Proceedings of the 40th IEEE
Conference on Decision and Control
Orlando, Florida USA, December 2001

Multi-Modal Control of Systems with Constraints

T. John Koo
Department of EECS
University of California
Berkeley, CA 94720
koo@eecs.berkeley.edu

George J. Pappas
Department of EE
University of Pennsylvania
Philadelphia, PA 19104
pappasg@ee.upenn.edu

Shankar Sastry
Department of EECS
University of California
Berkeley, CA 94720
sastry@eecs.berkeley.edu

Abstract

In multi-modal control paradigm, a set of controllers of satisfactory performance have already been designed and must be used. Each controller may be designed for a different set of outputs in order to meet the given performance objectives and system constraints. 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 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 robot manipulator example, where we determine the mode switching logic that achieves the given reachability task.

1 Introduction

The multi-objective nature of complex control systems such as automated highway systems, air traffic management systems[15], and unmanned aerial vehicles[9] requires the use of a suite of controllers instead of using a single controller. In this multi-modal control paradigm, a set of controllers of satisfactory performance have already been designed and must be used. Each controller may be designed for a different set of outputs in order to meet the given performance objectives and system constraints. 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.

Multi-modal control has been studied especially in the context of stability and safety; see [4] for stability results of switching between stable linear time-invariant (LTI) controllers, [7] for *safe* switching conditions for systems with pointwise-in-time constraints on state and control, and [12, 2] for controller designs and switching conditions for satisfying multiple objectives such as safety property and optimal performance.

In [8], we have proposed a framework for the synthesis of mode switching for reachability specifications. As opposed to [4], we are interested in reachability specifications rather than preserving stability of a global equilibrium point. The stability of each control mode is assumed to be preserved with respect to the state variables of interest and taken care of by the control design. Switching condition between modes defined in [8] is a generalization of the the result presented in [7] which requires the controllers to be designed in a total order with respect to safety specifications. The outcome of applying our algorithm results in a partial ordering of the given control modes, and not only can captures the outcome generated as shown in [7] but also is more expressive since it captures more possible switching combinations.

In this paper, 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 while simultaneously avoiding violation of specified (state or input) constraints. 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 the given reachability tasks. Hence, the mode switching problem can be posed as:

Problem 1.1 *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.*

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, as shown in [8], we place a *consistency* condition in our mode switching logic which is reminiscent of the notion of *bisimulation* [13]. 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

In this section, we introduce a concept of control mode, and precisely define Problem 1.1 as a *mode switching problem*. First, consider a nonlinear system modeled by differential equations of the form

$$\begin{aligned} \dot{x}(t) &= f(x(t)) + g(x(t))u(t) \\ x(t_0) &= x_0, \quad t \geq t_0 \end{aligned} \tag{1}$$

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. Each control mode corresponds to a output tracking controller applying to the nonlinear system (1). We now define a concept of control mode.

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

$$u(t) = k_i(x(t), r_i(t)) \tag{2}$$

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 $i \in \{1, \ldots, N\}$. 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 (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.

State constraints are specified by X_i for $i = 1, \ldots, N$. The state constraints are introduced due to the physical limits of the system and the control design. Since the controller is static, input constraints can be incorporated as state constraints for the the nonlinear system (1). For example, a control constraint $\|u(t)\|_\infty \leq \bar{u}$ implies $X_i = \{x \in \mathbb{R}^n | \; \forall r_i \in \mathcal{R}_1 \;\; k_i(x, r_i) \leq \bar{u}\}$. For dynamic controller with fixed structure, it has been shown in [7] that the input constraints can be similarly considered as state constraints in which both system and control states are considered.

Clearly, more general definitions of control mode can be defined. For example, one can define a mode as the operation of the system under an open loop control, or one can consider more general classes of trajectories, or one can incorporate system disturbance under modes of operation. Such generalizations will be considered in future research. 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, $i \in \{1, \ldots, N\}$, we assume that a controller of the form (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 (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 [5, 16]. Based on different design methodologies, the notion of *output tracking* could be different (uniform, asymptotic, exponential, etc.)

