VERIFICATION OF HYBRID SYSTEMS

Claire J. Tomlin and Alexandre M. Bayen
Department of Aeronautics and Astronautics, Stanford University, Stanford CA 94305-4035, USA

Ian Mitchell
Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA 94709, USA

Meeko M. K. Oishi
Department of Mechanical Engineering, Stanford University, Stanford CA 94305-4035, USA

Keywords: Hybrid Systems; Verification; Reachability; Safety.

Contents

Summary

Hybrid system theory lies at the intersection of the fields of engineering control theory and computer science verification. It is defined as the modeling, analysis, and control, of systems which involve the interaction of both discrete state systems, represented by finite automata, and continuous state dynamics, represented by differential equations. The embedded autopilot of a modern commercial jet is a prime example of a hybrid system: the autopilot modes correspond to the application of different control laws, and the logic of mode switching is determined by the continuous state dynamics of the aircraft, as well as through interaction with the pilot. To understand the behavior of hybrid systems, to simulate, and to control these systems, theoretical advances, analyses, and numerical tools are needed. In this chapter, a general model for a hybrid

system is first presented, along with an overview of methods for verifying continuous and hybrid systems. Then, a particular verification technique for hybrid systems, based on two-person zero-sum game theory for automata and continuous dynamical systems, is described. A numerical implementation of this technique using level set methods, and its use in the design and analysis of aircraft collision avoidance protocols, and in verification of autopilot logic, is demonstrated.

1. Introduction

The field of formal verification in computer science has achieved great success in the analysis of large scale discrete systems: using temporal logic to express discrete sequences of events, such as *Component A will request data until Component B sends data*, researchers in verification have uncovered design flaws in such safety critical systems as microprocessors which control aircraft cockpit displays and design standards for a military hardware bus. Discrete analysis, however, is not rich enough to verify systems which evolve according to both continuous dynamics and discrete events. *Embedded systems*, or physical systems controlled by a discrete logic, such as the current autopilot logic for automatically controlling an aircraft, or a future automated protocol for controlling an aircraft in the presence of other aircraft, are prime examples of systems in which event sequences are *determined* by continuous state dynamics. These systems use discrete logic in control because discrete abstractions make it easier to manage system complexity and discrete representations more naturally accommodate linguistic and qualitative information in controller design. While engineering control theory has successfully designed tools to verify and control continuous state systems, these tools do not extend to systems which mix continuous and discrete state, as in the examples above.

Hybrid systems theory lies at the intersection of the two traditionally distinct fields of computer science verification and engineering control theory. It is loosely defined as the modeling and analysis of systems which involve the interaction of both discrete event systems (represented by finite automata) and continuous time dynamics (represented by differential equations). The goals of this research are in the design of verification techniques for hybrid systems, the development of a software toolkit for efficient application of these techniques, and the use of these tools in the analysis and control of large scale systems. In this chapter, recent research results are summarized, and a detailed set of references is presented, on the development of tools for the verification of hybrid systems, and on the application of these tools to some interesting examples.

The problem that has received much recent research attention has been the verification of the *safety* property of hybrid systems, which seeks a mathematically precise answer to the question: *is a potentially unsafe configuration, or state, reachable from an initial configuration?* For discrete systems, this problem has a long history in mathematics and computer science and may be solved by posing the system dynamics as a discrete game; in the continuous domain, control problems of the safety type have been addressed in the context of differential games. For systems involving continuous dynamics, it is very difficult to compute and represent the set of states reachable from some initial set. In this chapter, recent solutions to the problem are presented, including a method, based on the level set techniques of Osher and Sethian, which determines an implicit

representation of the boundary of this *reachable set*. This method is based on the theorem, proved using two-person zero-sum game theory for continuous dynamical systems, that the viscosity solution of a particular Hamilton-Jacobi partial differential equation corresponds exactly to the boundary of the reachable set. In addition, it is shown that useful information for the control of such systems can be extracted from this boundary computation.

