Mode Switching Synthesis for Reachability Specifications

T. John Koo[1,2], George J. Pappas[1], and Shankar Sastry[2]

[1] Department of EE, University of Pennsylvania, Philadelphia, PA 19104
{jkoo,pappasg}@grasp.cis.upenn.edu,
[2] Department of EECS, University of California at Berkeley, Berkeley, CA 94704,
{koo,sastry}@eecs.berkeley.edu

Abstract. In many control applications, a specific set of output tracking controllers of satisfactory performance have already been designed and must be used. When such a collection of *control modes* is available, an important problem is to be able to accomplish a variety of high level tasks by appropriately switching between the low-level control modes. In this paper, we define a concept of control modes, and propose a framework for determining the sequence of control modes that will satisfy reachability tasks. Our framework exploits the structure of output tracking controllers in order to extract a finite graph where the mode switching problem can be efficiently solved, and then implement it using the continuous controllers. Our approach is illustrated on a helicopter example, where we determine the mode switching logic that achieves the high-altitude takeoff task from a hover mode.

1 Introduction

Large scale systems like automated highway systems, air traffic management systems, unmanned aerial vehicles are multi-agent, multi-objective systems that operate in many modes of operation. This results in systems of very high complexity which may dramatically limit the applicability of current analysis and design methods. A natural way to reduce the complexity of system design uses compositional methods which solve a complex problem by decomposing it into a sequence of smaller problems of manageable complexity. For example, in sophisticated flight management systems [3], modern aircraft fly from origin to destination while satisfying a large number of aerodynamic, scheduling, and air traffic constraints by switching among a finite set of *flight modes*, where each flight mode essentially corresponds to a different output tracking controller.

More generally, given a continuous control system, a *control mode* is defined as the operation of the system under a controller that is *guaranteed* to track a certain class of output trajectories. Different outputs of interest correspond to different control modes. Given a set of control modes, the mode switching problem attempts to find a finite *sequence* of the control modes as well as *switching conditions* in order to satisfy various tasks. In this paper, we focus on reachability tasks.

M.D. Di Benedetto, A. Sangiovanni-Vincentelli (Eds.): HSCC 2001, LNCS 2034, pp. 333–346, 2001.

Problem 1. Given a control system and a finite set of control modes for the system, determine whether there exists a finite sequence of modes that will steer the system from an initial control mode to a desired final control mode. If such a sequence exists, then determine the switching conditions.

Clearly, in this setup, many more interesting problems can be formulated. For example one can ask what are the optimal switching conditions, where optimality can mean minimum time, or minimum number of switchings. Furthermore, one can ask whether a set of modes is sufficient for performing a reachability, or more general, task. In this paper, we focus on Problem 1, while setting up the framework for considering these more general questions in the future.

In its full generality, Problem 1 can be tackled using controller synthesis methods for hybrid systems [1,7,12,14]. However, termination conditions for such synthesis procedures are limited [6], and the computational complexity of such procedures could be prohibitive due to nested reachability computations. It is therefore evident that in order to scale our methods to real-life examples, structure must be imposed on the system, and subsequently exploited in our analysis and synthesis methods.

In order to reduce the complexity of the mode switching problem, we start by assuming that output tracking control laws have been designed for each control mode. Feedback greatly simplifies the continuous models in each discrete location since the complexity of the continuous behavior is now reduced to the complexity of the trajectories we design. Therefore, many reachability computations that are required in our approach can be greatly simplified by properly *designing* the desired trajectories. Even though feedback control simplifies the continuous complexity, the problem of having nested reachability computations is still present. In order to avoid such expensive computations, we place a *consistency* condition in our mode switching logic which is reminiscent of the notion of *bisimulation.* We propose an algorithm which given an initial set of control modes, constructs a *control mode graph* which refines the initial control modes but is consistent. Construction of the mode graph can be done off-line or every time a new control mode is designed, allowing the mode switching problem to be efficiently solved on-line, in real time.

2 Problem Formulation

Throughout this paper, we consider a nonlinear system modeled by differential equations of the form

$$\dot{x}(t) = f(x(t)) + g(x(t))u(t), \quad x(t_0) = x_0, \quad t \geq t_0 \tag{2.1}$$

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}^p$, $f(x) : \mathbb{R}^n \to \mathbb{R}^n$ and $g(x) : \mathbb{R}^n \to \mathbb{R}^n \times \mathbb{R}^p$. The system is assumed to be as smooth as needed. We now define a concept of control mode.

Definition 1 (Control Modes). *A control mode, labeled by q_i where $i \in \{1, \ldots, N\}$, is the operation of the nonlinear system (2.1) under a closed-loop feedback controller of the form*

$$u(t) = k_i(x(t), r_i(t)) \tag{2.2}$$

associated with an output $y_i(t) = h_i(x(t))$ such that $y_i(t)$ shall track $r_i(t)$ where $y_i(t), r_i(t) \in \mathbb{R}^{m_i}$, $h_i : \mathbb{R}^n \to \mathbb{R}^{m_i}$, $k_i : \mathbb{R}^n \times \mathbb{R}^{m_i} \to \mathbb{R}^p$ for each $i \in \{1, \dots, N\}$. We assume that $r_i \in \mathcal{R}_i$, the class of output trajectories associated with the control mode q_i, when the initial condition of the system (2.1) starts in the set $S_i(r_i) \subseteq X_i$, output tracking is guaranteed and the state satisfies a set of state constraints $X_i \subseteq \mathbb{R}^n$.

The trajectory $r_i(t)$ is the desired output trajectory, and $y_i(t)$ is the output vector which shall track $r_i(t)$. Notice that in general the initial set may be a function of the trajectory r_i, thus we denote it as $S_i(r_i)$. This is because even though trajectory tracking controllers are guaranteed to converge for any initial condition, trajectory tracking in the presence of state constraints or input constraints can be guaranteed only if the initial tracking error is sufficiently small. In this paper we are interested in switching between controllers, rather than the design of output tracking controllers. We therefore make the following assumption.

