybrid technology tool," in *Hybrid Systems II*, (P. Antsaklis, A. Nerode, W. Kohn, and S. Sastry, eds.), pp. 265–293, Springer-Verlag, 1995.

[88] T. Henzinger, P. H. Ho, and H. Wong-Toi, "A user guide to HyTeCH," in *TACAS 95: Tools and Algorithms for the Construction and Analysis of Systems*, (E. Brinksma, W. Cleaveland, K. Larsen, T. Margaria, and B. Steffen, eds.), pp. 41–71, Springer-Verlag, 1995.

[89] T. Henzinger, P. H. Ho, and H. Wong-Toi, "Algorithmic analysis of nonlinear hybrid systems," *IEEE Transactions on Automatic Control*, vol. 43, pp. 540–554, April 1998.

[90] T. Henzinger, X. Nicollin, J. Sifakis, and S. Yovine, "Symbolic model checking for real-time systems," *Information and Computation*, vol. 111, no. 2, pp. 193–244, 1994.

[91] T. Henzinger, J. Preussig, and H. Wong-Toi, "Some lessons from the HyTech experience," in *Proc. of the 40th IEEE Conf. on Decision and Control*, pp. 2886–2892, 2001.

[92] T. A. Henzinger, "The theory of hybrid automata," in *Logic in Computer Science*, pp. 278–292, IEEE Computer Society Press, 1996.

[93] T. A. Henzinger, "Masaccio: A formal model for embedded components," in *TCS 00: Theoretical Computer Science*, (J. van Leeuwen, O. Watanabe, M. Hagiya, P. D. Mosses, and T. Ito, eds.), pp. 549–563, Springer-Verlag, 2000.

[94] T. A. Henzinger and P.-H. Ho, "Model checking strategies for linear hybrid systems," in *Proc. Workshop on Hybrid Systems and Autonomous Control*, (Ithaca, NY), 1994.

[95] T. A. Henzinger, P. H. Ho, and H. Wong-Toi, "HyTech: A model checker for hybrid systems," *International Journal on Software Tools for Technology Transfer*, vol. 1, no. 1–2, pp. 110–122, 1997.

[96] T. A. Henzinger, P. W. Kopke, A. Puri, and P. Varaiya, "What's decidable about hybrid automata?," in *Proc. 27th Annual ACM Symp. on Theory of Computing (STOC)*, pp. 373–382, 1995.

[97] T. A. Henzinger, M. Minea, and V. Prabhu, "Assume-guarantee reasoning for hierarchical hybrid systems," in *HSCC 01: Hybrid Systems—Computation and Control*, (M. di Benedetto and A. Sangiovanni-Vincentelli, eds.), pp. 275–290, Springer-Verlag, 2001.

[98] T. A. Henzinger and S. Sastry, eds., *Hybrid Systems: Computation and Control, First International Workshop, HSCC'98, Berkeley, California, USA, April 13-15, 1998*, Springer, 1998.

[99] C. A. R. Hoare, *Communicating Sequential Processes*. Prentice-Hall, 1985.

[100] G. J. Holzmann, "The Model Checker SPIN," *Software Engineering*, vol. 23, no. 5, pp. 279–295, 1997.

[101] G. P. Hong and T. G. Kim, "The DEVS formalism: A framework for logical analysis and performance," in *Fifth Annual Conference on AI, Simulation and Planning in High Autonomy Systems*, (Gainesville, Florida), pp. 170–278, 1994.

[102] A. Hsu, F. Eskafi, S. Sachs, and P. Varaiya, "Protocol design for and automated highway system," in *Discrete Event Dynamic Systems: Theory and Applications 2*, pp. 183–206, 1993.

[103] C. Hylands, E. A. Lee, J. Liu, X. Liu, S. Neuendorffer, and H. Zheng, "HyVisual: A hybrid system visual modeler," Tech. Rep. UCB/ERL M03/1, UC Berkeley, 2003. available at `http://ptolemy.eecs.berkeley.edu/hyvisual/`.

[104] K. Inan and P. Varaiya, "Finitely recursive process models for discrete-event systems," *IEEE Transactions on Automatic Control*, vol. 33, pp. 626–639, July 1988.

[105] *Specification and description language SDL*. 1988.

[106] *Estelle - A formal description technique based on extended state transition model*.

[107] J. Jang, R. Teo, and C. Tomlin, "Embedded software design for the Stanford DragonFly UAV," Tech. Rep., Stanford Univ., Dept. of Aeronautics and Astronautics, 2002.