Much of the excitement in hybrid system research stems from the potential applications. With techniques such as the above, it is now possible to verify, and design safe, automated control schemes for low dimensional systems. Two interesting examples, one in the verification of protocols for aircraft collision avoidance, and one in the verification of mode switching logic in autopilots, are presented in this chapter. Other applications that have been studied in this framework are surveyed. This chapter concludes with a discussion of problem complexity.

2. Hybrid Model and Verification Methodology

2.1. Continuous, Discrete, and Hybrid Systems

Much of control theory is built around continuous-state models of system behavior. For example, the differential equation model given by

$$\dot{x} = f(x,u,d) \tag{1}$$

describes a system with *state* $x \in \mathbb{R}^n$ that evolves continuously in time according to the dynamical system $\dot{x} = f(\cdot,\cdot,\cdot)$, a function of x, $u \in \mathcal{U} \subseteq \mathbb{R}^{n_u}$, $d \in \mathcal{D} \subseteq \mathbb{R}^{n_d}$. In general, u is used to represent variables that can be controlled, called *control inputs*, and d represents *disturbance inputs*, which are variables that cannot be controlled, such as the actions of another system in the environment. The initial state $x(0) = x_0$ is assumed to belong to a set $X_0 \subseteq \mathbb{R}^n$ of allowable initial conditions. A *trajectory* of (1) is represented as $(x(t),u(t),d(t))$, such that $x(0) \in X_0$, and $x(t)$ satisfies the differential equation (1) for control and disturbance input trajectories $u(t)$ and $d(t)$. Sastry and Doyle are recommended as current references for continuous-state control systems.

Discrete-state models, such as finite automata, are also prevalent in control. The finite automaton given by

$$(Q, \Sigma, \text{Init}, R) \tag{2}$$

models a system which is a finite set of *discrete state variables* Q, a set of input variables $\Sigma = \Sigma_u \times \Sigma_d$ which is the Cartesian product of *control actions* $\sigma_u \in \Sigma_u$ and *disturbance actions* $\sigma_d \in \Sigma_d$, a set of *initial states* $\text{Init} \subseteq Q$, and a *transition relation* $R : Q \times \Sigma \to 2^Q$ which maps the state and input space to subsets of the state space (2^Q). A trajectory of (2) is a sequence of discrete states and inputs, which satisfies the transition relation at each step. The original work of Ramadge and Wonham brought the

use of discrete state systems to control, though parallels can be drawn between this work and that of Church, Büchi and Landweber who originally analyzed the von Neumann-Morgenstern discrete games. A comprehensive reference for modeling and control of discrete state systems is Cassandras and Lafortune.

Control algorithms are concerned with the design of a signal, either a continuous or discrete function of time, which when applied to the system causes the system state to exhibit desirable properties. These properties should hold despite possible disruptive action of the disturbance. A concrete example of a continuous-state control problem is in the control of an aircraft: here the state (position, orientation, velocity) of the aircraft evolves continuously over time in response to control inputs (throttle, control surfaces), as well as to disturbances (wind, hostile aircraft).

A *hybrid automaton* combines continuous-state and discrete-state dynamic systems, in order to model systems which evolve both continuously and according to discrete jumps. A hybrid automaton is defined to be a collection:

$$(S, \mathrm{Init}, In, f, \mathrm{Dom}, R) \tag{3}$$

where $S = Q \times \mathbb{R}^n$ is the Cartesian product of discrete and continuous states; $\mathrm{Init} \subseteq S$ is a set of initial states; $In = (\Sigma_u \times \Sigma_d) \times (\mathcal{U} \times \mathcal{D})$ is the set of actions and inputs; f is a function which takes state and input and maps to a new state, $f : S \times In \to S$; $\mathrm{Dom} \subseteq S$ is a *domain*; and $R : S \times In \to 2^S$ is a *transition relation.*

The state of the hybrid automaton is represented as a pair (q, x), describing the discrete and continuous state of the system. The continuous-state control system is "indexed" by the mode and thus may change as the system changes modes. Dom describes, for each mode, the subset of the continuous state space within which the continuous state may exist, and R describes the transition logic of the system, which may depend on continuous state and input, as well as discrete state and action. A trajectory of this hybrid system is defined as a sequence of continuous flows combined with discrete jumps. The introduction of disturbance parameters to both the control system defined by f and the reset relation defined by R will allow us to treat uncertainties, environmental disturbances, and actions of other systems.