Assumption 1 *For each control mode q_i, $i \in \{1, \dots, N\}$, we assume that a controller of the form (2.2) has been designed which achieves output tracking such that $y_i(t)$ shall track $r_i(t)$ where $r_i \in \mathcal{R}_i \neq \emptyset$, while the state satisfies the set of state constraints $x(t) \in X_i \subseteq \mathbb{R}^n$, when the initial condition of the system (2.1) starts in the set $S_i(r_i) \subseteq X_i \subseteq \mathbb{R}^n$.*

The above assumption is justified given the maturity of output tracking controllers for large classes of linear and nonlinear systems [15]. Based on different design methodologies, the notion of *output tracking* could be different as it could be uniform asymptotic, exponential, etc. Depending on the complexity on the computation, one may choose a specific notion of tracking for solving Problem 1. In order to motivate the discussion, we present a planar helicopter model and a set of controllers in which each controller satisfies Assumption 1 but with different output functions and state constraints.

Example 1. **Multi-Modal Control of a Planar Helicopter Model**. In this example, a helicopter model [4] described in longitudinal and vertical axes with simplified force and moment generation processes is considered. The x, z-axes of the spatial frame are pointing to north and down directions. The body x-axis is defined from the center of gravity to the nose of the helicopter, and body z-axis is pointing down from the center of gravity. The motion of the helicopter is controlled by main rotor thrust, T_M and longitudinal tilt path angle, a_M. The pitch angle is defined by θ. The equations of motion can be expressed as:

$$\begin{bmatrix} \ddot{p}_x(t) \\ \ddot{p}_z(t) \end{bmatrix} = \frac{1}{m} \begin{bmatrix} \cos\theta(t) & \sin\theta(t) \\ -\sin\theta)(t) & \cos\theta(t) \end{bmatrix} \begin{bmatrix} -T_M(t)\sin a_M(t) \\ -T_M(t)\cos a_M(t) \end{bmatrix} + \begin{bmatrix} 0 \\ g \end{bmatrix} \tag{2.3}$$

$$\ddot{\theta}(t) = \frac{1}{I_y}(M_M a_M(t) + h_M T_M(t) \sin a_M(t)) \tag{2.4}$$

The state vector and input vector are defined as $x = [p_x, \dot{p}_x, p_z, \dot{p}_z, \theta, \dot{\theta}]^T \in \mathbb{R}^6$ and $u = [T_M, a_M]^T \in \mathbb{R}^2$, respectively.

Control Mode	Output	Reference	Constraint
q_1: Hover	$y_1 = [p_x, p_z]^T$	r_1	X_1
q_2: Cruise	$y_2 = [\dot{p}_x, \dot{p}_z]^T$	r_2	X_2
q_3: Ascend	$y_3 = [\dot{p}_x, \dot{p}_z]^T$	r_3	X_3
q_4: Descend	$y_4 = [\dot{p}_x, \dot{p}_z]^T$	r_4	X_4

Define $X_1 = X_2 = \mathbb{R} \times (\underline{v}_x, \overline{v}_x) \times \mathbb{R} \times (\underline{v}_z, \overline{v}_z) \times (-\pi/2, \pi/2) \times \mathbb{R}$, $X_3 = \mathbb{R} \times (\underline{v}_x^{cr}, \overline{v}_x) \times \mathbb{R} \times (\underline{v}_z, \overline{v}_z^{as}) \times (-\pi/2, \pi/2) \times \mathbb{R}$, and $X_4 = \mathbb{R} \times (\underline{v}_x^{cr}, \overline{v}_x) \times \mathbb{R} \times (\underline{v}_z^{de}, \overline{v}_z) \times (-\pi/2, \pi/2) \times \mathbb{R}$ where $\underline{v}_x < 0 < \underline{v}_x^{cr} < \overline{v}_x$ and $\underline{v}_z < \underline{v}_z^{de} < 0 < \overline{v}_z^{as} < \overline{v}_z$. To satisfy Assumption 1, several control design methodologies can be used to design a controller for each discrete control mode q_i where $i \in \{1, 2, 3, 4\}$. Each controller implementation can be specified as $u = k_i(x, r_i)$ with $r_i \in \mathcal{R}_i$ where $\mathcal{R}_i$ defines the class of admissible output trajectories in mode q_i, and the performance of the closed-loop system can be specified by *initial set*, $S_i(r_i)$, and *flow*, $\phi_i(t, r_i, x_0)$ where $x_0 \in S_i(r_i)$.

Given two control modes, one cannot simply switch from one control mode to another due to incompatible constraints. A natural question is then whether this mode reachability task can be achieved by a *finite sequence* of modes. Based on the above example, we can now define the mode switching problem that we will address in this paper.

Problem 2 (Mode Switching Problem). Given an initial control mode q_S with desired reference r_S, does there exist a sequence of control modes such that the system can reach a desired mode q_F with reference r_F? If so, then determine a mode sequence $q_S \rightarrow \cdots q_i \rightarrow q_j \cdots \rightarrow q_F$ along with trajectories r_i for each control mode q_i, as well as conditions for switching between the control modes.

For the control modes defined in Example 1, one can define a task of having the Hover mode q_1 as an initial mode and ask for a finite control mode sequence to reach the Ascend mode q_3. Any solution to this problem leads to a feasible execution of the task called *high-altitude takeoff* according to flight instruction for helicopter pilots. Note that Problem 2 is a reachability problem. More generally, one can envision more complicated tasks that can be specified in temporal logic, but in this paper we restrict our attention to reachability specifications.

3 A Mode Switching Condition

In its full generality, Problem 2 can be posed as a controller synthesis problem for hybrid systems [7,12]. Such synthesis methods involve *nested*, and possibly *cyclic* reachability computations, where each reachability computation involves computing the capture set of a differential game. Furthermore, termination guarantees for controller synthesis methods are rather limited [6].

