

Design and synthesis of robust controllers for hybrid systems using modal logic:

Part I

(extended abstract)

J.M. Davoren and T. Moor

Research School of Information Sciences and Engineering

Australian National University

Canberra ACT 0200

Australia

E-mail: davoren@arp.anu.edu.au, tmoor@syseng.anu.edu.au

Hybrid systems are heterogeneous dynamical systems characterized by interacting continuous and discrete dynamics, and typically arise in the embedded software control of physical processes. Such mathematical models have proved fruitful in a great diversity of engineering applications, including automated transportation, robotics, and automated manufacturing. The design and synthesis of controllers is one of the most active areas in the field; the recent papers [4, 8, 10] include surveys of a range of approaches. Methods include the adaption of *optimal control* and *game-theoretic* techniques from continuous systems, and *supervisory control* ideas from *discrete events systems* (DES). The other dominant trend within hybrid systems comes from computer science. There, the focus is on extending computational models and methods for the formal analysis and verification of computer hardware and software to the setting of mixed discrete-continuous systems; see, for example, [1, 2, 3]. Performance specifications are encoded as formulas of a *temporal* or *modal logic*, or as an *automaton formal language* (as in DES theory), and hybrid system models (notably, the *hybrid automaton* model) are formally represented as some form of *transition system*, or generalized automaton. The task is to give a formal *proof* or *computation* demonstrating that a system satisfies given performance specifications. The principal methods are *symbolic model checking*, which consists of the direct computation of the set of states in a model at which a modal or temporal logic formula is satisfied, and the use of *deductive proof systems* for such logics, where one seeks to give a formal deduction of a specification formula from a theory (set of formulas) already known to be true of the system model. A comprehensive tutorial and survey of various logics for hybrid systems is offered in [6].

The present work combines ideas from both control theory and computer science to develop a logic-based approach to the design and synthesis of hybrid control systems. The *plant*, or system to be controlled, consists of a finite number of continuous systems $\dot{x} = F_c(x)$ over a common state space $X \subseteq \mathbb{R}^n$, indexed by symbols $c \in C$ in a finite (discrete) control alphabet. For example, the plant could arise from a single continuous control system $\dot{x} = f(x, u)$ subject to a finite collection of continuous state feedback control maps $g_c : X \rightarrow U$. The *controller* (or supervisor) exhibits discrete dynamics, realized on a finite state space Q , and includes an output mapping from Q to the control alphabet C . The controller must decide when to switch its discrete state q to another q' , and output a new control symbol c' , based on its continuous measurement of the plant state x . This hybrid control loop is illustrated in Figure 1.

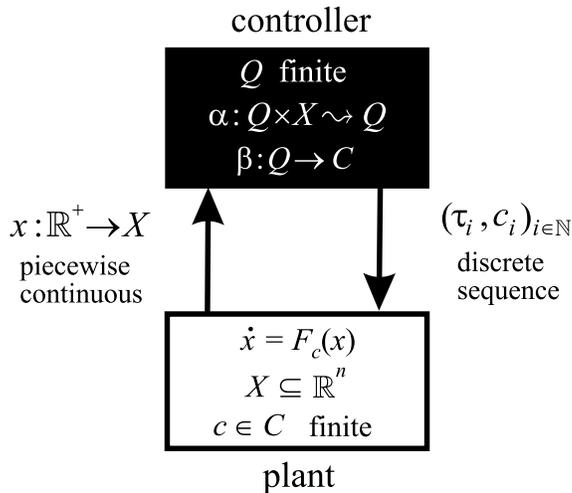


Figure 1: Basic hybrid control feedback loop

The controller transition relation $\alpha : Q \times X \rightsquigarrow Q$ determines two sorts of regions or subsets of the plant state space: regions $Inv_q \subseteq X$ in which the controller grants permission to *stay* in discrete state q and continue evolution according to the equation $\dot{x} = F_{\beta(q)}(x)$, where $\beta : Q \rightarrow C$ is the controller output function, and regions $Grd_{q,q'} \subseteq X$ in which the controller grants permission to *switch* discrete state from q to q' , and begin evolution according to $\dot{x} = F_{\beta(q')}(x)$. This switching control mechanism is illustrated in Figure 2. The widely accepted hybrid automaton model encodes this form of switched control, and can be seen as generating the closed-loop trajectories of such a plant-controller feedback system. In work on hybrid automata, a staying region Inv_q is also known as the *invariant region* for controller state $q \in Q$, and

a switching region $Grd_{q,q'}$ is also known as the *guard region* for the discrete transition (q, q') ; we continue with this established notation. This hybrid control configuration is an extension of the switching controller framework of [4], and is closely related to the DES supervisory control framework of [8]; the latter relationship is developed in more detail in [6].