The hybrid automaton model presented above allows for general nonlinear dynamics. This model was developed from those of Brockett, Branicky, Lygeros, and Nerode and Kohn, for which the emphasis was on extending the standard modeling, reachability and stability analyses, and controller design techniques to capture the interaction between the continuous and discrete dynamics. Other approaches to modeling hybrid systems involve extending finite automata to include simple continuous dynamics: these include the timed automata of Alur and Dill, linear hybrid automata of Henzinger, and hybrid I/O automata of Lynch.

2.2. Safety Verification

Much of the research in hybrid systems has been motivated by the need to verify the

behavior of safety critical system components. The problem of *safety verification* may be encoded as a condition on the region of operation in the system's state space: given a region of the state space which represents unsafe operation, *prove that the set of states from which the system can enter this unsafe region has empty intersection with the system's set of initial states*.

This problem may be posed as a property of the system's *reachable set* of states. There are two basic types of reachable sets. For a *forward reachable* set, the initial conditions are specified and one seeks to determine the set of all states that can be reached along trajectories that start in that set. Conversely, for a *backward reachable* set, a final or target set of states is specified, and one seeks to determine the set of states from which trajectories start that can reach that target set. For time invariant systems $\dot{x} = f(x)$ without input, it is easy to show that the backwards reachable set is the forwards reachable set of $\dot{x} = -f(x)$. It is interesting to note that the forward and backward reachable sets are not simply time reversals of each other

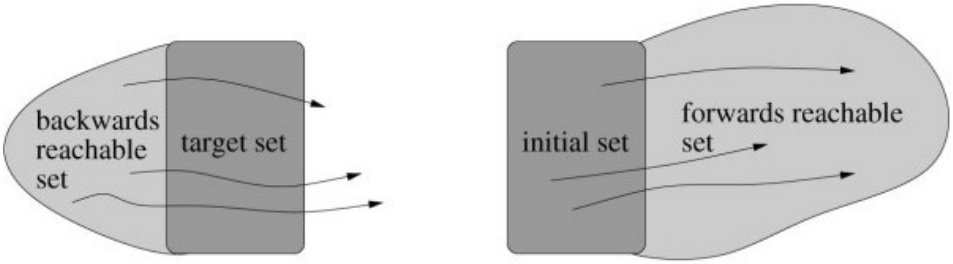

Figure 1: Difference between backwards and forwards reachable sets.

The difference is illustrated in Figure 1 for generic target and initial sets, in which the arrows represent trajectories of the system. Figure 2 illustrates how a backwards reachable set may be used to verify system safety.

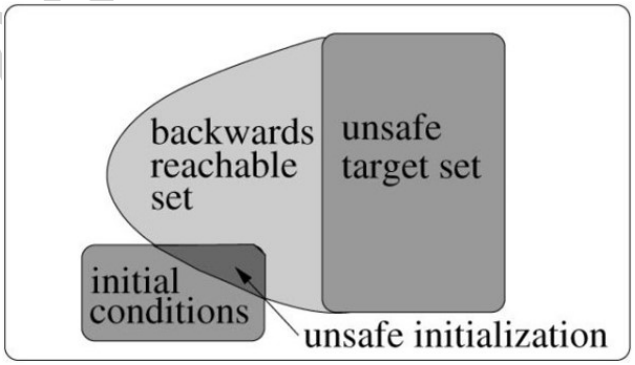

Figure 2: Using the backwards reachable set to verify safety.

Powerful software tools for the automatic safety verification of discrete systems have existed for some time, such as Murø (Dill), PVS, SMV, and SPIN. The verification of hybrid systems presents a more difficult challenge, primarily due to the uncountable number of distinct states in the continuous state space. In order to design and implement a methodology for hybrid system verification, it is necessary to represent reachable sets of continuous systems, and to evolve these reachable sets according to the system's dynamics.