In our mode switching problem, however, there is enough structure to take advantage of in order to simplify the complexity of the synthesis task. First of all, the continuous controllers are assumed to have been designed, and therefore we do not have to design the continuous part of the system, but simply determine the mode switching conditions. Furthermore, by imposing certain conditions on the allowable mode switches, we reduce the complexity of the synthesis problem, by *maximally decoupling the discrete and continuous aspects of the synthesis.*

To address the problem, we have to characterize the reachable set of each mode and switching condition among them. Let $\phi_i(t, r_i, x_0)$ denote the *flow* of system (2.1) operating in mode q_i with the controller defined by (2.2) for initial condition x_0, and desired output trajectory r_i.

Definition 2 (Predecessor set). *Given a set $P \subseteq X_i$, a trajectory $r_i \in \mathcal{R}_i$, the reach set $Pre_i(P, r_i)$ in mode q_i is defined by*

$$Pre_i(P, r_i) = \{\ x_0 \in X_i \ \mid \ \exists t \geq 0 \ \exists x \in P \ \text{ such that } \ x = \phi_i(t, r_i, x_0)\ \} \quad (3.1)$$

Therefore $Pre_i(P, r_i)$ consists of all states that can reach the set P in mode q_i for a given output trajectory r_i, at *some* future time. Furthermore, because of Assumption 1, we have a guarantee that throughout the whole trajectory, the state constraints are satisfied, that is $\phi_i(t, r_i, x_0) \in X_i$ for all $t \geq 0$.

Given control modes q_i, and q_j, one would typically allow a switch from mode q_i to q_j if during the operation of the system under mode q_i for some $r_i \in \mathcal{R}_i$, the state reaches the allowable set of initial conditions $S_j(r_j)$ for some $r_j \in \mathcal{R}_j$, *i.e.* there exist $r_i \in \mathcal{R}_i$ and $r_j \in \mathcal{R}_j$ such that

$$S_i(r_i) \cap Pre_i(S_j(r_j), r_i) \neq \emptyset. \quad (3.2)$$

If one allows this type of mode switching, then reachability critically depends on the particular choice of initial conditions since some initial conditions in $S_i(r_i)$ may reach the set $S_j(r_j)$ of mode q_j while others may not. If this is the case, then nested reachability computations seem necessary for the solution of the mode switching problem. However, such nested computations can be avoided if one places the following condition on mode switching.

Definition 3 (Consistent mode switching). *Assume that control mode q_i satisfies Assumption 1, that is $\phi_i(t, r_i, x_0) \in X_i$ for all $t \geq 0$ with initial conditions starting from $S_i(r_i)$ where $r_i \in \mathcal{R}_i$. A transition from mode q_i to mode q_j is allowed only if there exist $r_i \in \mathcal{R}_i$ and $r_j \in \mathcal{R}_j$ such that*

$$S_i(r_i) \subseteq Pre_i(S_j(r_j), r_i) \quad (3.3)$$

$$\Leftrightarrow \forall x_0 \in S_i(r_i) \ \exists t \geq 0 \ \exists x \in S_j(r_j) \ \text{ such that } \ x = \phi_i(t, r_i, x_0) \quad (3.4)$$

Therefore, if there exist trajectories r_i (in mode q_i) and r_j (in mode q_j) such that, if the system starts at *any* $x_0 \in S_i(r_i)$, then switching from mode q_i to q_j can occur at some time t such that $\phi_i(t, r_i, x_0) \in S_j(r_j)$. The consistent mode switching condition is shown in Figure 1. The condition expressed in Definition 3

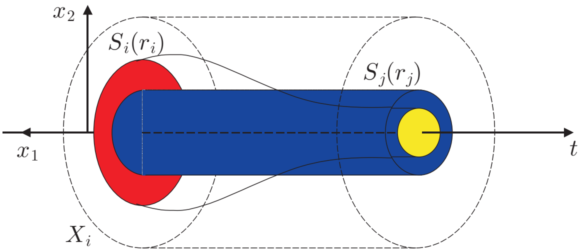

Fig. 1. Visualization of consistent mode switching condition

is a consistency condition that guarantees that our ability to get from mode q_i to mode q_j for the particular trajectory pair (r_i, r_j) is independent of the choice of initial condition in $S_i(r_i)$. The condition is reminiscent of the time-abstract bisimulation property from formal verification [8]. In this case, however, Definition 3 is quite different since no partitioning of the state space is involved. Now define

$$\mathcal{R}^{ij} = \{(r_i, r_j) \in \mathcal{R}_i \times \mathcal{R}_j \mid \text{condition (3.3) is satisfied } \} \tag{3.5}$$

Hence, if $\mathcal{R}^{ij} \neq \emptyset$, then mode switching from q_i to q_j is possible since there exists a trajectory $r_i \in \mathcal{R}_i$ that will steer the system state to an initial set $S_j(r_j)$ with $r_j \in \mathcal{R}_j$ independently of where we start in $S_i(r_i)$. Therefore, *every* trajectory pair $(r_i, r_j) \in \mathcal{R}^{ij}$ will steer the system from mode q_i to mode q_j. For each $(r_i, r_j) \in \mathcal{R}^{ij}$, the only thing that depends on the initial condition is *when* the state will reach $S_j(r_j)$, but not *if* the state will reach $S_j(r_j)$.

To test the mode switching condition (3.3), and compute the sets $\mathcal{R}^{ij}$, one needs to compute the predecessor set $Pre_i(P, r_i)$. Even though there is extensive research in computing exactly, or approximately such reachable sets [7,9,12, 11,13], there is limited research for parametric reachability computations [10]. Furthermore, in our problem we take advantage of the fact that in each control mode, the output is tracking a reference trajectory r_i. Therefore, by designing trajectories we *design part of the reachable space* whereas the part of the state is not reflected in the output remains within the set X_i. Choosing simple, or better computable, classes of trajectories $\mathcal{R}_i$ will allow us to efficiently perform reachability computations for $Pre_i(P, r_i)$ with parameters $r_i \in \mathcal{R}_i$. To continue discussion, we assume that the Pre_i operators are available to us, and defer this important issue to Section 5.

