Hybrid Control of Intelligent Multiple Vehicle Systems

Claire J. Tomlin*
Hybrid Systems Laboratory
Department of Aeronautics and Astronautics
Stanford University, Stanford, CA 94305
tomlin@stanford.edu

1 Introduction

One of the key challenges in designing a hierarchical control scheme for multiple vehicle systems is the incorporation of different models of computation at different levels of the hierarchy. The interaction between local vehicle control and the physical vehicle itself (at the bottom of the hierarchy) is most naturally modeled as a continuous-time or discrete-time system; while high level commands and control laws governing the system of vehicles may be naturally modeled as a discrete-state system, in which each discrete state corresponds to a particular control mode of the system.

The area of *hybrid systems* is loosely defined as the study of systems which involve the interaction of discrete event and continuous time dynamics, with the purpose of proving properties such as reachability and stability. The discrete event models naturally accommodate linguistic and qualitative information, and are used to model modes of operation of the system, such as the mode of flight of an aircraft, or the interaction and coordination between several aircraft. The continuous dynamics model the physical processes themselves, such as the continuous response of an aircraft to the forces of aileron and throttle.

We believe that the correct modeling and control paradigm for mission-critical multiple vehicle systems is that of hybrid systems; yet in order to understand the behavior of hybrid systems, to simulate, and to control these systems, theoretical advances and analytical tools are needed. These include computational methods for real-time reachability analysis and controller synthesis, a theory for optimal mode switching, and a methodology for the analysis and control of asynchronous distributed systems, in which data between subsystems can be sent and received at any time.

In the Hybrid Systems Laboratory at Stanford, we are developing these methods and applying them to the design of new algorithms for automated air traffic systems, and to fleets of unmanned aerial vehicles.

*Research supported by DARPA under the Software Enabled Control Program (administered by AFRL under contract F33615-99-C-3014).

2 Hybrid System Model

We incorporate accurate, nonlinear models of the continuous dynamics with models for discrete event dynamics, and we include continuous input variables (u, d) and discrete transitions between modes (σ_1, σ_2) to model both parameters that the designer may control as well as disturbance parameters that the designer must control against [1, 2, 3, 4]. The model is compact, yet rich enough to describe both the evolution of continuous and discrete dynamics, is capable of modeling uncertainty in both the continuous and discrete variables. A hybrid system is described by

$$H = (Q \times \boldsymbol{R}^n, U \times D, \Sigma_1 \times \Sigma_2, f, \delta, Inv) \tag{1}$$

with discrete state $q_i \in Q$ and continuous state $x \in \boldsymbol{R}^n$. The nonlinear continuous dynamics in each mode may be modeled by the differential equation:

$$\dot{x} = f(q_i, x, u, d) \tag{2}$$

where $u \subseteq \boldsymbol{R}^u$ is the *control input* which models the actions of the controller, $d \subseteq \boldsymbol{R}^d$ is the *disturbance input* which models the actions of the environment. Transitions between discrete modes may be modeled by the discrete analog, a transition function:

$$(q_{i+1}, x_{new}) = \delta(q_i, x_{old}, \sigma_1, \sigma_2) \tag{3}$$

where $(\sigma_1, \sigma_2) \in \Sigma_1 \times \Sigma_2$ represent discrete input and disturbance vectors, or *events* that cause the system to switch modes. Associated to each discrete state $q_i \in Q$ is a subset of the continuous state space, called the *invariant* $\{x \in \boldsymbol{R}^n \mid (q_i, x) \in Inv\}$ in which the system may evolve when in q_i. A graphical version of the model is illustrated in Figure 1.

The "safety" verification problem and the hybrid controller synthesis problem may be posed as the reachability question: *How does one compute the continuous control inputs, as well as the discrete logic, so that the trajectories of the system remain within a set of desired states?* It has been proven that the answers to these questions are in general undecidable, for all but the simplest continuous dynamics. It has been proven in [2] that, even for these undecidable cases, a conceptual solution to the verification and controller design problem is obtained by solving a special form of a Hamilton-Jacobi partial differential equation as we discuss in the next section.

3 Computational Methods for the Design of Hybrid Systems

We have developed a constructive methodology for the design of hybrid controllers for multi-agent systems. In the continuous case, optimal control laws may be derived as solutions of the

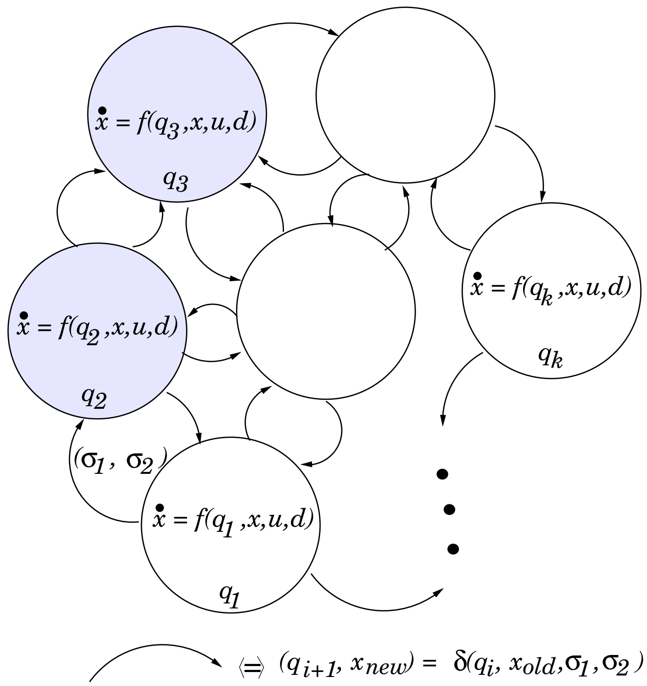

Figure 1: Nonlinear Hybrid System Model.