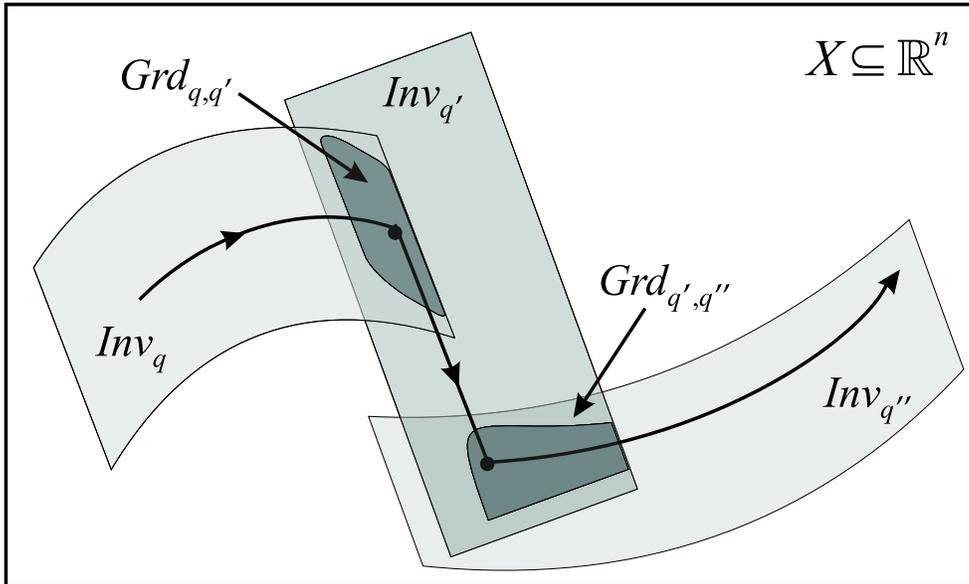


Figure 2: Controller designated regions of plant state space

We formulate and solve a quite general class of hybrid control problems. The problem is framed as the task of constructing a controller for a given plant, so that the resulting closed-loop hybrid automaton is guaranteed to satisfy a given list of performance specifications. The types of qualitative behavioural specifications we address go beyond the class of *safety*, *invariance* and *reachability* properties, which are the sole or primary focus of much of the current work on hybrid controller synthesis [4, 10]. Safety properties are usually formulated as negative reachability assertions, of the form: no hybrid trajectory starting in a given set $Init$ will ever enter a set Bad , where Bad is a proscribed set of plant states. In our target class of control problems, we additionally address positive or active behavioural requirements. We deal with a very general class of *event sequence* properties, of the form: all hybrid trajectories must traverse in a prescribed order through the blocks of a given finite partition $\{E_k\}_{k \in K}$ of the plant state space X , where the ordering is given by a total transition relation $next \subseteq K \times K$ on the partition index set K . This gives a general-purpose way of specifying the attainment of local goals along the course of hybrid trajectories, and integrating the type of

event sequence specifications examined in DES approaches to hybrid systems [8, 7, 9]. We also address the two basic forms of *liveness* properties: that all hybrid trajectories can be extended indefinitely, to make infinitely many discrete changes of state, and that all hybrid trajectories be *non-Zeno* (so not make infinitely many discrete switches in finite real time). Moreover, the construction is designed so that we can prove correctness with a quantifiable measure of robustness: every hybrid automaton within a class of *bounded variations* of the *nominal* closed-loop model satisfies each in the list of the specifications. The particular variation classes we consider can be interpreted as *sensor and actuator imprecision*, and fall within a larger framework of robustness concepts for hybrid automata proposed by Horn and Ramadge in [7].

At the workshop, this work will be presented in two 20 minute talks, one each by Davoren and Moor. In broad terms, the first will cover the setting-up and formulation of the class of control problems, and the statement of the main result, while the second will present the mathematical tools of modal logic used in the synthesis algorithm, and our software implementation of the algorithm using an approximated representation of state sets.

A full paper has been submitted for publication, and is available as a technical report [5] from the web address: http://arp.anu.edu.au/~davoren/hybrid_control/hybrid_control.html. There are also links to some video files of closed-loop simulations of solution controllers for example plant models and specifications generated by our prototype software implementation.

References

- [1] R. Alur, C. Courcoubetis, N. Halbwachs, T.A. Henzinger, P.-H. Ho, X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine. The algorithmic analysis of hybrid systems. *Theoretical Computer Science*, 138:3–34, 1995.
- [2] R. Alur, T.A. Henzinger, and P.-H. Ho. Automatic symbolic verification of embedded systems. *IEEE Transactions on Software Engineering*, 22:181–201, 1996.
- [3] R. Alur, T.A. Henzinger, G. Lafferriere, and G. Pappas. Discrete abstractions of hybrid systems. *Proceedings of the IEEE*, 88:971 – 984, July 2000.
- [4] E. Asarin, O. Bournez, T. Dang, O. Maler, , and A. Pnueli. Effective synthesis of switching controllers for linear systems. *Proceedings of the IEEE*, 88:1011 – 1025, July 2000.

- [5] J.M. Davoren and T. Moor. Logic-based design and synthesis of controllers for hybrid systems. Technical report, Dept. Systems Engineering, RSISE, ANU, July 2000. http://arp.anu.edu.au/~davoren/hybrid_control/hybrid_control.html.
- [6] J.M. Davoren and A. Nerode. Logics for hybrid systems. *Proceedings of the IEEE*, 88:985 – 1010, July 2000.
- [7] C. Horn and P.J. Ramadge. Robustness issues for hybrid systems. In *Proceedings of the 34th International Conference on Decision and Control, CDC'95*, pages 1467–1472. IEEE Press, 1995.
- [8] X. Koutsoukos, P.J. Antsaklis, J.A. Stiver, and M.D. Lemmon. Supervisory control of hybrid systems. *Proceedings of the IEEE*, 88:1026 – 1049, July 2000.
- [9] T. Moor and J. Raisch. Discrete control of switched linear systems. In *Proceedings of the European Control Conference 1999*, 1999.
- [10] C. Tomlin, J. Lygeros, and S. Sastry. A game-theoretic approach to controller design for hybrid systems. *Proceedings of the IEEE*, 88:949 – 970, July 2000.