6th annual IEEE Conference on Automation Science and Engineering
Marriott Eaton Centre Hotel
Toronto, Ontario, Canada, August 21-24, 2010

Trajectory-based Controller Design for Hybrid Systems with Affine Continuous Dynamics

A. Agung Julius
Department of Electrical, Computer and Systems Engineering
Rensselaer Polytechnic Institute
Troy, NY 12180, USA.
Email:agung@ecse.rpi.edu

Abstract— **In this paper we propose a method for feedback controller synthesis using the concept of control autobisimulation function. Control autobisimulation function (CAF) is the analog of control Lyapunov function for approximate bisimulation. Approximate bisimulation has been used to establish robustness (in ℓ_∞ sense) of execution trajectories of dynamical systems and hybrid systems, resulting in trajectory-based safety verification procedures.**

CAF is used to characterize the family of all feedback control laws that result in a close loop system with an autobisimulation function. Further, we use a heuristic potential function based idea to construct a safe feedback control law for a nominal initial state, and use the trajectory-robustness property to guarantee that the control law is also safe for other initial states in a neighborhood of the nominal initial state.

Keywords: hybrid system, trajectory based, controller synthesis.

I. Introduction

The issue of safety/reachability is very important in the theory of hybrid systems. The *analysis* part of this issue, i.e. the investigation whether a given hybrid system model with given initial conditions can reach a certain state, or set of states has received a lot of attention from the hybrid systems community. It has also resulted in a lot of practical applications. The *synthesis* part of the safety/reachability issue deals with the construction of control laws/algorithms for systems with input that result in safe executions. Some of the methods for safety/reachability analysis can be extended for controller synthesis. For example, the optimal control method in [1] and the simulation based method in [2] directly characterize the influence of the control input in the reachability formulation. The predicate abstraction technique for systems with piecewise affine dynamics in polytope sets leads to a control procedure based on the transversality of the vector field on the facets of the polytopes [3], [4]. The technique for discrete-time system presented in [5] utilizes partitioning of the state space by polygonal approximation of the reachable set. The notion of approximate bisimulation has previously been used for controller synthesis for nonlinear dynamical systems [6], [7]. In this case, the notion is used to establish a quantization of the continuous state space, which can result in a countable transition system approximation of the original dynamics.

The class of safety/reachability analysis methods that is closely related to this paper is the *trajectory-based analysis*. These are methods that aim to assess the safety/reachability based on the execution trajectories of the system, or the simulations thereof. The main conceptual tool that we use in this paper, the *approximate bisimulation*, was developed by Girard and Pappas [8], and has been used for trajectory based analysis of hybrid systems in [9], [10], [11]. In this paper we present a controller synthesis method that is based on trajectory-based analysis. We introduce the notion of *control autobisimulation function (CAF)*, to characterize a class of feedback laws, called the *admissible feedback laws*, that result in closed loop systems that admit an (auto)bisimulation function. Therefore, the control autobisimulation function can be thought of as an analog of control Lyapunov function [12], [13] for autobisimulation. For any given initial condition, we use a heuristic method based on the idea of potential function to construct an admissible feedback law that results in a "valid" execution trajectory[1]. The use of CAF enables us to use trajectory robustness (a la approximate bisimulation) to guarantee formally the validity of the control law for a neighborhood around that initial condition. By repeating this procedure for a finite set of initial conditions, we can cover a compact set of initial conditions.

The controller design method presented in this paper therefore consists of two steps. The first step is to characterize the class of admissible feedback laws. The second step is to construct an admissible feedback law for each initial condition, that can be verified to result in a valid trajectory (for example, through simulation). We present an example demonstrating that although the presented design method has a heuristic step in it, the resulting controller is formally guaranteed to be correct, and that it is possible to achieve that with only a few simulation runs. This approach can thus be regarded as a highly parallelizable and lightweight (no quantization of state space is required) complement to the more formal approaches, such as [6], [7].

II. Autobisimulation Function

We recall the use of (auto)bisimulation function for establishing trajectory robustness for autonomous systems (sys-

[1]What "valid" means will be discussed later.

tems without input). Given an autonomous dynamical system

$$\Sigma_{\text{aut}} : \frac{dx}{dt} = f(x), x \in \mathbb{R}^n, \tag{1}$$

with $f(x)$ locally Lipschitz.

Notation 1: The trajectory of the dynamical system (1) with initial condition $x(0) = x_0$ is denoted as $\xi(t, x_0)$.