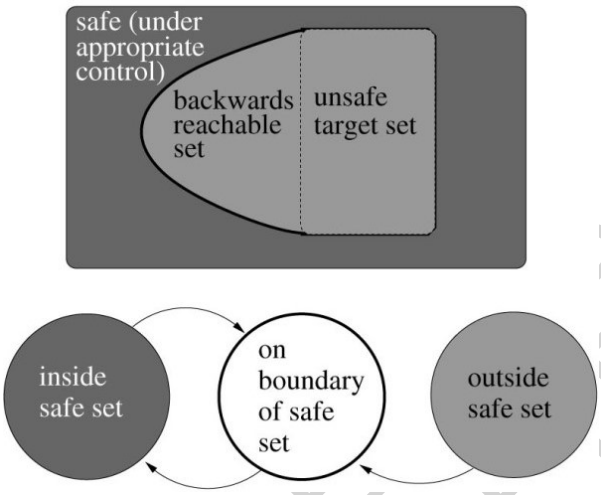

Figure 3: A discrete abstraction with appropriate control information.

It comes as no surprise that the size and shape of the reachable set depends on the control and disturbance inputs in the system: control variables may be chosen so as to minimize the size of the backwards reachable set from an unsafe target, whereas the full range of disturbance variables must be taken into account in this computation. Thus, the methodology for safety verification has two components. The first involves computing the backward reachable set from an *a priori* specified unsafe target set; the second involves extracting from this computation the control law which must be used on the boundary of the backwards reachable set, in order to keep the system state out of this reachable set. Application of this methodology results in a system description with three simple modes (see Figure 3). Outside of the backwards reachable set, and away from its boundary, the system may use any control law it likes and it will remain safe (labeled as "safe" in Figure 3). When the system state touches the reachable set or unsafe target set boundary, the particular control law which is guaranteed to keep the system from entering the interior of the reachable set must be used. Inside the reachable set (labeled as "outside safe set" in Figure 3), there is no control law which will guarantee safety, however application of the particular optimal control law used to compute the boundary may still result in the system becoming safe, if the disturbance is not playing optimally for itself.

3. Verifying Continuous Systems

Computing reachable sets for safety specifications has been a main focus of the control and computer aided verification communities for the past several years. In the past three years, several experimental reachability tools have been developed, and may be

classified according to how sets of states are represented, and the assumptions on the dynamics under which states are propagated. A group of methods which seek an efficient overapproximation of the reachable set is classified as "overapproximative". The tools **d/dt** (Maler) and *Checkmate* (Krogh) represent sets as convex polyhedra, and propagate these polyhedra under linear and affine dynamics, which could represent overapproximations of nonlinear dynamics along each surface of the polyhedra. *VeriSHIFT* uses ellipsoidal overapproximations of reach sets for linear systems with linear input; it implements techniques developed by Kurzhanski and Varaiya. The tool *Coho*, developed by Greenstreet and Mitchell uses as set representation two dimensional projections of higher dimensional non-convex polyhedra, and evolves these "projectagons" under affine over-approximations of nonlinear dynamics using linear programming. A recent algorithm by Tiwari and Khanna proposes to divide the continuous state space into a finite number of sets, and then to compute the reachable set using a discrete algorithm. The method works for polynomial dynamics and the subzero level sets of polynomials as set representation: by partitioning the state space into a "cylindrical algebraic decomposition" based on the system polynomials, a discrete approximation of the dynamics can be constructed.

A second group of methods is based on computing "convergent approximations" to reachable sets: here the goal is to represent as closely as possible the true reachable set. Methods include numerical computation of static Hamilton-Jacobi equations and to techniques from viability theory and set valued analysis. In our work, we have developed a reachability computation method based on level set techniques and viscosity solutions to Hamilton-Jacobi equations. A set is represented as the zero sublevel set of an appropriate function, and the boundary of this set is propagated under the nonlinear dynamics using a validated numerical approximation of a time dependent Hamilton-Jacobi-Isaacs (HJI) partial differential equation (PDE) governing system dynamics. These convergent approximative methods allow for both control inputs and disturbance inputs in the problem formulation, and they compute a numerical solution on a fixed grid (the mesh points do not move during the computation).