4 Mode Sequence Synthesis

The mode switching condition (3.3) makes the mode switching problem much more tractable since we can ignore the initial sets and focus on the trajectory sets $\mathcal{R}^{ij}$. Furthermore, the construction presented in this section will abstract the mode switching logic into a purely discrete graph. Therefore one can first determine the sequence of modes using standard algorithms for discrete graph reachability, and then determine the continuous parameters r_i for each mode. This will decouple the discrete from the continuous aspects of the problem, and allow continuous techniques for continuous problems, and discrete techniques for discrete problems.

Given a collection of control modes $Q = \{q_1, \ldots, q_N\}$, the first attempt at solving the mode switching would construct a graph as $(Q, \rightarrow)$ where the vertices of the graph would be the set of control modes Q, and we would define the transition relation $\rightarrow \subseteq Q \times Q$ as

$$(q_i, q_j) \in \rightarrow \qquad \Longleftrightarrow \qquad \mathcal{R}^{ij} \neq \emptyset \tag{4.1}$$

In other words, there would be a transition $q_i \rightarrow q_j$, if there exist trajectory pairs $(r_i, r_j) \in \mathcal{R}^{ij}$ that can transfer the system from mode q_i to q_j. This approach, however, leads immediately to problems because if $q_i \rightarrow q_j$ and $q_j \rightarrow q_k$ there may not exists a trajectory r_j, which will take a point $x \in S_i(r_i)$ to $S_k(r_k)$ via $S_j(r_j)$, if $\mathcal{R}^{ij} \cap \mathcal{R}^{jk} = \emptyset$. Hence, transitivity fails, and our mode switching graph is not a *consistent abstraction* as the high level mode switching logic is not implementable at the lower level by the continuous controllers.

In order to obtain a consistent *control mode graph*, denoted as $(Q_c, \rightarrow_c)$, that has feasible low level implementations, our original attempt must be refined. In particular, each control mode q_i gets refined to $2N$ *submodes*, where N submodes stand for entering mode q_i from any other mode q_j, and N more copies for exiting mode q_i towards any other mode q_j. This refinement is illustrated in Figure 2, where mode q_1 has two *submodes*, q_1^{12} which is the operation of the system in mode q_1 on the way to mode q_2, whereas q_1^{21} is the operation of the system under mode q_1 after being in mode q_2. Therefore, this control mode graph has some discrete memory, in the sense that each state represents not only which mode the system is in, but also which mode will either precede it or has preceded it.

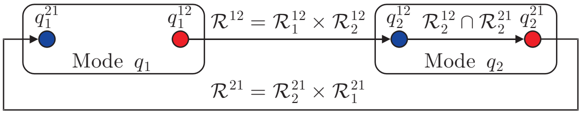
q_1^{21}
q_1^{12}
$\mathcal{R}^{12} = \mathcal{R}_1^{12} \times \mathcal{R}_2^{12}$
q_2^{12}
$\mathcal{R}_2^{12} \cap \mathcal{R}_2^{21}$
q_2^{21}
Mode q_1
Mode q_2
$\mathcal{R}^{21} = \mathcal{R}_2^{21} \times \mathcal{R}_1^{21}$

Fig. 2. Refining the mode switching logic by introducing submodes in order to obtain a consistent control mode graph

The N^2 pairwise reachability computations in order to compute the sets $\mathcal{R}^{ij}$, can immediately be embedded in the graph $(Q_c, \rightarrow_c)$. The computed sets $\mathcal{R}^{ij}$ can be used to go from submode q_i^{ij} to q_j^{ij}. After this initial step, the graph contains only isolated transition pairs between different modes as no transitions between submodes are considered.

If the set $\mathcal{R}^{ij}$ can be expressed as a decoupled product of the form $\mathcal{R}^{ij} = \mathcal{R}_i^{ij} \times \mathcal{R}_j^{ij}$ where $\mathcal{R}_i^{ij} = \{r_i \in \mathcal{R}_i \mid (r_i, r_j) \in \mathcal{R}^{ij}\}$ and $\mathcal{R}_j^{ij} = \{r_j \in \mathcal{R}_j \mid (r_i, r_j) \in \mathcal{R}^{ij}\}$, then the choice of trajectory $r_i \in \mathcal{R}_i^{ij}$ in mode q_i would work for any trajectory $r_j \in \mathcal{R}_j^{ij}$ in mode q_j, *i.e.*

$$\forall r_i \in \mathcal{R}_i^{ij}\ \forall r_j \in \mathcal{R}_j^{ij} \qquad \text{condition (3.3) is satisfied.}$$