We define an autobisimulation function as follows.

Definition 1: [10] A continuously differentiable function $\phi : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}_+$ is an **autobisimulation function** of (1) if for any $x, x' \in \mathbb{R}^n$,

$$\phi(x, x') \geq \|x - x'\|, \tag{2}$$

$$\nabla_x \phi(x, x') f(x) + \nabla_{x'} \phi(x, x') f(x') \leq 0. \tag{3}$$

Autobisimulation function is used to provide a formal guarantee that the distance between two execution trajectories of the dynamical system (1) is bounded in the ℓ_∞ sense. This result is stated in the following proposition.

Proposition 1: [10] Given a dynamical system (1) and an autobisimulation function ϕ, for any $x_0, x_0' \in \mathbb{R}^n$,

$$\|\xi(t, x_0) - \xi(t, x_0')\| \leq \phi(x_0, x_0'), \ \forall t \geq 0. \tag{4}$$

Further, if $\phi(x, x')$ is designed to be a (pseudo)metric in $\mathbb{R}^n$, $\phi(x, x') = \|x - x'\|_\phi$, or if it is a class $\mathcal{K}$ function of a metric in $\mathbb{R}^n$, $\phi(x, x') = \alpha(\|x - x'\|)$, where α is a class $\mathcal{K}$ function[2], then Proposition 1 can be used as the foundation of trajectory-based safety analysis [10].

Remark 1: The notion of autobisimulation is related to but different from the notion of incremental global asymptotic stability (δGAS) introduced by Angeli (cf. [14]) in the following sense. With autobisimulation, we do not require asymptotic convergence of the trajectories as in δGAS. Therefore, any Lyapunov function that characterizes δGAS can be used as autobisimulation function, but not vice versa.

Suppose that there is a given compact set of initial states Init $\subset \mathbb{R}^n$, where the state is initiated at $t = 0$, i.e. $x(0) \in$ Init. Also, we assume that there is a set of goal states, Goal$\subset \mathbb{R}^n$. We require that any execution trajectory starting in Init enters the goal set before time $t = T > 0$. For a given initial condition $x_0 \in \mathbb{R}^n$, suppose that

$$\inf_{0 \leq t \leq T} d_\phi(\xi(t, x_0), \text{Unsafe}) = \delta > 0, \tag{5}$$

$$\xi(T, x_0) \in \text{Goal}, \tag{6}$$

$$d_\phi(\xi(T, x_0), \text{Goal}^C) < \delta, \tag{7}$$

where

$$d_\phi(\xi(t, x_0), \text{Unsafe}) := \inf_{x' \in \text{Unsafe}} \phi(\xi(t, x_0), x'),$$
$$d_\phi(\xi(T, x_0), \text{Goal}^C) := \inf_{x' \notin \text{Goal}} \phi(\xi(T, x_0), x').$$

Notice that this implies that for the time interval $0 \leq t \leq T$ the trajectory $\xi(t, x_0)$ is safe (i.e. it does not enter the Unsafe set).

Notation 2: For any $x \in \mathbb{R}^n$ and $\delta \geq 0$, we denote the set $\{x' \in \mathbb{R}^n \mid \phi(x, x') \leq \delta\}$ as $B_\phi(x, \delta)$.

[2]A function $\alpha : R_+ \to R_+$ is a class $\mathcal{K}$ function if it is continuous, monotonically increasing, and $\alpha(0) = 0$.

We can show (c.f. [10]) that in this case, any initial condition $x_0' \in B_\phi(x_0, \delta)$ will also result in a safe trajectory, i.e. $\xi(t, x_0') \notin$ Unsafe, for $0 \leq t \leq T$. Moreover, we can formally guarantee that the trajectory $\xi(t, x_0')$ will enter the goal set before time $t = T$. Therefore, by computing the execution trajectory $\xi(t, x_0)$, we can generalize its safety property to a nonzero measure neighborhood of initial states around x_0. This is the foundation for formal safety verification of a compact set of initial states using a finite number of execution trajectories.

III. Control Autobisimulation Function and Trajectory-Based Feedback Control

Consider a dynamical system with input

$$\Sigma_{\text{inp}} : \frac{dx}{dt} = f(x, u), x \in \mathbb{R}^n, u \in \mathcal{U} \subset \mathbb{R}^m. \tag{8}$$

where the function $f(x, u)$ is locally Lipschitz in x and continuous in u. As discussed in the previous section, assume that we also have the set of initial states, Init, and the goal set, Goal. Suppose that we are given the following control problem:

Problem 1: Design a feedback control law $u = g(x)$ such that for any initial state $x_0 \in$ Init, the trajectory of the closed loop system enters Goal before time $t = T > 0$, and in the time interval $[0, T]$ the trajectory is safe. Such a trajectory is called a **valid trajectory**.