In most of the overapproximative schemes, the reachable set representation scales polynomially with the continuous state space dimension n. Exceptions include orthogonal polyhedra, which is exponential in n, and the algorithm based on cylindrical algebraic decomposition, in which the representation size depends on the dimension of the polynomials involved. Since algorithm execution time and its memory requirements generally scale linearly with the size of the representation of the reachable set, overapproximative schemes in which the set representation scales polynomially with n have a significant advantage over other schemes. However, these overapproximative schemes are generally too imprecise for problems in which the dynamics are nonlinear, and for which the shape of the reachable set is not a polygon or an ellipse. The schemes based on convergent approximations are exponential in n, and thus are not practical for problems of dimension greater than about five or six. However, these schemes can all handle nonlinear dynamics, they work within a differential game setting, and they make no assumptions about the shape of the reachable set.

In this section, using as motivation a classical pursuit-evasion game involving two identical vehicles, methodology and results for computing reachable sets for continuous

systems (1) are presented. The material in this section is presented in detail in the Ph.D. dissertation of Ian Mitchell.

-
-
-

Bibliography

Alur R., Courcoubetis C., Henzinger T.A., Ho P.H. (1993). Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems. In R.L. Grossman, A. Nerode, A.P. Ravn, H. Rischel, eds., *Hybrid Systems*, LNCS, pp. 366-392, New York: Springer Verlag.

Alur R., Dill D. (1994). A theory of timed automata. *Theoretical Computer Science* **126**, 183-235.

Anderson J. (1991). *Fundamentals of Aerodynamics*. New York: McGraw Hill Inc.

Asarin E., Bournez O., Dang T., Maler O. (2000). Approximate reachability analysis of piecewise-linear dynamical systems. In B. Krogh, N. Lynch, eds., *Hybrid Systems: Computation and Control*, LNCS 1790, pp. 21-31, Springer Verlag.

Aubin J.P., Lygeros J., Quincampoix M., Sastry S., Seube N. (2002). Impulse differential inclusions: A viability approach to hybrid systems. *IEEE Transactions on Automatic Control* **47** (1), 2-20.

Balluchi A., Benedetto M.D., Pinello C., Rossi C., Sangionvanni-Vincentelli A. (1998). Hybrid control for automotive engine management: The cut-off case. In T. Henzinger, S. Sastry, eds., *Hybrid Systems: Computation and Control*, no. 1386 in LNCS, pp. 13-32, New York: Springer Verlag.

Bardi M., Capuzzo-Dolcetta I. (1997). *Optimal Control and Viscosity Solutions of Hamilton-Jacobi-Bellman equations*. Boston: Birkhäuser.

Bayen A.M., Crück E., Tomlin C.J. (2002a). Guaranteed overapproximation of unsafe sets for continuous and hybrid systems: Solving the Hamilton-Jacobi equation using viability techniques. In C.J. Tomlin, M.R. Greenstreet, eds., *Hybrid Systems: Computation and Control*, LNCS 2289, pp. 90-104, Springer Verlag.

Bayen A.M., Mitchell I., Oishi M., Tomlin C.J. (2002b). Automatic envelope protection and cockpit interface analysis of an autoland system using hybrid system theory. Submitted to the AIAA Journal of Guidance, Control and Dynamics.

Bayen A.M., Tomlin C.J. (2001). A construction procedure using characteristics for viscosity solutions of the Hamilton-Jacobi equation. In *Proceedings of the IEEE Conference on Decision and Control*, pp. 1657-1662, Orlando, FL.