This decoupling allows us to consider switching via submodes. In Figure 2, if $\mathcal{R}_2^{12} \cap \mathcal{R}_2^{21}$ is non empty, then that means that there exists a trajectory r_2 which is common for both submodes. Notice that in this case, we do not have to do any reachability computations, we simply have to compute intersections of trajectory sets. Therefore, within each mode, we can check for submode consistency by simply performing set intersections. Since there are maximally $2N$ submodes of N modes, a total of $N(N)^2 = N^3$ intesections must be computed. We now summarize the ideas and present an algorithm for constructing the consistent control mode garph. The algorithm starts with the pairwise reachability computations (3.3,3.5), and performs the submode interconnections.

```
**Algorithm 1 : (Consistent Control Mode Graph)**
**Input** Control Modes $Q = \{q_1, \ldots, q_N\}$
**Output** Control Mode Graph $(Q_c, \rightarrow_c)$
**Initialize** $Q_c := \emptyset$, $\rightarrow_c = \emptyset$

**Determine Mode Interconnections**
for $i = 1 : N$
    for $j = 1 : N$
        Compute sets $\mathcal{R}^{ij}$ using (3.3) and (3.5)
        if $\mathcal{R}^{ij} = \mathcal{R}_i^{ij} \times \mathcal{R}_j^{ij}$;
        $q_i^{ij} := q_i,\ q_j^{ij} := q_j$,
        $Q_c := Q_c \cup \{q_i^{ij}, q_j^{ij}\}$,
        $\rightarrow_c := \rightarrow_c \cup \{(q_i^{ij}, q_j^{ij})\}$
        end if
    end for
end for

**Determine Submode Interconnections**
for $j = 1 : N$
    for all $q_j^{ij} \in \{q_j^{nj} \in Q_c | \exists n \ \ s.t.\ \ (q_n^{nj}, q_j^{nj}) \in \rightarrow_c\}$
        for all $q_j^{jk} \in \{q_j^{jm} \in Q_c | \exists m \ \ s.t.\ \ (q_j^{jm}, q_k^{jm}) \in \rightarrow_c\}$
            if $\mathcal{R}_j^{ij} \cap \mathcal{R}_j^{jk} \neq \emptyset$;
            $\rightarrow_c := \rightarrow_c \cup \{(q_j^{ij}, q_j^{jk})\}$
            end if
        end for
    end for
end for
```

Overall, Algorithm 1 requires N^2 reachability computations for the mode interconnections, and N^3 set intersections for the submode interconnections. After applying Algorithm 1, we obtain a finite *control mode graph* $(Q_c, \rightarrow_c)$ which, as the following proposition shows, is *consistent*.

Proposition 1. *For any* $j \in \{1, \ldots, |Q|\}$, *if* $\exists q_j^{ij} \in \{q_j^{nj} \in Q_c | \; \exists n$ *such that* $(q_n^{nj}, q_j^{nj}) \in \rightarrow_c\}$, $\exists q_j^{jk} \in \{q_j^{jm} \in Q_c | \; \exists m$ *such that* $(q_j^{jm}, q_m^{jm}) \in \rightarrow_c\}$ *and* $\mathcal{R}_j^{ij} \cap \mathcal{R}_j^{jk} \neq \emptyset$, *then there exists* $r_i \in \mathcal{R}_i^{ij}$, $r_j \in \mathcal{R}_j^{ij} \cap \mathcal{R}_j^{jk}$ *and* $r_k \in \mathcal{R}_k^{jk}$ *such that*

$$S_i(r_i) \subseteq Pre_i(Pre_j(S_k, r_k), r_j).$$

Proof: Given $(q_i^{ij}, q_j^{ij}) \in \rightarrow_c$, we can pick any $r_i \in \mathcal{R}_i^{ij}$ and since $\mathcal{R}_j^{ij} \cap \mathcal{R}_j^{jk} \neq \emptyset$ we can pick any $r_j \in \mathcal{R}_j^{ij} \cap \mathcal{R}_j^{jk}$, so that $\forall x_0 \in S_i(r_i) \; \exists t \geq 0 \; \exists x \in S_j(r_j)$ such that $x = \phi_i(t, r_i, x_0)$. Then, pick any $r_k \in \mathcal{R}_k^{jk}$, since $(q_j^{jk}, q_j^{jk}) \in \rightarrow_c$, $r_j \in \mathcal{R}_j^{ij} \cap \mathcal{R}_j^{jk}$ and the switching occurs whenever $\phi_i(t, r_i, x_0) \in S_j(r_j)$, it can be easily seen that $\exists s \geq 0 \; \exists y \in S_k(r_k)$ *such that* $y = \phi_j(s, r_j, \phi_i(t, r_i, x_0)) = \phi_j(s, r_j, x)$. The choice on the trajectories is illustrated in Figure 3. Since by Assumption 1, $\phi_i(t, \cdot, \cdot) \in X_i$ and $\phi_j(s, \cdot, \cdot) \in X_j$ for the choice of initial conditions and reference trajectories, by directly applying the definition we have shown the result. □

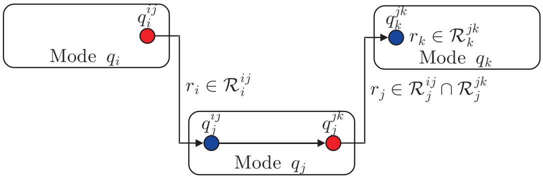
q_i^{ij}
Mode q_i
$r_i \in \mathcal{R}_i^{ij}$
q_k^{jk}
$r_k \in \mathcal{R}_k^{jk}$
Mode q_k
$r_j \in \mathcal{R}_j^{ij} \cap \mathcal{R}_j^{jk}$
q_j^{ij}
q_j^{jk}
Mode q_j

Fig. 3. Graphical illustration of feasible trajectories between control modes.

Without loss of generality, in the following discussion, we assume that the given initial and final control mode in Q can be represented by $q_S \in Q_c$ and $q_F \in Q_c$ respectively. Given an initial control mode $q_S \in Q_c$, the problem of whether we can reach control mode $q_F \in Q_c$, can be efficiently solved using standard reachability algorithms. Furthermore, one can determine the shortest path (minimum number of mode switches) between mode q_S and q_F, in the control mode graph. The structure that we have imposed on our control mode graph, immediately results in the following solution to the mode switching problem.

Theorem 1 (Mode Switching Solution). *Given a collection of control modes* Q, *consider the mode switching Problem 2. Construct the consistent control mode graph* $(Q_c, \rightarrow_c)$ *as described in Algorithm 1. If there exists a path in the consistent control mode graph between* q_S *and* q_F *with feasible trajectories* r_S *and* r_F, *then Problem 2 is solvable.*

Having determined the sequence of modes that can steer our system from q_S to q_F, we are left with the problem of determining the parameters r_i for each mode of the sequence. By construction, such parameters exist and may be selected from the computed sets. Furthermore, it is reasonable to pose the problem of choosing r_i within mode q_i as an optimization or an optimal control problem. A key issue for this approach (as well as for *most* controller synthesis approaches for hybrid systems), is to be able to compute $Pre_i(S_j, r_i)$ in order to check condition 3. This is the focus of the following section of this paper.