We will discuss how the notion of trajectory-robustness discussed in the previous section can also be used in trajectory-based controller synthesis. The key concept in this approach is the *control autobisimulation function (CAF)*.

Definition 2: A continuously differentiable function $\psi : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}_+$ is a **control autobisimulation function** of (8) if for any $x, x' \in \mathbb{R}^n$,

$$\psi(x, x') \geq \|x - x'\|, \tag{9}$$

and there exists a function $k : \mathbb{R}^n \to \mathcal{U}$ such that

$$\nabla_x \psi(x, x') f(x, k(x)) + \nabla_{x'} \psi(x, x') f(x', k(x')) \leq 0. \tag{10}$$

The control autobisimulation function is an analog of the control Lyapunov function (CLF) [12], for approximate bisimulation [8], [10]. While control Lyapunov function has been used to construct control laws that guarantee stability (e.g. [13]), we shall use the control autobisimulation function to construct control laws that guarantee trajectory robustness.

Remark 2: One can compare the control autobisimulation function with control Lyapunov function of the product of the system (8) with itself

$$\frac{d}{dt} \begin{bmatrix} x \\ x' \end{bmatrix} = \begin{bmatrix} f(x, u) \\ f(x', u') \end{bmatrix}.$$

In this case, notice that unlike for CLF, for CAF we cannot set u and u' to be any functions of x and x'. Rather, the inputs u and u' must be the same function of their respective states (x and x'). Therefore, in this aspect, the requirement for CAF is more stringent than that for CLF.

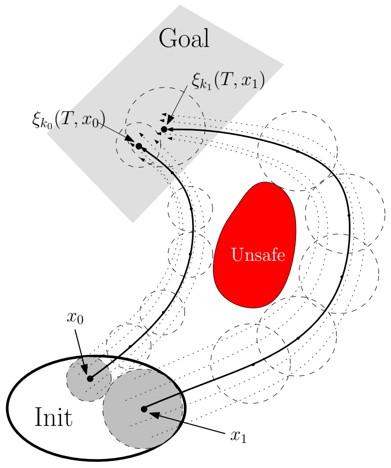

Fig. 1. An illustration for trajectory-based controller synthesis.

A consequence of the existence of a CAF as in Definition 2 is the existence of feedback control laws

$$u = k(x), \tag{11}$$

such that the closed loop system obtained from (8) and (11),

$$\frac{dx}{dt} = f(x, k(x)), x \in \mathbb{R}^n, \tag{12}$$

has $\psi(\cdot, \cdot)$ as an autobisimulation function. This fact can be deduced immediately from Definition 2.

Definition 3: For a given dynamical system with input Σ_{inp} and a control autobisimulation function ψ, the class of all feedback control laws $k(\cdot)$ that satisfy (10) is called the **class of admissible feedback laws**, $\eta(\Sigma_{\text{inp}}, \psi)$.

Notation 3: For a given dynamical system with input Σ_{inp} and a feedback control law $u = k(x)$, the closed loop trajectory with initial condition $x(0) = x_0$ is denoted by $\xi_k(t, x_0)$.

The controller synthesis paradigm in this paper can be stated as follows. We construct feedback controllers from the class of feasible feedback laws. By definition, the closed loop system will then admit a predefined autobisimulation function. This means that the trajectory-robustness property discussed in Section II is guaranteed to hold. Please refer to Figure 1. Suppose that for a given initial state $x_0 \in \text{Init}$, we can design a feedback law $u = k_0(x)$ that results in a closed loop execution trajectory $\xi_{g_0}(t, x_0)$ satisfying

$$\inf_{0 \leq t \leq T} d_\phi(\xi_{k_0}(t, x_0), \text{Unsafe}) = \delta_0 > 0, \tag{13}$$

$$\xi_{k_0}(T, x_0) \in \text{Goal}, \tag{14}$$

$$d_\phi(\xi_{k_0}(T, x_0), \text{Goal}^C) > \delta_0. \tag{15}$$

Then, as previously shown, we can obtain a neighborhood around x_0, $B_\phi(x_0, \delta_0)$ consisting of other initial states for which the feedback law $u = k_0(x)$ is guaranteed to result in execution trajectories that are safe and meet the goal state.