Bemporad A., Morari M. (1999). Verification of hybrid systems via mathematical programming. In F. Vaandrager, J.H. van Schuppen, eds., *Hybrid Systems: Computation and Control*, no. 1569 in LNCS, pp. 30-45, Berlin: Springer Verlag.

Botchkarev O., Tripakis S. (2000). Verification of hybrid systems with linear differential inclusions using ellipsoidal approximations. In B. Krogh, N. Lynch, eds., *Hybrid Systems: Computation and Control*, LNCS 1790, pp. 73-88, Springer Verlag.

Branicky M.S. (1994). *Control of Hybrid Systems*. Ph.D. thesis, Department of Electrical Engineering and Computer Sciences, Massachusetts Institute of Technology.

Brockett R. (1993). Hybrid models for motion control systems. In H. Trentelman, J. Willems, eds.,

Perspectives in Control, pp. 29-54, Boston: Birkhauser.

Büchi J.R., Landweber L.H. (1969). Solving sequential conditions by finite-state operators. In *Proceedings of the American Mathematical Society*, pp. 295-311.

Burch J., Clarke E.M., McMillan K., Dill D., Hwang L. (1992). Symbolic model checking: 10^{20} states and beyond. *Information and Computation* **98**(2), 142-170.

Cardaliaguet P., Quincampoix M., Saint-Pierre P. (1999). Set-valued numerical analysis for optimal control and differential games. In M. Bardi, T. Parthasarathy, T.E.S. Raghavan, eds., *Stochastic and Differential Games: Theory and Numerical Methods*, vol. 4 of *Annals of International Society of Dynamic Games*, Birkhäuser.

Cassandras C., Lafortune S. (1999). *Introduction to Discrete Event Systems*. Boston: Kluwer.

Church A. (1962). Logic, arithmetic, and automata. In *Proceedings of the International Congress of Mathematicians*, pp. 23-35.

Chutinan A., Krogh B.H. (2001). Verification of infinite-state dynamic systems using approximate quotient transition systems. *IEEE Transactions on Automatic Control* **46**(9), 1401-1410.

Crandall M.G., Evans L.C., Lions P.L. (1984). Some properties of viscosity solutions of Hamilton-Jacobi equations. *Transactions of the American Mathematical Society* **282**(2), 487-502.

Dang T., Maler O. (1998). Reachability analysis via face lifting. In S. Sastry, T. Henzinger, eds., *Hybrid Systems: Computation and Control*, no. 1386 in LNCS, pp. 96-109, Springer Verlag.

Dang T. (2000). *Vérification et synthèse des systèmes hybrides*. Ph.D. thesis, Institut National Polytechnique de Grenoble (Verimag).

Dill D.L. (1996). The Murø verification system. In *Conference on Computer-Aided Verification*, LNCS, pp. 390-393, Springer-Verlag.

Doyle J., Francis B., Tannenbaum A. (1992). *Feedback Control Theory*. New York: Macmillan.

Esprit (2001). Verification of Hybrid Systems: Results of a European Union Esprit Project. In O. Maler, ed., *European Journal of Control*, Vol. 7, Issue 4.

Greenstreet M., Mitchell I. (1999). Reachability analysis using polygonal projections. In F. Vaandrager, J.H. van Schuppen, eds., *Hybrid Systems: Computation and Control*, no. 1569 in LNCS, pp. 103-116, New York: Springer Verlag.

Henzinger T.A., Ho P., Wong-Toi H. (1997). HyTech: A model checker for hybrid systems. *Software Tools for Technology Transfer* **1**, 110-122.

Henzinger T. (1996). The theory of hybrid automata. In *Proceedings of the 11th Annual Symposium on Logic in Computer Science*, pp. 278-292, IEEE Computer Society Press.

Holzmann G. (1997). The model checker Spin. *IEEE Transactions on Software Engineering* **23**(5), 279-295. Special issue on Formal Methods in Software Practice.

Isaacs R. (1967). *Differential Games*. John Wiley.

Kurzhanski A. B., Varaiya P. (2000). Ellipsoidal techniques for reachability analysis. In B.Krogh, N. Lynch, eds., *Hybrid Systems: Computation and Control*, LNCS 1790, pp. 202-214, Springer Verlag.