[108] J. S. Jang and C. Tomlin, "Design and implementation of a low cost, hierarchical and modular avionics architecture for the DragonFly UAVs," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, (Monterey), August 2002.

[109] K. Keutzer, S. Malik, A. R. Newton, J. Rabaey, and A. Sangiovanni-Vincentelli, "System level design: orthogonalization of concerns and platform-based design," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 19, pp. 1523–1543, December 2000.

[110] A. B. Kurzhanski and P. Valyi, *Ellipsoidal Calculus for Estimation and Control*. Birkhaeuser, Boston, 1997.

[111] A. B. Kurzhanski and P. Varaiya, "Ellipsoidal techniques for reachability analysis," in *HSCC 00: Hybrid Systems—Computation and Control*, (B. Krogh and N. Lynch, eds.), pp. 202–214, Springer-Verlag, 2000.

[112] A. B. Kurzhanski and P. Varaiya, "Ellipsoidal techniques for reachability analysis: Internal approximation," *Systems and Control Letters*, vol. 41, pp. 201–211, 2000.

[113] A. B. Kurzhanski and P. Varaiya, "On ellipsoidal techniques for reachability analysis," *Optimization Methods and Software*, vol. 17, pp. 177–237, 2000.

[114] A. B. Kurzhanski and P. Varaiya, "On ellipsoidal techniques for reachability analysis - Part I: external approximations," *Optimization Methods and Software*, vol. 17, no. 2, pp. 177–206, 2002.

[115] A. B. Kurzhanski and P. Varaiya, "On ellipsoidal techniques for reachability analysis - Part II: internal approximations based-valued constraints," *Optimization Methods and Software*, vol. 17, no. 2, pp. 207–337, 2002.

[116] A. B. Kurzhanski and P. Varaiya, "Reachability analysis for uncertain systems - the ellipsoidal technique," *Dynamics of Continuous, Discrete and Impulsive Systems*, vol. 9, no. 3, pp. 347–367, 2002.

[117] A. B. Kurzhanski and P. Varaiya, "Ellipsoidal techniques for hybrid dynamics: the reachability problem," in *New Directions and Applications in Control Theory*, (A. Lindquist, W. Dayawensa, and Y. Zhou, eds.), pp. 193–206, Springer-Verlag, 2005.

[118] A. B. Kurzhanski and P. Varaiya, "On verification of controlled hybrid dynamics through ellipsoidal techniques," in *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference 2005, Seville, Spain*, pp. 4682–4686, December 2005.

[119] A. A. Kurzhanskiy and P. Varaiya, "Ellipsoidal techniques for reachability analysis of discrete-time linear systems," *IEEE Transactions on Automatic Control*, Submitted for Publication, June 2005.

[120] A. A. Kurzhanskiy and P. Varaiya, *Ellipsoidal toolbox - technical report*. University of California, Berkeley, http://www.eecs.berkeley.edu/~akurzhan/ellipsoids, 2006.

[121] M. Kvasnica, P. Grieder, M. Baotic, and M. Morari, "Multi-parametric toolbox (MPT).," in *HSCC*, pp. 448–462, 2004.

[122] G. Lafferriere, G. J. Pappas, and S. Yovine, "A new class of decidable hybrid systems," in *HSCC '99: Proceedings of the Second International Workshop on Hybrid Systems*, (F. W. Vaandrager and J. H. van Schuppen, eds.), (London, UK), pp. 137–151, Springer-Verlag, 1999.

[123] L. Lamport, "Proving the correctness of multiprocess programs," *IEEE Transactions on Software Engineering*, vol. 2, pp. 125–143, February 1977.

[124] L. Lavagno, J. Moondanos, T. Meyerowitz, and Y. Watanabe, "Modeling of architectural resources in Metropolis," Internal Document, Cadence, 2002.

[125] E. A. Lee and S. Neuendorffer, "Concurrent models of computation for embedded software," *IEE Proceedings*, vol. 153, pp. 239–250, March 2005.

[126] E. A. Lee and A. L. Sangiovanni-Vincentelli, "A framework for comparing models of computation," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 17, pp. 1217–1229, December 1998.

[127] E. A. Lee and Y. Xiong, "System-level types for component-based design," in *Embedded Software. Proceeding of the First International Workshop, EMSOFT 2001. Tahoe City, CA*, (T. A. Henzinger and C. M. Kirsch, eds.), (Berlin), pp. 237–253, Springer Verlag, October 2001.