Hamilton-Jacobi equation (arising from continuous games), while discrete controllers can be synthesized by solving games on finite automata. Both methods can be seen as special cases of a generic game-theoretic approach to safety control. This insight can likely be extended to more general hybrid control objectives. Here our software tools and methods are based on efficient algorithms for the solution of Hamilton-Jacobi equations. We discuss this briefly in the context of computing backwards reachable sets while avoiding "escape" sets in which the state can jump to safety in another discrete state (this is the heart of the controller synthesis algorithm for the hybrid system of 1 as presented in [2]):

$$
\begin{array}{ll}
\text{Let} & V^0 = G, V^{-1} = \emptyset, i = 0. \\
\text{While} & V^i \neq V^{i-1} \text{ do} \\
& V^{i-1} = V^i \cup \mathit{Reach}(\mathit{Pre}_2(V^i), \mathit{Pre}_1(V^i))) \\
& i = i - 1 \\
\text{end} &
\end{array}
\tag{4}
$$

where $Pre_1(V^i)$ represents an "escape" set, and contains all states for which a controllable action σ_1 can force the state to remain outside V^i, $Pre_2(V^i)$ contains all states for which an uncontrollable action σ_2 can force the state into V^i, and $\mathit{Reach}(Pre_2(V^i), Pre_1(V^i))$ are all states which can evolve in the continuous evolution to $Pre_2(V^i)$, avoiding $Pre_1(V^i)$. The computation of $\mathit{Reach}(Pre_2(V^i), Pre_1(V^i))$ is performed by a solving a set of "coupled" Hamilton-

Jacobi equations.

Even though theoretically the resulting algorithms for controller synthesis are sound, their applicability is limited in practice because of the difficulty of solving the PDEs; in certain cases solutions in the conventional sense may not even exist, and one may have to resort to weaker solution concepts, such as viscosity solutions. In addition, the class of systems for which solutions may be obtained analytically is even more limited; solutions will have to be computed numerically in most cases. At Stanford, we have been working on numerical tools for efficiently computing or at least approximating the solution of Hamilton-Jacobi PDEs: a level set method, suitably adapted to take discrete transitions into account, is the heart of a new computational algorithm for performing reachability analysis, and subsequently verification and controller synthesis, on hybrid systems [5]. Indeed, the computation of $Reach(Pre_2(\cdot), Pre_1(\cdot))$ in (4) could be performed as a coupling of two evolving fronts. The reachable set of a hybrid system may have a non-smooth boundary due to switches in (u^*, d^*), non-smooth initial data, or the formation of shocks. The level set scheme propagates these discontinuities, yet its implementation may require a very small time step to do this accurately.

4 Testbeds: Air Traffic Control and Unmanned Aerial Vehicles

In the current air traffic control system in the United States, all of the control is concentrated on the ground: sensory information from aircraft distributed throughout the airspace is sent to human air traffic controllers who use ground-based navigation and surveillance equipment to manually route aircraft along "highways in the sky". As a result, localized events, such as bad weather, aircraft failure, and runway or airport closure have repercussions throughout the whole country. The FAA has proposed a solution called "free flight", in which aircraft fly along optimal routes which ensure that the flight time is short, the fuel consumption is minimized, inclement weather is avoided, and safe distances between aircraft are maintained. Using the tools of hybrid systems, we intend to show that free flight is possible in a new air traffic management architecture which automates much of the current air traffic control functionality. My students and I are focusing on the problem of multiple aircraft collision avoidance, which seeks a safe and efficient method to resolving trajectory conflicts between aircraft in a free flight environment. To test our algorithms, we are building a realistic simulation environment for free flight, as well as a network of small unmanned aircraft which are equipped with onboard GPS sensors, datalink, and automatic control algorithms. Our first vehicle (12 foot wingspan) is illustrated below; we are currently working on the avionics for the second vehicle (10-foot wingspan).

Figure 2: The Stanford DragonFly Aircraft.

References

[1] C. Tomlin, J. Lygeros, and S. Sastry, "Synthesizing controllers for nonlinear hybrid systems," in *Hybrid Systems: Computation and Control* (T. Henzinger and S. Sastry, eds.), no. 1386 in LNCS, pp. 360–373, New York: Springer Verlag, 1998.

[2] C. J. Tomlin, *Hybrid Control of Air Traffic Management Systems*. PhD thesis, Department of Electrical Engineering, University of California, Berkeley, 1998.

[3] J. Lygeros, C. Tomlin, and S. Sastry, "Controllers for reachability specifications for hybrid systems," *Automatica*, vol. 35, no. 3, 1999.

[4] C. Tomlin, J. Lygeros, and S. Sastry, "Controller design for hybrid systems," *Proceedings of the IEEE*, vol. 88, no. 7, July 2000.

[5] I. Mitchell and C. Tomlin, "Level set methods for computation in hybrid systems," in *Hybrid Systems: Computation and Control* (B. Krogh and N. Lynch, eds.), LNCS, Springer Verlag, 2000.