Larsen K., Pettersson P., Yi W. (1997). Uppaal in a nutshell. *Software Tools for Technology Transfer* **1**.

Lygeros J. (1996). *Hierarchical, Hybrid Control of Large Scale Systems*. Ph.D. thesis, Department of Electrical Engineering and Computer Sciences, University of California at Berkeley.

Lynch N., Segala R., Vaandraager F. (2002). Hybrid I/O automata Submitted. Also, Technical Report MIT-LCS-TR-827b, MIT Laboratory for Computer Science, Cambridge, MA 02139.

Merz A.W. (1972). The game of two identical cars. *Journal of Optimization Theory and Applications* **9**(5), 324-343.

Mitchell I., Bayen A.M., Tomlin C.J. (2001a). Computing reachable sets for continuous dynamic games using level set methods. *IEEE Transactions on Automatic Control Submitted*.

Mitchell I., Bayen A.M., Tomlin C.J. (2001b). Validating a Hamilton-Jacobi approximation to hybrid system reachable sets. In M.D.D. Benedetto, A. Sangiovanni-Vincentelli, eds., *Hybrid Systems: Computation and Control*, LNCS 2034, pp. 418-432, Springer Verlag.

Mitchell I., Tomlin C.J. (2002). Overapproximating reachable sets by Hamilton-Jacobi projections. *Journal of Scientific Computing* Submitted May 2002. Accepted with minor revisions August 2002.

Mitchell I. (2001). Games of two identical vehicles. Technical report, SUDAAR 740, Stanford University Department of Aeronautics and Astronautics.

Mitchell I. (2002). *Application of Level Set Methods to Control and Reachability Problems in Continuous and Hybrid Systems*. Ph.D. thesis, Scientific Computing and Computational Mathematics, Stanford University.

Nerode A., Kohn W. (1993). Models for hybrid systems: Automata, topologies, controllability, observability. In R.L. Grossman, A. Nerode, A.P. Ravn, H. Rischel, eds., *Hybrid Systems*, LNCS 736, pp. 317-356, New York: Springer Verlag.

Oishi M., Tomlin C.J., Gopal V., Godbole D. (2001). Addressing multiobjective control: Safety and performance through constrained optimization. In M.D.D. Benedetto, A. Sangiovanni-Vincentelli, eds., *Hybrid Systems: Computation and Control*, LNCS 2034, pp. 459-472, Springer Verlag.

Osher S., Fedkiw R. (2002). *The Level Set Method and Dynamic Implicit Surfaces*. Springer-Verlag.

Osher S., Sethian J.A. (1988). Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics* **79**, 12-49.

Owre S., Rushby J.M., Shankar N. (1992). PVS: A prototype verification system. In D. Kapur, ed., *11th International Conference on Automated Deduction (CADE)*, vol. 607 of *Lecture Notes in Artificial Intelligence*, pp. 748-752, Saratoga, NY: Springer-Verlag.

Ramadge P.J.G., Wonham W.M. (1989). The control of discrete event dynamical systems. *Proceedings of the IEEE* **Vol. 77**(1), 81-98.

Sastry S.S. (1999). *Nonlinear Systems: Analysis, Stability and Control*. New York: Springer Verlag.

Tiwari A., Khanna G. (2002). Series of abstractions for hybrid automata. In C.J. Tomlin, M.R. Greenstreet, eds., *Hybrid Systems: Computation and Control*, LNCS 2289, pp. 465-478, Springer Verlag.

Tomlin C.J., Lygeros J., Sastry S. (July 2000). A game theoretic approach to controller design for hybrid systems. *Proceedings of the IEEE* **88**(7), 949-970.

Tomlin C.J., Mitchell I., Ghosh R. (2001). Safety verification of conflict resolution maneuvers. *IEEE Transactions on Intelligent Transportation Systems* **2**(2), 110-120. June.