5 Reachability Computations

There has been a growing interest recently in computing reachable sets for various classes of systems [9,11,7,13]. In particular, the approach of [9] has been extended to classes of parametric linear control systems [10], which is highly relevant for computing the operator (3.1).

In our case, however, the continuous dynamics are those of output-tracking, closed-loop systems. Therefore part of the state is forced to converge to a trajectory that we get to design, and part of the trajectory is guaranteed to satisfy state constraints. This gives us the opportunity to obtain very reasonable approximations of the reachable sets, and even *design reachable sets* by appropriately designing output trajectories. The following example illustrates how continuous controller design results in reachability computations which are very easy to check.

Example 2. **Multi-Modal Control of a Helicopter Model(Continued)** Reconsider the four control modes shown in Example 1. We first present the controller design to illustrate how to compute the reachable sets, then we show how to check the consistent mode switching condition between control modes. In this example, we assume that all output trajectories are *constant* trajectories, therefore, all controllers are setpoint regulators. Choosing computable classes of trajectories makes the reachability computations simpler.

Given the specifications for the control modes, a nonlinear control scheme [5] based on outer flatness is applied for the design of the controllers. For each mode, the closed-loop dynamics with states defined by $x_{ex} = [p_x, \dot{p}_x, p_z, \dot{p}_z, \theta, \dot{\theta}, T_M, a_M]^T \in \mathbb{R}^8$ can be decoupled into an inner system and two outer subsystems which specify the dynamics in x and z directions. In the following presentation, the Hover mode is presented to illustrate how the reachable set can be computed.

For q_1, the output tracking controller is designed such that $y_1(t)$ shall track $r_1 = [r_{1x}, r_{1z}]^T$ and the output tracking error is uniformly ultimately bounded. Furthermore, because of satisfying Assumption 1, the controller is designed with initial set $S_1(r_1) = B^1([r_{1x}, 0]^T, \epsilon_{1x}) \times B([r_{1z}, 0]^T, \epsilon_{1z}) \times S_{in}$ where $r_1 \in \mathcal{R}_1 = \mathbb{R}^2$, $\epsilon_{1x}, \epsilon_{1z} > 0$ and $S_{in} \subseteq (-\pi/2, \pi/2) \times \mathbb{R}^3$ such that for $x(t_0) \in S_1(r_1)$ then

$$\begin{cases} \|e_{1x}(t)\| \leq M_{1x} \exp(-\alpha_{1x} t)(\|e_{1x}(t_0)\| + \delta_{1inx}), \\ \|e_{1z}(t)\| \leq M_{1z} \exp(-\alpha_{1z} t)(\|e_{1z}(t_0)\| + \delta_{1inz}), \\ x_{in} \in X_{in}, \ \forall t_0 \leq t < t_0 + T_1; \end{cases} \text{and} \begin{cases} \|e_{1x}(t)\| \leq \delta_{1x}, \\ \|e_{1z}(t)\| \leq \delta_{1z}, \\ x_{in} \in S_{in}, \ \forall t \geq t_0 + T_1 \end{cases} \tag{5.1}$$

[1] $B(r, \epsilon) = \{\eta | \ \|\eta - r\| < \epsilon\}$.

for some T_1 M_{1x}, M_{1z}, α_{1x}, α_{1z}, δ_{1inx}, δ_{1inz}, δ_{1x}, $\delta_{1z} > 0$. In above, $e_{1x} = [p_x - r_{1x}, \dot{p}_x]^T$, $e_{1z} = [p_z - r_{1z}, \dot{p}_z]^T$, and $x_{in} = [\theta, \dot{\theta}, T_M, a_M]^T$. Equation (5.1) explicitly over specifies the reachable set of the mode q_1 by examing the stability property. For other modes, although the control designs are slightly modified for tracking different outputs, the reachable sets of other modes are similarly computed. In Figure 4, we show the inital sets of all the control modes by projecting them onto $p_x - \dot{p}_x$ plane and $p_z - \dot{p}_z$ plane where the projection operator is defined as $\Pi_i : x_{ex} \mapsto (p_i, \dot{p}_i)$ for $i \in \{x, z\}$. In summary, the control modes can be specified by

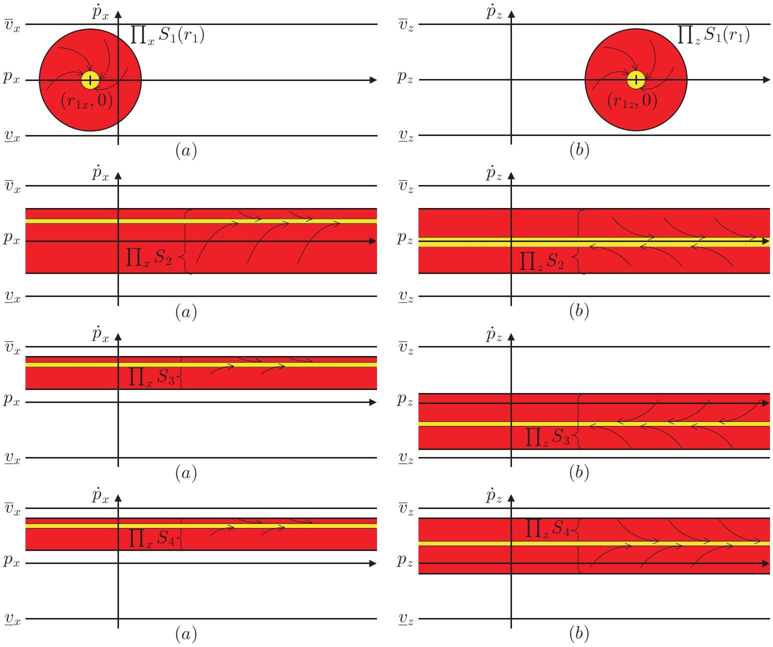

Fig. 4. Projection of $S_1(r_1)$, $S_2(r_2)$, $S_3(r_3)$, and $S_4(r_4)$ onto: (a) $p_x - \dot{p}_x$ plane; (b) $p_z - \dot{p}_z$ plane