[128] E. A. Lee and H. Zheng, "Operational semantics of hybrid systems," in *HSCC*, pp. 25–53, 2005.

[129] J. Lygeros, C. Tomlin, and S. Sastry, "Controllers for reachability specifications for hybrid systems," in *Automatica, Special Issue on Hybrid Systems*, 1999.

[130] N. Lynch, R. Segala, F. Vaandrager, and H. Weinberg, "Hybrid I/O automata," in *Hybrid Systems III: Verification and Control*, pp. 496–510, Springer-Verlag, 1996.

[131] N. A. Lynch and B. H. Krogh, eds., *Hybrid Systems: Computation and Control, Third International Workshop, HSCC 2000, Pittsburgh, PA, USA, March 23-25, 2000*, Springer, 2000.

[132] O. Maler, ed., *Hybrid and Real-Time Systems, International Workshop. HART'97, Grenoble, France, March 26-28, 1997, Proceedings*, Springer, 1997.

[133] O. Maler, Z. Manna, and A. Pnueli, "From timed to hybrid systems," in *Real-Time: Theory in Practice, REX Workshop*, pp. 447–484, Springer-Verlag, 1991.

[134] O. Maler and A. Pnueli, eds., *Hybrid Systems: Computation and Control, 6th International Workshop, HSCC 2003 Prague, Czech Republic, April 3–5, 2003*, Springer, 2003.

[135] Matisse, “Available at `http://wiki.grasp.upenn.edu/~graspdoc/hst/`,”.

[136] K. L. McMillan, *Symbolic Model Checking*. Kluwer Academic Publishers, 1993.

[137] R. Milner, *A Calculus of Communicating Systems. Lecture Notes in Computer Science*, Springer-Verlag, 1980.

[138] MoBIES Group, “HSIF syntax (version 3),” Internal Document, Vanderbilt University, October 22, 2002.

[139] Modelica Association, “Modelica - A unified object-oriented language for physical systems modeling, language specification,” December 2000.

[140] M. Morari and L. Thiele, eds., *Hybrid Systems: Computation and Control, 8th International Workshop, HSCC 2005, Zurich, Switzerland, March 9–11, 2005*, Springer, 2005.

[141] P. Mosterman, “On the normal component of centralized frictionless collision sequences,” *ASME Journal of Applied Mechanics*, 2005.

[142] S. Neema, “Analysis of Matlab Simulink and Stateflow data model,” Tech. Rep. ISIS 01-204, Vanderbilt University, Nashville, TN, March 2001.

[143] R. Nikoukhah and S. Steer, “SCICOS - a dynamic system builder and simulator user’s guide - version 1.0,” Tech. Rep. 0207, INRIA, Rocquencourt, France, June 1997.

[144] A. Pinto, L. P. Carloni, R. Passerone, and A. L. Sangiovanni-Vincentelli, “Interchange formats for hybrid systems: Abstract semantics,” in *Proceedings of the The 9th International Workshop on Hybrid Systems: Computation and Control (HSCC 2006)*, (J. P. Hespanha and A. Tiwari, eds.), (Santa Barbara, California), pp. 491–506, Springer-Verlag, March 2006.

[145] A. Pinto, A. L. Sangiovanni-Vincentelli, L. P. Carloni, and R. Passerone, “Interchange formats for hybrid systems: Review and proposal,” in *Proceedings of the The 8th International Workshop on Hybrid Systems : Computation and Control (HSCC 2005)*, (M. Morari, L. Thiele, and F. Rossi, eds.), pp. 526–541, March 2005.

[146] A. Puri and P. Varaiya, “Driving safely in smart cars,” in *American Control Conference*, pp. 3597–3599, 1995.

[147] S. Ratschan and Z. She, “Safety verification of hybrid systems by constraint propagation based abstracition refinement,” in *HSCC*, pp. 573–589, 2005.

[148] M. Rivoire and J. L. Ferrier, *MATLAB Simulink Stateflow avec des Exercices d’Automaticue Rsolus (MATLAB Simulink Stateflow with Solved Exercises in Automatic Control)*. Editions TECHNIP, 2001.

[149] A. L. Sangiovanni-Vincentelli, “Defining platform-based design,” in *EEDesign. Available at www.eedesign.com/story/OEG20020204S0062)*, February 2002.

[150] L. Semenzato, A. Deshpande, and A. Gollu, “Shift reference manual,” Tech. Rep., California PATH, June 1996.