We can repeat the procedure for a different initial state, say $x_1 \in \text{Init}$. Suppose that we can then design a feedback law $u = k_1(x)$ that results in a closed loop execution trajectory $\xi_{k_1}(t, x_1)$ that is safe and meets the goal set as shown in Figure 1. As before, we also obtain a neighborhood around x_1 for which the feedback law $u = k_1(x)$ is guaranteed to yield execution trajectories that are safe and meet the goal state. As the result of this process, we now obtain two feedback laws which are valid for two different subsets of Init (not necessarily disjoint). The goal of the controller synthesis procedure is then to cover the entire initial set Init with different control laws as such.

IV. Controller Synthesis for Systems with Affine Dynamics

Based on the exposition in the previous section, it is clear that to implement the idea of trajectory-based controller synthesis, we need to have a control autobisimulation function (CAF) ψ, and the feedback control laws for each initial condition that we evaluate. Moreover, each the feedback control laws must belong to the class of admissible controller $\eta(\Sigma_{\text{inp}}, \psi)$. The synthesis of the CAF and the controllers for systems with linear affine dynamics is discussed in this section.

A. Two-stage Controller Design

A specific class of these systems are systems with linear affine dynamics. These are systems of the form

$$\Sigma_{\text{lin}} : \frac{dx}{dt} = Ax + f + Bu, \; x \in \mathbb{R}^n, u \in \mathbb{R}^m, \tag{16}$$

where $A \in \mathbb{R}^{n \times n}$, $f \in \mathbb{R}^n$, and $B \in \mathbb{R}^{n \times m}$. For such systems, we propose to construct CAF as quadratic functions [8], [10], [15]. That is, we assume that

$$\psi(x, x') = \frac{1}{2}(x - x')^T P (x - x'), \tag{17}$$

where $P \in \mathbb{R}^{n \times n}$ is a positive definite matrix. From Definition 2, it follows that inequality (10) becomes

$$(x - x')^T P \left(A(x - x') + B(k(x) - k(x'))\right) \leq 0. \tag{18}$$

We propose to construct a feedback law of the form

$$u(t) = k(x) = Kx + v(t), \tag{19}$$

where $K \in \mathbb{R}^{m \times n}$ and $v(t) \in \mathbb{R}^m$ is a time-varying function, both to be determined later. By substituting (19) into (18), we obtain

$$(x - x')^T P \left(A + BK\right)(x - x') \leq 0. \tag{20}$$

Finding K that satisfies inequality (20) is equivalent to finding K such that $(A + BK)$ is Hurwitz. A well known result in control theory (c.f. [16]) states that there exist P and K such that (20) holds if and only if (A, B) is stabilizable. In this case, there are well known methods to synthesize the suitable P and K. For example, by solving the following linear matrix inequality[3] (LMI) (see Section 7.2.1 in [17])

$$A\tilde{P} + B\tilde{D} + \tilde{P}A^T + D^T B \leq 0, \; \tilde{P} > 0, \tag{21}$$

[3]We use the fact that A is Hurwitz if and only if A^T is Hurwitz.

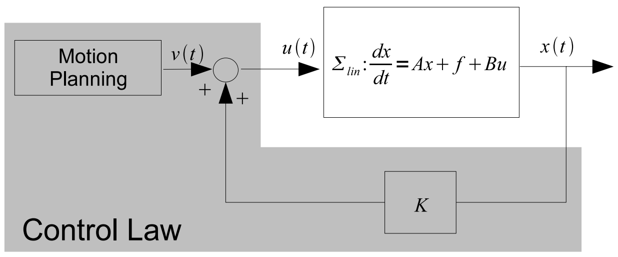

Fig. 2. The control law to be designed is shown in the shaded area. First, we compute the feedback gain K such that the inner loop system admits an autobisimulation function. Then, the signal $v(t)$ is designed such that the control goal is achieved for a given initial state.

for $\tilde{P} \in \mathbb{R}^{n\times n}$ and $D \in \mathbb{R}^{m\times n}$. The feedback gain K can be computed from

$$K^T = D\tilde{P}^{-1}. \tag{22}$$

From here, P can be obtained by solving the Lyapunov equation

$$(A+BK)^T P + P(A+BK) \le 0,\ \ P > 0. \tag{23}$$

By applying the feedback control law (19) to Σ_{lin}, we obtain a closed loop system

$$\Sigma_{\text{cl}} : \frac{dx}{dt} = (A+BK)x + f + Bv,\ \ x \in \mathbb{R}^n, v \in \mathbb{R}^m. \tag{24}$$