Control Mode	Trajectory Set	Initial Set
q_1	$\mathcal{R}_1 = \mathbb{R}^2$	$S_1 = B(0,4) \times B(0,4) \times S_{in}$
q_2	$\mathcal{R}_2 = [-3,3] \times \{0\}$	$S_2 = \mathbb{R} \times (-3.5, 3.5) \times (-3.5, 3.5) \times S_{in}$
q_3	$\mathcal{R}_3 = [2,4] \times [-3,0]$	$S_3 = \mathbb{R} \times (1.5, 4.5) \times (-3.5, 0.5) \times S_{in}$
q_4	$\mathcal{R}_4 = [2,4] \times [0,3]$	$S_4 = \mathbb{R} \times (1.5, 4.5) \times (-0.5, 3.5) \times S_{in}$

where $S_{in} = B(0, 0.2)$, $X_{in} = (-\pi/2, \pi/2) \times \mathbb{R}^3$ and the associated parameters are defined as $\underline{v}_x = -6$, $\underline{v}^{cr} = 1$, $\bar{v}_x = 6$, $\underline{v}_z = -6$, $\bar{v}_z^{as} = 1$, $\underline{v}_z^{de} = -1$, $\bar{v}_z = 6$.

Given the set of control modes, we generated the consistent control mode graph by applying Algorithm 1. In Figure 5, we illustrate the idea of computing the reachable sets on $p_x - \dot{p}_x$ plan. One can easily see the advantage of using feedback, since it is straight forward not only to check the consistent mode switching condition but also to determine the feasible range of trajectory, that is compute the sets $\mathcal{R}^{ij}$. In particular consider the pair (q_1, q_2), that is the transition from hover to cruise. As can be seen from the left side of Figure 5, the consistency condition is trivially satisfied since the ball $S_1(r_1)$ will *eventually* shrink towards the setpoint $(r_{1x}, 0)$, and as a result, will be totally contained inside $S_2(r_2)$ for any r_2. Therefore, in this case $\mathcal{R}^{12} = \mathcal{R}_1 \times \mathcal{R}_2$. Therefore, feedback allows us to check very easily the consistency condition and compute the sets $\mathcal{R}^{ij}$. The right side of Figure 5 shows the similar graphical computation for the mode transition (q_2, q_3), from cruise to ascend. In a similar manner, we have checked the following pairs,

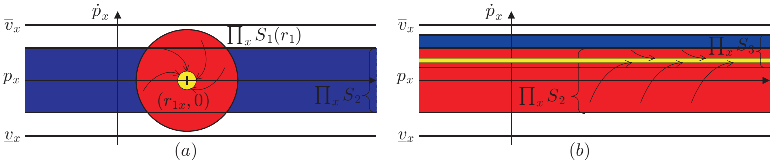

Fig. 5. Graphical illustration of performing reachability computation for checking consistent mode switching condition on $p_x - \dot{p}_x$ plane: (a) $q_1 \to q_2$; (b) $q_2 \to q_3$

$$\{(q_1, q_2), (q_2, q_2), (q_2, q_3), (q_2, q_4), (q_3, q_2), (q_3, q_3), (q_3, q_4), (q_4, q_2), (q_4, q_3), (q_4, q_4)\}$$

All of the above reachability computations were extremely simple to check. The result of applying Algorithm 1 is summarized in the control mode graph that is shown in Figure 2.

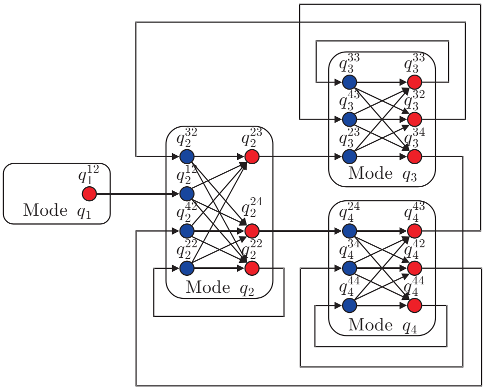

Fig. 6. Consistent control mode graph for the multi-modal helicopter control example

Recall the *high-altitude takeoff* task, which is the task of having the Hover mode q_1 as an initial mode and ask for a finite control mode sequence to reach the Ascend mode q_3. We can now see that from Figure 2 that q_1 has $\{q_1^{12}\}$ as a submode, and $q_3 = \{q_3^{23}, q_3^{43}, q_3^{32}, q_3^{34}\}$, and there exit many paths which are feasible for achieving the task. However, $q_1^{12}q_2^{12}q_2^{23}q_3^{23}$ gives a solution to the task with the minimum number of mode switches, *i.e.*, $q_1 \to q_2 \to q_3$. Given a cost function with respect to the continuous variables, the performance of the sequence can now be optimized with respect to the feasible trajectories. We have therefore decoupled the problem in a purely discrete graph search problem, and a collection of continuous designs within each mode.