[151] B. I. Silva, K. Richeson, B. Krogh, and A. Chutinan, “Modeling and verifying hybrid dynamic systems using CheckMate,” in *Proceedings of 4th International Conference on Automation of Mixed Processes*, pp. 323–328, September 2000.

[152] B. I. Silva, O. Stursberg, B. H. Krogh, and S. Engell, "An assessment of the current status of algorithmic approaches to the verification of hybrid systems," in *Proc. of the 40th IEEE Conf. on Decision and Control*, pp. 2867 – 2874, 2001.

[153] T. Simsek, "SHIFT tutorial: A first course for SHIFT programmers," Tech. Rep., UC Berkeley, 1999.

[154] T. Simsek, "The λ-SHIFT specification language for dynamic networks of hybrid automata," Tech. Rep., UC Berkeley, 2000.

[155] E. D. Sontag, "Nonlinear regulation: the piecewise linear approach," *IEEE Transactions on Automatic Control*, vol. 26, pp. 346–357, April 1981.

[156] Y. Sorel, "Massively parallel computing systems with real time constraints - the "Algorithm Architecture Adequation" methodology," in *Massively Parallel Computing Systems Conference*, pp. 44–54, 1994.

[157] G. Steele, *Common Lisp: The Language*. Digital Press, Second edition ed., 1990.

[158] D. Stipanovic, G. Inalhan, and C. Tomlin, "Decentralized overlapping control of a formation of unmanned aerial vehicles," in *Proceedings of the 41st IEEE Conference on Decision and Control*, (Las Vegas, NV), December 2002.

[159] O. Stursberg, S. Kowalewski, J. Preussig, and H. Treseler, "Block-diagram based modelling and analysis of hybrid processes under discrete control," *Journal Europeen des Systemes Automatises (JESA)*, vol. 32, no. 9-10, pp. 1097–1118, 1998.

[160] The Metropolis Project Team, "The Metropolis meta model version 0.4," Tech. Rep. UCB/ERL M04/38, University of California, Berkeley, September 2004.

[161] The University of Pennsylvania MoBIES Group, "HSIF semantics (version 3, synchronous edition)," Internal Document, The University of Pennsylvania, August 22, 2002.

[162] M. M. Tiller, *Introduction to physical modeling with Modelica*. Kluwer Academic Publishers, 2001.

[163] C. Tomlin and M. R. Greenstreet, eds., *Hybrid Systems: Computation and Control, 5th International Workshop, HSCC 2002, Stanford, CA, USA, March 25-27, 2002*, Springer, 2002.

[164] F. D. Torrisi and A. Bemporad, "Discrete-time hybrid modeling and verification," in *Proc. of the 40th IEEE Conf. on Decision and Control*, pp. 2899 – 2904, 2001.

[165] F. D. Torrisi and A. Bemporad, "HYSDEL - a tool for generating computational hybrid models for analysis and synthesis problems," *IEEE Transactions on Control Systems Technology*, vol. 12, pp. 235–249, March 2004.

[166] F. D. Torrisi, A. Bemporad, G. Bertini, P. Hertach, D. Jost, and D. Mignone, "Hysdel 2.0.5 - user manual," Tech. Rep., ETH Zurich, 2002.

[167] A. C. Uselton and S. A. Smolka, "A compositional semantics for Statecharts using labeled transition systems," in *Concurrency Theory*, pp. 2–17, Springer-Verlag, 1994.

[168] F. W. Vaandrager and J. H. van Schuppen, eds., *Hybrid Systems: Computation and Control, Second International Workshop, HSCC'99, Bergen Dal, The Netherlands, 1999*, Springer, 1999.

[169] P. Varaiya, "Smart cars on smart roads: problems of control," *IEEE Transactions on Automatic Control*, vol. 38, pp. 195–207, February 1993.

[170] P. Varaiya, "Reach set computation using optimal control," in *KIT Workshop on Verification of Hybrid Systems, Grenoble, France*, October 1998.

[171] S. M. Veres, *User's manual - reference of the geometric bounding toolbox - version 7.3*. SysBrain Ltd, Southampton, United Kingdom, url = http://sysbrain.com/gbt/, March 2004.

[172] K. Weihrauch, *Computable Analysis - An Introduction*. *Texts in Theoretical Computer Science*, Berlin: Springer-Verlag, 2000.

[173] S. Wolfram, *The* MATHEMATICA *book, fifth edition*. Wolfram Media, 2003.

[174] XML, "see `http://www.w3.org/XML/`,".

[175] B. Zeigler, *Multifaceted Modeling and Discrete Event Simulation*. Academic Press, London, 1984.