Example 2.2 Point Mass. *Consider the dynamics of a point mass that can be modeled as a double chain of integrators,*

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= u \end{aligned} \tag{3}$$

where $x_1, x_2, u \in \mathbb{R}$ represent the position, velocity and acceleration of the point mass, respectively. Define the state as $x = [x_1 \; x_2]^T \in \mathbb{R}^2$. Assume that there are two controllers designed for (3) and the corresponding control modes are specified as:

Mode	*Output*	*Reference*	*Constraint*
q_1	$y_1 = x_1$	r_1	X_1
q_2	$y_2 = x_1$	r_2	X_2

where $X_2 \subset X_1 = \mathbb{R} \times (\underline{v}, \overline{v})$ with $\underline{v} < 0 < \overline{v}$. The given controllers are linear controllers and the design is simply based on pole-placement. In order to satisfy Assumption 1, the controller in control mode q_1 is designed such that $\mathcal{R}_1 \subseteq \mathbb{R}^n$ and $S_1(r_1) = B([r_1 \; 0]^T, \delta_1)$ with $\delta_1 < \frac{\min(\underline{v}, \overline{v})}{M_1}$ where M_1 is the overshoot constant which can easily be obtained by Lyapunov theorems. Similarly, control mode q_2 are similarly defined but the poles *are placed differently. It results in faster response but larger overshoot, i.e.* $M_2 > M_1$. *Hence,* $\mathcal{R}_2 \subseteq \mathbb{R}^n$ *and* $S_2(r_2) = B([r_2\ 0]^T, \delta_2)$ *with* $\delta_2 < \frac{\min(\underline{v},\overline{v})}{M_2}$. *Therefore, we have* $S_2(r_2) \subset S_1(r_1)$ *if* $r_1 = r_2$.

Given two control modes as shown in Example 2.2, one cannot simply switch from one control mode to another due to incompatible constraints and trajectories. One can easily complicate the situation by introducing many more control modes to serve different performance objectives. A natural question is then whether this mode reachability task as defined Problem 1.1 can be achieved by a *finite sequence* of modes. Based on the discussion, we can now define the mode switching problem that we will address in this paper.

Problem 2.3 (Mode Switching Problem) *Given an initial control mode* q_0 *with desired reference* r_0*, 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* $q_0 \rightarrow \cdots q_i \rightarrow q_j \cdots \rightarrow q_F$ *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 2.2, one can define a task of having mode q_1 as an initial mode and ask for a finite control mode sequence to reach mode q_2. Note that Problem 2.3 is a reachability problem. In this simple example, the problem can be solved by examining the mode switching condition between modes.

In the above mode switching problem, 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.*

3 A Mode Switching Condition

Consider a mode switch from mode q_i to mode q_j. A mode switch from mode q_i to q_j could be allowed 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

$$\exists x_0 \in S_i(r_i)\ \exists t \geq 0\ \exists x \in S_j(r_j) \text{ s.t. } x = \phi_i(t, r_i, x_0) \tag{4}$$

where $\phi_i(t, r_i, x_0)$ denotes the *flow* of system (1) operating in mode q_i with the controller defined by (2) for initial condition x_0, and desired output trajectory r_i. 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.

In general, nested reachability computations are required for solving such a reachability problem. Furthermore, since loops can be considered in feasible sequences, the number of possible mode sequences could be infinite. Therefore, for this type of mode switching, decidability becomes a central issue [1]. We now characterize the reachable set within each mode.

Definition 3.1 (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*

$$Pre_i(P, r_i) = \{x_0 \in X_i |\ \exists t \geq 0\ \exists x \in P \\ s.t.\ x = \phi_i(t, r_i, x_0)\} \tag{5}$$

$Pre_i(P, r_i)$ 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 $t \geq 0$. Hence, using (5), condition (4) can be rewritten as

$$S_i(r_i) \cap Pre_i(S_j(r_j), r_i) \neq \emptyset. \tag{6}$$

In order to avoid the nested computations mentioned above, as well as break free of restricted decidability results, we constrain our allowable mode switches.