Notice that following to the discussion above, given that $(A+BK)$ is Hurwitz, we are still free to design $v(t)$. In other words, whatever $v(t)$ is, the control law is admissible (see Definition 3). The remaining task in the controller design is therefore to use $v(t)$ to steer the trajectories of the closed loop system. The goal is to steer any given initial state in Init to the goal set, without entering the unsafe set. We propose to use heuristic ideas related to motion planning problem with obstacle avoidance. In Subsection IV-C, we present an example where this problem is solved. Please refer to Figure 2 for the block diagram of the controller synthesis.

B. Bounded Input Set

Consider the case where the input set is bounded, i.e. for all $t \in \mathbb{R}_+$

$$\|u(t)\| \le M, \tag{25}$$

for some positive bound M. We need to ensure that the feedback law given in (19) satisfies this condition. The fact that

$$\|u(t)\| \le \|K\|\,\|x\| + \|v(t)\|, \tag{26}$$

indicates that minimizing $\|K\|$ can alleviate the difficulty of designing $v(t)$ that satisfies the control input bound, as demonstrated by the example in Subsection IV-C. We can approach this problem by modifying the LMI (21) into the following semidefinite programming problem [18]

$$\begin{aligned} &\min \|D\| \ \text{ subject to} \\ &A\tilde{P} + BD + \tilde{P}A^T + D^T B^T \le 0, \\ &\tilde{P} - I > 0. \end{aligned} \tag{27}$$

It is clear that any $(\tilde{P}, D)$ that is feasible for (21) can be scaled so that it is feasible for (27). However, we also have

$$\|K\| \le \|D\| \left\|\tilde{P}^{-1}\right\| \le \|D\|, \tag{28}$$

which shows that solving (27) effectively leads to the minimization of an upper bound for $\|K\|$.

C. Numerical Example

The following example illustrates the controller synthesis procedure. Consider the following problem. Given an affine linear system

$$\Sigma : \frac{dx}{dt} = Ax + f + Bu, \tag{29}$$
$$A = \begin{bmatrix} 0 & 1 \\ -1 & 0.1 \end{bmatrix}; B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}; f = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

We want to design feedback control laws that will bring any state in the initial set Init $= \{x \in \mathbb{R}^2 \mid \|x\| \le 0.2\}$ to the goal set Goal $= \{x \in \mathbb{R}^2 \mid \|x - [2,0]^T\| \le 0.1\}$ without entering the unsafe set Unsafe $= \{x \in \mathbb{R}^2 \mid \|x - [1,1]^T\| \le 0.3\}$, with the input constraint $\|u(t)\| \le 2$.

We want to implement a feedback control

$$u(t) = Kx + v(t) = \begin{bmatrix} k_1 & k_2 \end{bmatrix} x + v(t). \tag{30}$$

By solving the semidefinite program (27), we obtain the controller

$$K_{\text{opt}} = \begin{bmatrix} 0 & -0.1 \end{bmatrix},\ \text{and}\ \|K_{\text{opt}}\| = 0.1, \tag{31}$$

and a suitable control autobisimulation function $\psi(x,x') = \frac{1}{2}\|x - x'\|^2$.

For any given initial condition $[x_{1,0}, x_{2,0}]^T \in$ Init, the next step is to synthesize the steering input $v(t)$ of the closed loop system that results in a valid trajectory[4]. We also need to satisfy the bound on the input magnitude

$$\|K_{\text{opt}}x(t) + v(t)\| \le 2. \tag{32}$$

The design that we pick is rather *ad hoc*, and inspired by the potential function/navigation function technique in motion planning with obstacle avoidance (see e.g. [19], [20]). We notice that there are two tasks at hand: (i) steer the trajectory to the goal set, and (ii) avoid the unsafe set. We design a controller for each task, and combine them. The controller for the first task can be designed as a feedback law as follows

$$v_{\text{goal}}(t) = 1 - x_2(t). \tag{33}$$

The rationale behind this design is that it makes the center of the goal set a stable equilibrium. Thus, if we ignore the unsafe set, this steering input will ensure that the goal set will be reached.

[4]i.e. $v(t)$ can vary depending on the initial condition.