Vidal R., Schaffert S., Lygeros J., Sastry S.S. (2000). Controlled invariance of discrete time systems. In B. Krogh, N. Lynch, eds., *Hybrid Systems: Computation and Control*, LNCS 1790, pp. 437-450, Springer Verlag.

von Neumann J., Morgenstern O. (1947). *Theory of Games and Economic Behavior*. Princeton University Press.

Yovine S. (1997). Kronos: A verification tool for real-time systems. *Software Tools for Technology Transfer* **1**, 123-133.

Biographical Sketches

Claire Tomlin received the Ph.D. degree in Electrical Engineering from the University of California, Berkeley, in 1998. Since September 1998 she has been an Assistant Professor in the Department of Aeronautics and Astronautics at Stanford University, with a courtesy appointment in Electrical Engineering. She was a graduate fellow in the Division of Applied Sciences at Harvard University in 1994, and she has been a visiting researcher at NASA Ames Research Center during 1994-1998, at Honeywell Technology Center in 1997, and at the University of British Columbia in 1994.

Claire Tomlin is a recipient of the Eckman Award of the American Automatic Control Council (2003), the AIAA Outstanding Teacher Award, Stanford (2001), NSF Career Award, Stanford (1999), Terman

Fellowship, Stanford (1998), the Bernard Friedman Memorial Prize in Applied Mathematics, Berkeley (1998), and the Zonta Amelia Earhart Awards for Aeronautics Research (1996-98). She was an invited participant in the National Academy of Engineering's Frontiers of Engineering Program in 2002, and she is currently a member of DARPA's Information Systems and Technology (ISAT) study group. Her research interests are in hybrid control systems, air traffic control automation, and flight management system analysis and design. Also, during the past three years, she has been involved in a project with the Stanford Medical School in the modeling and analysis of biological cell networks.

Alexandre Bayen received the B.S. degree in applied mathematics from the Ecole Polytechnique, Paris, France, in June 1998, the M.S. degree in aeronautics and astronautics from Stanford University in June 1999, and the Ph.D. in aeronautics and astronautics from Stanford University in December 2003. He is currently working as Director of the Autonomous Navigation Laboratory at the Delegation Generale de l'Armement, at the Department of Defense in France, where he holds the rank of Major. He was a Visiting Researcher at NASA Ames Research Center from 2000 to 2003. His research interests include combinatorial optimization, hybrid systems, air traffic automation, viability theory and optimal control. Mr. Bayen is the recipient of the Graduate Fellowship of the Délégation Générale pour l Armement (1998-2002) from France.

Ian Mitchell received a B.A.Sc. in Engineering Physics and an M.Sc. inComputer Science from the University of British Columbia, Canada in1994 and 1997 respectively, and a Ph.D. in Scientific Computing andComputational Mathematics from Stanford University in 2002. Afterspending a year as a postdoctoral researcher in the Department ofElectrical Engineering and Computer Science at the University ofCalifornia, Berkeley and the Department of Computer Science atStanford, Dr. Mitchell joined the faculty in the Department ofComputer Science at the University of British Columbia as an AssistantProfessor in August, 2003. He is the recipient of a 1999 SIAM/AAASMass Media Fellowship and a 1997-98 Stanford School of EngineeringGraduate Fellowship. His research interests include scientificcomputing, hybrid systems, verification and optimization.

Meeko Oishi received her Ph.D. from Mechanical Engineering at Stanford University in January 2004. She received the M.S. in Mechanical Engineering from Stanford in 2000 and the B.S.E. in Mechanical Engineering from Princeton University in 1998. Meeko has been a visiting researcher at NASA Ames Research Center (2001-2003) and at Honeywell Technology Center (2000), and has held summer internships at Boeing, Intel, and Sandia National Laboratories. She is the recipient of the NSF Graduate Research Fellowship (1998-1999, 2000-2002) and the John Bienkowski Memorial Prize from the Mechanical and Aerospace Department at Princeton (1998).

Her research interests include hybrid and nonlinear systems, user-interface analysis and design, flight management systems, and theoretical ecology.