Definition 3.2 (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*

$$S_i(r_i) \subseteq Pre_i(S_j(r_j), r_i) \tag{7}$$

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 3.2 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 $S_i(r_i)$. Hence, a mode switching from mode q_i to mode q_j is possible, if 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 $S_i(r_i)$. The collection of the trajectory pair (r_i, r_j) is specified by the following definition

$$\mathcal{R}^{ij} = \{(r_i, r_j) \in \mathcal{R}_i \times \mathcal{R}_j \mid \text{Cond. (7) is satisfied}\} \tag{8}$$

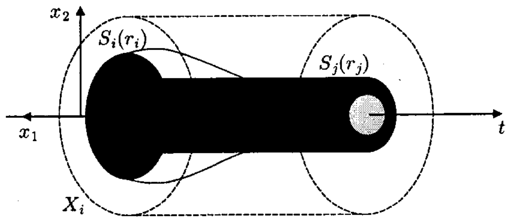

Figure 1: Consistent mode switching condition

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 $S_j(r_j)$.

In our problem we apply the existing results in computing the reachable sets [2, 3, 10, 11, 12] to test the mode switching condition (7) and compute the sets $\mathcal{R}^{ij}$. Since most of these reachability computations are approximate, one must consider an over-approximation of $S_i(r_i)$ and an under-approximation of $Pre_i(S_j(r_j), r_i)$, in order to satisfy condition (7), that is

$$\overline{S}_i(r_i) \subseteq \underline{Pre}_i(\underline{S}_j(r_j), r_i) \tag{9}$$

where $\overline{P}$ and $\underline{P}$ denote the over-approximation and under-approximation of a set P, respectively.

4 Mode Sequence Synthesis

By focusing on the trajectory sets $\mathcal{R}^{ij}$ rather than the initial sets, the mode switching condition (7) makes the mode switching problem much more tractable. 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.

Consider a collection of control modes $Q = \{q_1, \ldots, q_N\}$. 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, there would be a transition $q_i \rightarrow q_j$. However, given $q_i \rightarrow q_j$ and $q_j \rightarrow q_k$, if $\mathcal{R}^{ij} \cap \mathcal{R}^{jk} = \emptyset$ there does not exist a trajectory r_j, which will take a point $x \in S_i(r_i)$ to $S_k(r_k)$ via $S_j(r_j)$. In order to construct a consistent *control mode graph* such that the high level mode switching logic is implementable at the lower level by the continuous controllers, transitivity should be preserved.

As shown in [8], 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. 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. 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.*

$$\forall r_i \in \mathcal{R}_i^{ij} \; \forall r_j \in \mathcal{R}_j^{ij} \quad \text{condition (7) is satisfied.} \tag{10}$$

This decoupling allows us to consider switching via submodes. 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^2 pairwise reachability computations and $N(N)^2 = N^3$ intersections must be computed.

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 (7) and (8)
    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 $j = 1:N$

  $\check{Q} := \{q_j^{nj} \in Q_c \mid \exists n \; s.t. \; (q_n^{nj}, q_j^{nj}) \in \rightarrow_c\}$
  $\hat{Q} := \{q_j^{jm} \in Q_c \mid \exists m \; s.t. \; (q_j^{jm}, q_k^{jm}) \in \rightarrow_c\}$
  for all $q_j^{ij} \in \check{Q}$; **for all** $q_j^{jk} \in \hat{Q}$
    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

We now summarize the ideas and present an algorithm for constructing the consistent control mode graph. The algorithm starts with the pairwise reachability computations (7,8), and performs the submode interconnections. After applying the algorithm, we obtain a finite *control mode graph* $(Q_c, \rightarrow_c)$ which has been shown in [8] to be *consistent*. Without loss of generality, in the following discussion, we assume that the given initial and final control mode in Q can be represented by $q_0 \in Q_c$ and $q_F \in Q_c$ respectively. Given an initial control mode $q_0 \in Q_c$, 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 4.1 (Mode Switching Solution) *Given a collection of control modes Q, consider the mode switching Problem 2.3. 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_0 and r_0, then Problem 2.3 is solvable.*

Having determined the sequence of modes that can steer our system from q_0 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.