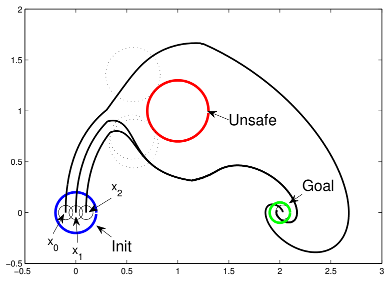

Fig. 3. The close loop execution trajectories resulting from the controller synthesis procedure. For each initial state, we can observe that the trajectories are robustly safe and meet the goal set. We also show the robust neighborhood around each initial state.

The design of the second controller borrows an idea from potential function technique. We first define a scalar weight function $W : \mathbb{R}^2 \rightarrow \mathbb{R}_+$,

$$W(x) = \begin{cases} \frac{0.1}{(0.3 - \|[1,1]^T - x\|)^2}, & \|[1,1]^T - x\| \leq 0.7 \\ 0, & \text{otherwise.} \end{cases} \tag{34}$$

We then define the second controller also as feedback law, as follows

$$v_{\text{safe}}(t) = \begin{cases} W(x), & \text{if } B^T\left([1,1]^T - x\right) \geq 0 \\ -W(x), & \text{if } B^T\left([1,1]^T - x\right) < 0. \end{cases} \tag{35}$$

The rationale behind this controller design is as follows. Notice that the sign of $v_{\text{safe}}(t)$ is chose such that $Bv_{\text{safe}}(t)$ always pushes the state away from the unsafe set. The magnitude of this "pushing force" is defined by the weight function $W(x)$. Thus, it is zero if the state is sufficiently far away from the unsafe set, and blows up to infinity as it approaches the unsafe set.

Finally, we need to ensure that (32) is met. Based on the size of the goal set, any robust neighborhood that we obtain for any particular initial condition cannot have radius more than 0.1. Therefore, we can bound the variation of the control input $u(t)$ within any robust neighborhood as follows

$$\|\Delta u\| \leq \|K\| \, \|\Delta x\| = 0.1 \cdot 0.1 = 0.01. \tag{36}$$

We then define the overall steering input such that

$$u(t) = Z(K_{\text{opt}} x(t) + v_{\text{goal}}(t) + v_{\text{safe}}(t), 1.99), \tag{37}$$

where $Z : \mathbb{R} \times \mathbb{R}_+ \rightarrow \mathbb{R}$ is simply the saturation function

$$Z(x,y) := \begin{cases} -1.99, & x < -y. \\ x, & |x| \leq y \\ 1.99, & x > y. \end{cases} \tag{38}$$

The benefit of the trajectory-based controller synthesis **allows us to simply simulate and tune the proposed controller for a finite set of initial conditions**. The robustness property of the trajectories then allows us to generalize the results to cover the whole initial set. Figure 3 shows the application of this design procedure for three different initial conditions. In each case, we can see that the designed steering input $v(t)$ results in a valid trajectory. We also compute the robust neighborhood around each initial state. Only three trajectories are shown so as not to clutter the figure. However we can further show that with uniform sampling of the initial set, we can cover it with 21 robust neighborhood.

V. Controller Synthesis for Hybrid Systems

The design paradigm presented in the previous section can be also applied to hybrid systems. Consider a standard model of hybrid systems, $\mathcal{H} = (\mathcal{X}, \mathcal{L}, E, Inv, \Sigma)$, where $\mathcal{X}$ is the continuous state space of the system, $\mathcal{L}$ is the finite set of discrete states (locations), E is the set of transitions, $Inv : \mathcal{L} \rightarrow 2^{\mathcal{X}}$ is the invariant set of a location, and Σ is a family of dynamical systems with input that defines the continuous dynamics in each location. That is, for each location $l \in \mathcal{L}$, we define the continuous dynamics as

$$\Sigma(l) : \frac{dx}{dt} = f_l(x,u), x \in \mathcal{X}, u \in \mathcal{U}. \tag{39}$$

A transition $e \in E$ is a 4-tuple (l, l', g, r), where $l \in \mathcal{L}$ is the origin of the transition, $l' \in \mathcal{L}$ is the target of the transition and that each location, $g \subset \partial Inv(l)$ is the guard of the transition, which is a subset of the boundary of the invariant set of location l, and $r : g \rightarrow Inv(l')$ is the reset map that resets the continuous state at the new location. We assume that the reset map r is continuous, the continuous state space is $\mathbb{R}^n$, the invariant sets are closed, $f_l(x,u)$ is locally Lipschitz in x and continuous in u for all $l \in L$, the transitions are deterministic in the sense that the guards of all outgoing transitions from a location are disjoint, and that the system does not deadlock or possess Zeno behavior. In analyzing the safety of the system, we assume that there is a subset Unsafe $\subset \mathcal{X} \times \mathcal{L}$ of unsafe states. A trajectory of the hybrid system corresponds to an unsafe execution if it intersects with the unsafe set.

