Safe Implementations of Supervisory Commands

Michael Lemmon and Christopher J. Bett*

Department of Electrical Engineering
University of Notre Dame
Notre Dame, IN 46556

Abstract. This paper compares two different types of control strategies used to safely implement supervisory commands of hybrid dynamical systems. Both approaches considered in this paper switch between members of a family of control agents to ensure that constraints on the plant state are not violated at any time. The first approach is motivated by a hybrid system architecture outlined in [7] and uses a Fliess functional series of the plant's output to form a system of linear inequalities characterizing safe control inputs. Control signals are determined by solving a sequence of linear programs. The second approach is a model reference control approach to hybrid systems introduced in [8] and uses a known safe dynamical reference model to characterize the desired plant behavior. The controller is determined by representing the resulting error dynamics as a linear parameter varying system and applying linear robust control techniques to enforce a bounded amplitude performance level. The fundamental results underlying each of the methods are derived; both approaches are compared with regard to their complexity, performance, and sensitivity to modeling uncertainty. A numerical example is included for illustration.

1 Introduction

This paper considers the high level supervision of continuous time dynamical control systems evolving over a state set which is dense in $\mathbb{R}^n$. It is assumed that a supervisory command is characterized by a set of guard conditions and a goal condition. These guard and goal conditions are inequality conditions on the plant's state. A control system is used to implement the supervisory command. This controller is said to be “safe” when the controlled plant’s state trajectory triggers the goal condition in finite time without triggering any of the guard conditions. This paper compares two different types of controllers used to safely implement supervisory commands.

Both approaches considered in this paper switch between members of a family of control agents to ensure the guard conditions are not triggered. The first approach is motivated by a hybrid system architecture outlined in [7]. This approach uses a Fliess functional series of the plant’s output to form a system of linear inequalities characterizing safe control inputs. In this method, control signals can be determined by solving a sequence of linear programs (LP). The second approach is a model reference control approach to hybrid systems introduced in [8]. In this approach, the controlled plant follows a reference model which is known to be safe. The error dynamics of this system are represented as a linear parameter varying (LPV) system whose controllers enforce a bounded amplitude performance level. This paper formally derives the fundamental results behind both of these methods and compares both approaches with regard to their complexity, performance, and sensitivity to modeling uncertainty.

A formal definition of “safe” controllers is given in section 2. The remainder of the paper discusses the two methods for characterizing safe controllers which were outlined above. The first method will be referred to as the LP-method since it solves a sequence of linear programs to determine safe control signals. The LP-method is discussed in section 3. The fundamental result in section 3 is a set of inequality constraints characterizing locally safe piecewise constant control signals. The second method is referred to as the MRC-method since it uses a model reference control (MRC) approach to formulate the controller synthesis problem. The MRC method is discussed in section 4. The fundamental results in this section are sufficient conditions characterizing controllers ensuring bounded-amplitude performance for the switched control system. Section 5 compares both methods and draws some general conclusions about their relative strengths and weaknesses.

* The authors gratefully acknowledge the partial financial support of the Army Research Office (DAAH04-95-1-0600, DAAH04-95-1-0134).
2 Safe Supervisory Controllers

Hybrid dynamical systems arise when the time and/or the state space have mixed continuous and discrete natures. Such systems frequently arise when computers are used to control continuous state systems. In recent years, specific attention has been focused on hybrid systems in which a discrete-event system is used to supervise the behavior of plants whose state spaces are dense in \( \mathbb{R}^n \). In this class of hybrid control systems, commands are issued by a discrete-event system to direct the behavior of the plant. These commands are high-level directives to the plant which require that the supervised plant satisfy logical conditions on the plant’s state. The simplest set of conditions are inequality conditions on the plant’s state.

Assume that the plant’s dynamics are generated by the differential equation

\[ \dot{x} = f(x, u) \]  

where \( x \in \mathbb{R}^n \) is the state, \( u \in \mathbb{R}^m \) is the control input, and \( f : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n \) is a Lipschitz continuous mapping. A supervisory directive to this system is characterized by a set of functionals, \( h_j : \mathbb{R}^n \rightarrow \mathbb{R} \) for \( j = 0, \ldots, N \), that separate the state space. The functionals, \( h_j \), are said to separate the state space if and only if for all \( x, y \in \mathbb{R}^n \) such that \( h_j(x) > 0 \) and \( h_j(y) < 0 \), there exists \( 0 < \lambda < 1 \) such that \( h_j(\lambda x + (1 - \lambda)y) = 0 \). The functional, \( h_0 \), is said to be the goal trigger and the other functionals, \( h_j \) for \( j = 1, \ldots, N \), are called the guard triggers. Consider a state feedback controller,

\[ u = k(x) \]  

Such a controller is said to be safe if and only if there exist finite times \( T_1 \) and \( T_2 \) (\( T_1 < T_2 \)) such that

- \( h_j(x(t)) < 0 \) for all \( t_0 \leq t < T_2 \) (\( j = 1, \ldots, N \)),
- \( h_0(x(t)) < 0 \) for all \( t_0 \leq t < T_1 \),
- and \( h_0(x(t)) > 0 \) for all \( T_1 < t < T_2 \).

Essentially, these conditions state that the goal condition is triggered in finite time without any of the guard triggers being violated. Assume that we have a monotone increasing function \( r(t) \) such that \( r(0) = h_0(x(0)) \) and \( r(T_1) = 0 \). We can use this “reference” function to rewrite the preceding list of conditions as a set of inequality constraints such that the guard triggers (\( j = 1, \ldots, N \)) satisfy \( h_j(x(t)) < 0 \) and the goal trigger satisfies \( h_0(x(t)) - r(t) > 0 \) for all \( t \in [0, T_2] \).

3 LP-Method

The LP-method is motivated by a hybrid system architecture outlined in [7]. This method characterizes safe control signals as a set of linear inequality constraints. The LP-method assumes that the plant’s differential equation has the form

\[ \dot{x} = f_0(x) + \sum_{i=1}^m f_i(x)u_i(t) \]  

where \( f_i : \mathbb{R}^n \rightarrow \mathbb{R}^n \) are analytic functions forming a nonsingular distribution of vector fields in \( \mathbb{R}^n \). It is also assumed that the set of trigger functions \( \{h_j\}_{j=1}^N \) is analytic.

Assume that the trigger functions, \( h_j(x(t)) \), are known at time, \( t \). Under appropriate conditions, it is possible to represent the trigger functions at time \( t + \delta \) as a Fliess functional series. To formally state these results, some notational conventions need to be introduced. Let \( f : \mathbb{R}^n \rightarrow \mathbb{R}^n \) be a vector of analytic functions, \( f = [f_1, f_2, \ldots, f_n] \) where \( f_i : \mathbb{R}^n \rightarrow \mathbb{R}^n \) (\( i = 1, \ldots, n \)). The Lie derivative of an analytic function \( h : \mathbb{R}^n \rightarrow \mathbb{R} \) with respect to vector field \( f \) is

\[ L_f h(x) = \sum_{i=1}^n \frac{\partial h}{\partial x_i} f_i(x) \]  

Let \( i \in \{1, \ldots, m\} \) be an index and let \( i_1, \ldots, i_k \) be a sequence of indices of length \( k \) called a multi-index. The set of all multi-indices will be denoted as \( I^k \). Associated with the multi-index \( i_1, \ldots, i_k \) is the iterated integral,