5 Multi-Modal Control of Robot Manipulator

The design framework presented in previous sections has been applied to the control of a robot manipulator. The manipulator is a two revolute jointed robot moving on a horizontal plane. The dynamics of the two-link robot can be written as:

$$M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} = u \tag{11}$$

where $\theta = [\theta_1 \ \theta_2]^T \in \mathbb{R}^2$ is the set of configuration variables for the robot and $u = [u_1 \ u_2]^T \in \mathbb{R}^2$ denotes the torques applied at the joints. Furthermore, we are given a joint trajectory $\theta_d(t)$ which we wish to track. For the given robot, the entries of the matrices $M(\theta)$ and $C(\theta, \dot{\theta})$ can be found in [6].

It has been shown that in [16], a proportional plus derivative (PD) control law gives global asymptotic setpoint stabilization, *i.e.* $\dot{\theta}_d \equiv 0$. In its simplest form, a PD control law has the from

$$u = -K_v\dot{e} - K_p e \tag{12}$$

where K_v and K_p are positive definite matrices and $e = \theta - \theta_d$. For tracking a given trajectory $\theta_d(t)$, computed torque control law which has the form

$$u = M(\theta)(\ddot{\theta}_d - K_v\dot{e} - K_p e) + C(\theta, \dot{\theta})\dot{\theta} \tag{13}$$

is widely used. It can be shown that the control law results in global exponential trajectory tracking. However, in the presence of state constraints, in order to satisfy Assumption 1 trajectory tracking can be guaranteed only if the initial tracking error is sufficiently small.

Consider the robot is requested to move from one setpoint, θ_d^A, to another setpoint θ_d^B. There exists velocity constraint for each joint. Define $x = [\theta_1 \ \theta_2 \ \dot{\theta}_1 \ \dot{\theta}_2]^T \in \mathbb{R}^4$ For each setpoint, three PD controllers are designed for different design specifications. Define q_1, q_2, q_3 as the associated control modes for setpoint θ_d^A and q_6, q_7, q_8 as the associated control modes for setpoint θ_d^B.

For moving from θ_d^A to θ_d^B, two different controllers, which are based on computed torque control law, along with the trajectories are designed. Control mode q_4 can perform the task with minimum energy while control mode q_5 is designed for minimum time criterion. Therefore, we have $|Q| = 8$. To simplify the discussion, we further assume that each $\mathcal{R}_i$ is a singleton, for $i = 1, \ldots, 8$. By applying condition (9), the switching conditions between control modes can be easily checked by examining the stability properties of the closed loop systems. In general, controller with faster response will have smaller region of attraction while robust controller has wider region of attraction but worse tracking capability. Since each trajectory contains only one element, it makes the checking of submode connection extremely simply. After applying Algorithm 1, we obtain the consistent control mode graph as shown in Figure 2.