To define the control problem, we define a set of initial state Init$\subset \mathcal{X} \times \mathcal{L}$, in which we assume the hybrid state begins at $t = 0$. We also define a goal set, Goal$\subset \mathcal{X} \times \mathcal{L}$ in which all executions must terminate. As before, the control problem is defined as finding the feedback control strategy that is guaranteed to bring any initial state in Init to the goal set without entering the unsafe set. Without any loss of generality, we can assume that the set Init is contained in (the invariant set of) one location, called $l_{\text{init}} \in \mathcal{L}$. If this is not the case, we can divide the problem into several subproblems, each with an Init set contained in a specific location. Similarly, we can assume the Goal is also entirely contained in one location, called $l_{\text{goal}} \in \mathcal{L}$.

We approach this problem with a *hierarchical control* design, which can be described in the following steps:

Step 1: Discrete Synthesis. We compute a discrete trajectory that starts in l_{init} and ends in l_{goal}. By discrete trajectory, we mean an alternating sequence of locations and transitions

$$l_{\text{init}} = l_0 \xrightarrow{e_1} l_1 \xrightarrow{e_2} l_2 \xrightarrow{e_3} \cdots \xrightarrow{e_N} l_N = l_{\text{goal}}. \tag{40}$$

Such a discrete trajectory is not necessarily unique, but at this step we only need one. The computation of a discrete trajectory like this, albeit formally undecidable, is a standard procedure in formal verification of discrete event systems [21].

Step 2: Continuous Synthesis. In this step, we synthesize the continuous controller for each of the visited locations ($l_{0,1,\ldots,N}$) in order to implement the computed discrete trajectory. Basically, in each location l_i, we define an initial set based on how l_i is reached from l_{i-1}. We then formulate the control problem of bringing the continuous state from this initial set to the goal set, which is defined as a set beyond the guard transition e_i that will bring the state to location l_{i+1} without entering the forbidden set. The forbidden set is defined as the union of Unsafe, and the guards of other outgoing transitions from l_i. If we are able to construct a continuous controller that implements the discrete trajectory, then the hybrid control problem is solved. Otherwise, we go back to Step 1, and compute another discrete trajectory.

Remark 3: Similar two-step approach to solve the control problem with application in motion control synthesis for fully actuated robots has been discussed in the literature (see [22] and the references therein). The discrete part of the control goal in [22] is expressed as a temporal logic formula, which is richer than the one presented in this paper. However, we would like to point out that the continuous synthesis presented in this paper can also be applied to implement the continuous part of the controller in [22].

Remark 4: Due to space limitation, we refer the reader to the extended version of this manuscript [23] for an explicit representation of the algorithm outlined in this section, and a numerical example for its implementation.

VI. Discussion

The result presented in this paper can be generalized by replacing the motion planning box in Figure 2 with other means of obtaining valid trajectories for given initial conditions. These include other heuristics based methods, such as fuzzy control [24], or expert system based methods (cf. [25]) that allow for integration of human operators' experience into the control strategy. The advantage offered by the theory of trajectory-based analysis is that we can formally guarantee the safety and correctness of the resulting controllers.

Finally, we would like remark that although in this paper we restrict our attention to affine continuous dynamics, the theory of trajectory-based analysis is applicable to nonlinear dynamics as well. In further investigation, we will explore the use of local trajectory-based analysis techniques for nonlinear dynamics, extending the results presented in [15].

Acknowledgement: This work is partially supported by an NSF CAREER grant (#0953976). The author would like to thank Jeff Ban for the stimulating discussions that lead to this paper.

References

[1] I. Mitchell and C. J. Tomlin, "Level set methods in for computation in hybrid systems," in *Hybrid Systems: Computation and Control*, vol. 1790 of *LNCS*, pp. 310–323, Springer Verlag, 2000.

[2] J. Kapinski, B. H. Krogh, O. Maler, and O. Stursberg, "On systematic simulation of open continuous systems," in *Hybrid Systems: Computation and Control*, vol. 2623 of *LNCS*, pp. 283–297, Springer, 2003.