Simulation results of the controlled system based on the selected sequence are shown in Figure 2. In the simulation, we can choose $r_1 = [0\ 0]^T \in \mathcal{R}_1^{12}$, $r_2 = [2\ 0]^T \in \mathcal{R}_2^{12} \cap \mathcal{R}_2^{23}$ and $r_3 = [3\ -1]^T \in \mathcal{R}_3^{23}$. The initial conditions of the outer system are $p_x(0) = -2$, $\dot{p}_x(0) = -0.2$, $p_z(0) = 1$, $\dot{p}_z(0) = 0.5$. The initial condition of the inner system, $x_{in}(0) \in S_{in}$. Mode switchings occur at $t = 20$ for $q_1 \to q_2$ and at $t = 45$ for $q_2 \to q_3$.

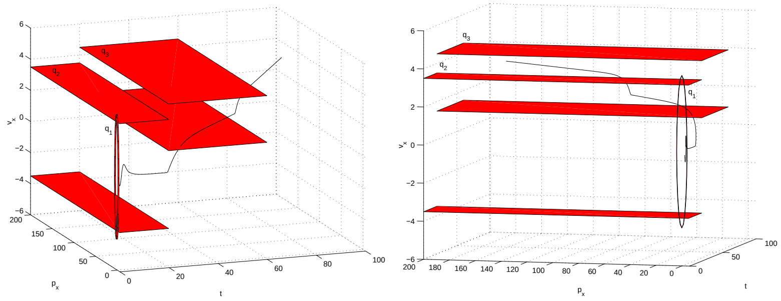

Fig. 7. Projected trajectories of the helicopter along with the initial sets of the next control modes from different view angles are shown. Notice that immediate transition $q_2 \to q_3$ after $q_1 \to q_2$ is not allowed until $x(t)$ enters the initial set $S_3(r_3)$.

6 Conclusion

In this paper, we have considered the mode switching problem among a collection of output tracking controllers for nonlinear systems. Our approach consists of extracting a finite graph which refines the original collections of modes, but is consistent with the physical system. Extracting a finite graph critically depends on the fact the closed loop, output tracking controllers reduce the complexity of the model to the complexity of the output trajectories.

Even though, our framework reduces the continuous complexity so that many of the computations can be done by hand, obtaining a consistent mode graph for a large scale helicopter or aircraft (a Boeing 747 has approximately 500 modes) will clearly require the development of a computational tool. Such a mode switching tool can be used off-line for synthesizing the mode switching logic every time a new mode is designed. The control mode graph can then be used on-line for efficient and dependable real-time mode switching.

Acknowledgments. The authors would like to thank Paulo Tabuada for extremely intense discussions on the subject. This work is partially supported by the DARPA SEC grant, F33615-98-C-3614, DARPA MoBIES grant F33615-00-C-1707, DARPA JFACC grant N66001-99-C-8510, and the University of Pennsylvania Research Foundation.

References

1. A. Balluchi, L. Benvenuti, M.D. Di Benedetto, C. Pinello, and A.L. Sangiovanni-Vincentelli, Automotive engine control and hybrid systems: challenges and opportunities, *Proceedings of the IEEE*, 88(7):888 -912, July 2000.
2. T. J. Koo, S. Sastry. Output Tracking Control Design of a Helicopter Model Based on Approximate Linearization. In *Proceedings of the 37th Conference on Decision and Control*, pp.3635-40, Tampa, Florida, December 1998.
3. T. J. Koo, B. Sinopoli, A. Sangiovanni-Vincentelli, and S. Sastry. A Formal Approach to Reactive System Design: A UAV Flight Management System Design Example. In *Proceedings of IEEE International symposium on Computer-Aided Control System Design* Kohala Coast, Hawaii. September 1999.
4. J. Liu, X. Liu, T. J. Koo, B. Sinopoli, S. S. Sastry, and E. A. Lee. Hierarchical Hybrid System Simulation. In *Proceedings of the 38th Conference on Decision and Control*, Phoenix, Arizona. December 1999.
5. O. Shakernia, Y. Ma, T. J. Koo, and S. Sastry, *Landing an Umanned Air Vehicle: Vision Based Motion Estimation and Nonlinear Control*, Asian Journal of Control, Vol. 1, No.3, pp. 128-145, September 1999.
6. Omid Shakernia, George J. Pappas, Shankar Sastry, *Decidable Controller Synthesis for Classes of Linear Systems*, Hybrid Systems : Computation and Control, Lecture Notes in Computer Science, volume 1790, pages 407-420, 2000.
7. J. Lygeros, C. Tomlin, S. Sastry. Controllers for Reachability Specifications for Hybrid Systems, Automatica, Volume 35, Number 3, March 1999.
8. R. Milner. Communication and Concurrency, Prentice Hall, 1989.
9. G. Lafferriere, G.J. Pappas, S. Yovine. Reachability Computation for Linear Hybrid Systems, In Proceedings of the 14th IFAC World Congress, volume E, pages 7-12, Beijing, 1999.
10. G. Lafferriere, G.J. Pappas, and S. Yovine. Symbolic Reachability Computations for Families of Linear Vector Fields, Journal of Symbolic Computation, To appear.
11. A.B. Kurzhanski, P.Varaiya, Ellipsoidal Techniques for Reachability Analysis, Hybrid Systems : Computation and Control, Lecture Notes in Computer Science, 2000.
12. E. Asarin, O. Bournez, T. Dang, O. Maler, and A. Pnueli, Effective Synthesis of Switching Controllers for Linear Systems, Proceedings of the IEEE, 88(2):1011-1025.
13. A. Chutinan, B.H. Krogh, Verification of polyhedral-invariant hybrid systems using polygonal flow pipe approximations, Hybrid Systems : Computation and Control, Lecture Notes in Computer Science, 1999.
14. I. Kolmanovsky, E. G. Gilbert. Multimode Regulators for Systems with State & Control Constraints and Disturbance Inputs. In *Lecture Notes in Control and Information Sciences 222, Control Using Logic-Based Switching*, A. Stephen Morse(Ed.),pp. 104-117, Springer-Verlag, London, 1997.
15. S. Sastry. *Nonlinear Systems: Analysis, Stability, and Control.* Springer-Verlag, New York, 1999.