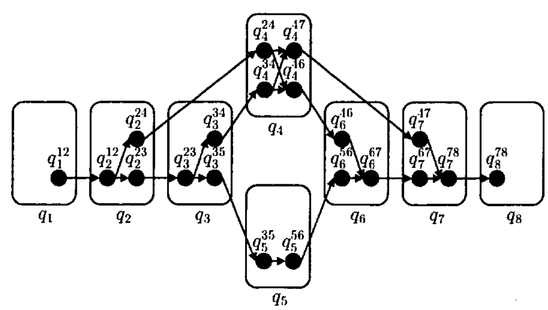

Figure 2: Consistent control mode graph for the multi-modal control of robot manipulator

Assume that the task to be performed by the set of controllers is specified as a reachability task for moving from θ_d^A to θ_d^B. It is also desirable to perform the task with wide region of attraction and small tracking error. This task specification can be translated into the mode switching problem with q_1 as the staring mode and q_8 as the ending one. To check if the reachability task can be achieved with the given set of control modes, we can perform reachability algorithm on the control mode graph. There exists more than one path to achieve the given task specifications. Based on additional performance criterion, a specific path can be chosen to execute the task. For example, to achieve the task in minimum time, the resulting switching sequence would be $q_1q_2q_3q_5q_6q_7q_8$.

6 Conclusion

In this paper, we have considered the mode switching problem among a collection of output tracking controllers for constrained systems. Our approach consists of extracting a finite graph which refines the original collections of modes, but is consistent with the physical system, in the sense that high level design has feasible implementation. 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. Given an initial mode and a final mode, if there exist any path connecting the two modes, it shows that the given task is feasible to be solved. Furthermore, each pat on the graph basically encodes a specific set of performance criteria. By choosing different paths, different performance can be achieved for executing the same task. We have used the robot manipulator example how the multiple objectives can be achieved.

Acknowledgments

The authors would like to thank Paulo Tabuada for various discussions on the subject. This work is partially supported by the DARPA SEC grant F33615-98-C-3614, and DARPA MoBIES grant F33615-00-C-1707.

References

[1] R. Alur, T. Henzinger, G. Lafferriere, and G. J. Pappas. Discrete Abstractions of Hybrid Systems. *Proceedings of the IEEE*, 88(2):971-984, July 2000.

[2] 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.

[3] 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.

[4] J. P. Hespanha, A. Stephen Morse. Switching Between Stabilizing Controllers, Submitted to *Auotmatica*, 2000.

[5] A. Isidori, C. I. Byrnes. Output Regulation of Nonlinear systems. In *IEEE Transactions on Automatic Control*, vol.35(2), pp. 131-140, February 1990.

[6] R. Kelly, R. Salgado PD Control with Computed Feedforward of Robot Manipultors: A Design Procedure. In *IEEE Transactions on Robotics and Automation*, vol.10(4), pp. 566-571, August 1994.

[7] 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.

[8] T. J. Koo, G. J. Pappas, and S. Sastry. Mode Switching Synthesis for Reachability Specifications, Hybrid Systems : Computation and Control, Lecture Notes in Computer Science, volume 2034, 2001.

[9] 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.

[10] A.B. Kurzhanski, P.Varaiya, Ellipsoidal Techniques for Reachability Analysis, Hybrid Systems : Computation and Control, Lecture Notes in Computer Science, 2000.

[11] G. Lafferriere, G.J. Pappas, and S. Yovine. Symbolic Reachability Computations for Families of Linear Vector Fields, Journal of Symbolic Computation, 32(3):231-253, September 2001.

[12] J. Lygeros, C. Tomlin, S. Sastry. Controllers for Reachability Specifications for Hybrid Systems, Automatica, Volume 35, Number 3, March 1999.

[13] R. Milner. Communication and Concurrency, Prentice Hall, 1989.

[14] G.J. Pappas, G. Lafferriere, and S. Sastry, Hierarchically Consistent Control Systems, IEEE Transactions on Automatic Control, 45(6):1144-1160, June 2000.

[15] C. Tomlin, G. Pappas, J. Lygeros, D. Godbole, and S. Sastry. Hybrid Control Models of Next Generation Air Traffic Management, *Hybrid Systems IV*, volume 1273 of Lecture Notes in Computer Science, pages 378-404, Springer Verlag, Berlin, Germany, 1997.

[16] S. Sastry. *Nonlinear Systems: Analysis, Stability, and Control.* Springer-Verlag, New York, 1999.