[3] C. Belta and L. Habets, "Controlling a class of nonlinear systems on rectangles," *IEEE Trans. Automatic Control*, vol. 51, no. 11, pp. 1749–1759, 2006.

[4] L. Habets, P. J. Collins, and J. H. van Schuppen, "Reachability and control synthesis for piecewise-affine hybrid systems on simplices," *IEEE Trans. Automatic Control*, vol. 51, no. 6, pp. 938–948, 2006.

[5] G. Reissig, "Computation of discrete abstractions of arbitrary memory span for nonlinear sampled systems," in *Hybrid Systems: Computation and Control*, vol. 5469 of *LNCS*, pp. 306–320, Springer, 2009.

[6] P. Tabuada, "An approximate simulation approach to symbolic control," *IEEE Trans. Automatic Control*, vol. 53, no. 6, pp. 1406–1418, 2008.

[7] G. Pola, A. Girard, and P. Tabuada, "Approximately bisimilar symbolic models for nonlinear control systems," *Automatica*, vol. 44, no. 10, pp. 2508–2516, 2008.

[8] A. Girard and G. J. Pappas, "Approximation metrics for discrete and continuous systems," *IEEE Trans. Automatic Control*, vol. 52, no. 5, pp. 782–798, 2007.

[9] A. Girard and G. J. Pappas, "Verification using simulation," in *Hybrid Systems: Computation and Control*, vol. 3927 of *LNCS*, pp. 272–286, Springer Verlag, 2006.

[10] A. A. Julius, G. Fainekos, M. Anand, I. Lee, and G. J. Pappas, "Robust test generation and coverage for hybrid systems," in *Hybrid Systems: Computation and Control*, vol. 4416 of *LNCS*, pp. 329–342, Springer Verlag, 2007.

[11] F. Lerda, J. Kapinski, E. M. Clarke, and B. H. Krogh, "Verification of supervisory control software using state proximity and merging," in *Hybrid Systems: Computation and Control*, vol. 4981 of *LNCS*, pp. 344–357, 2008.

[12] Z. Artstein, "Stabilization with relaxed controls," *Nonlinear Analysis*, vol. 15, no. 11, pp. 1163–1170, 1983.

[13] E. D. Sontag, "A 'universal' construction of Artstein's theorem on nonlinear stabilization," *Systems and Control Letters*, vol. 13, no. 2, pp. 117–123, 1989.

[14] D. Angeli, "A Lyapunov approach to incremental stability properties," *IEEE Trans. Automatic Control*, vol. 47, no. 3, pp. 410–421, 2002.

[15] A. A. Julius and G. J. Pappas, "Trajectory based verification using local finite-time invariance," in *Hybrid Systems: Computation and Control*, vol. 5469 of *LNCS*, pp. 223–236, Springer, 2009.

[16] W. L. Brogan, *Modern control theory*. New Jersey: Prentice Hall International, 1991.

[17] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in Systems and Control Theory*. Philadelphia: SIAM, 1994.

[18] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004. Available online at www.stanford.edu/ boyd/cvxbook/.

[19] A. Rimon and D. E. Koditschek, "Exact robot navigation using artificial potential fields," *IEEE Trans. on Robotics and Automation*, vol. 8, no. 5, pp. 501–518, 1992.

[20] D. C. Conner, A. A. Rizzi, and H. Choset, "Composition of local potential functions for global robot control and navigation," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3546–3551, 2003.

[21] E. M. Clarke, O. Grumberg, and D. A. Peled, *Model Checking*. MIT Press, 1999.

[22] G. E. Fainekos, A. Girard, H. Kress-Gazit, and G. J. Pappas, "Temporal logic motion planning for dynamic mobile robots," *Automatica*, vol. 45, no. 2, pp. 343–352, 2009.

[23] A. A. Julius, "Trajectory-based controller design for hybrid systems with affine continuous dynamics." submitted to IEEE CASE 2010, extended manuscript. Available online at http://www.ecse.rpi.edu/∼agung, 2010.

[24] K. M. Passino and S. Yurkovich, *Fuzzy Control*. Addison-Wesley, 1998.

[25] B. K. Bose, "Expert system, fuzzy logic, and neural network applications in power electronics and motion control," *Proceedings of the IEEE*, vol. 82, no. 8, pp. 1303 – 1323, 1994.