Chapter 4

Robust Regulation of Polytopic Uncertain Linear Hybrid Systems with Networked Control System Applications*

Hai Lin and Panos J. Antsaklis

Abstract: In this chapter, a class of discrete time uncertain linear hybrid systems, affected by both parameter variations and exterior disturbances, is considered. The main question is whether there exists a controller such that the closed loop system exhibits desired behavior under dynamic uncertainty and exterior disturbances. The notion of *attainability* is introduced to refer to the specified behavior that can be forced to the plant by a control mechanism. We give a method for attainability checking that employs the predecessor operator and backward reachability analysis, and a procedure for controller design that uses finite automata and linear programming techniques. Finally, Networked Control Systems (NCS) are proposed as a promising application area of the results and tools developed here, and the ultimate boundedness control problem for the NCS with uncertain delay, package dropout and quantization effects is formulated as a regulation problem for an uncertain hybrid system.

*This work was supported in part by the National Science Foundation (ECS99-12458 & CCR01-13131), and by the DARPA/ITO-NEST Program (AF-F30602-01-2-0526).

4.1 Introduction

Hybrid Systems are heterogeneous dynamical systems of which the behavior is determined by interacting continuous-variable and discrete-event dynamics [2, 26]. The last decade has seen considerable research activities in modeling, analysis and synthesis of hybrid systems involving researchers from a number of traditionally distinct fields. On one hand, computer scientists extend their computational models and verification methods from discrete systems to hybrid systems by embedding the continuous dynamics into their discrete models. Typically these approaches are able to deal with complex discrete dynamics described by finite automata and emphasize analysis results (verification) and simulation methodologies. From this perspective, the safety or invariance properties have gained the most attention [3, 23]. Other properties investigated include the qualitative temporal notions of liveness, non-blocking, fairness along infinite trajectories and qualitative ordering of events along trajectories [24]. One of the main formal methods is symbolic model checking, which is based on the computation of reachable sets for hybrid systems [1]. As a result, a good deal of research effort has been focused on developing sophisticated techniques drawn from optimal control, game theory, and computational geometry to calculate or approximate the reachable sets for various classes of hybrid systems [9, 3]. However, the reachability (hence verification) problem is undecidable for most interesting classes of systems [1]. Working in parallel, researchers from the areas of dynamical systems and control theory have viewed hybrid systems as collections of differential/ difference equations with discontinuous or multivalued right-hand sides [7, 19, 4]. In these approaches, the models and methodologies for continues-valued variables which are described by ordinary differential/ difference equations were extended to include discrete variables that exhibit jumps or extend results to switching systems. Typically these approaches are able to deal with complex continuous dynamics and mainly concern stability [19, 10], robustness [15, 10] and synthesis issues [7, 19, 10]. However, there has been little work on integrating these concerns within a framework for formal methods. Perhaps because formal methods traditionally lie in the realm of discrete mathematics, while these concerns from control theory lie separately in the realm of continuous mathematics. In this chapter, we will attempt to integrate these concerns within a framework for formal methods.

The model uncertainty and robust control of hybrid systems is an under-explored and highly promising field [15, 24]. Reachability analysis for uncertain hybrid systems has appeared in [17, 21], and there is also some work on analyzing the induced gain of switched systems [14, 29, 30]. In [29], the $\mathcal{L}_2$ gain of continuous-time linear switched systems is studied by an average dwell time approach incorporated with a piecewise linear Lyapunov function, and the results are extended to discrete-time case in [30]. In [14], the root-mean-square (RMS) gain of a continuous time switched linear system is computed in terms of the solution to a differential Riccati equation when the interval between consecutive switchings is large. In [24], the authors give an abstract algorithm, based on modal logic formalism, to design the switching mechanism among a finite number of continuous systems, and the closed-loop system forms a hybrid automata and satisfies the specifications. In [13], the problem of controlling a poorly modeled continuous-time linear system is addressed. The proposed approach is to employ logic-based switching among a family of candidate controllers. Note that most of the existing methods for synthesizing hybrid control systems either manually decouple the synthesis of the continuous control law and the design of the discrete event control signal, or only design the switching mechanism. Considering the fact that the continuous dynamics and discrete dynamics are interacting (coupling) tightly in hybrid systems, we believe that the synthesis problem of hybrid systems has not been solved in a satisfactory way. In this chapter we attempt to present an integrated framework that directly addresses synthesis issues for both the continuous and discrete parts of the hybrid control systems.

In this chapter, we concentrate on a class of uncertain hybrid systems with polytopic uncertain continuous dynamics, called discrete time polytopic uncertain linear hybrid systems. The motivation for introducing uncertainty into the hybrid dynamical systems' model can be described as follows. First, *uncertainty* of the plant and environment is one of the main challenges to control theory and engineering. Therefore, it is very important for the controller design stage to ensure that the desired performances are preserved even under the effect of uncertainties. The system parameters are often subject to unknown, possibly time-varying, perturbations. Moreover, the real processes are often affected by disturbances and it is necessary to consider them in control design. Another challenge to control theory and engineering is the *nonlinearity* of the real world dynamics, since no general methodologies that deal effectively with nonlinear systems exist as yet. In order to avoid dealing directly with a set of nonlinear equations one may choose to work with sets of simpler equations (e.g., linear) and switch among these simpler models. This is a rather common approach in modeling physical phenomena. In control, switching among simpler dynamical systems has been used successfully in practice for many decades. Recent efforts in hybrid systems research along these lines typically concentrate on the analysis of the dynamic behaviors and aim to design controllers with guaranteed stability and performance, see for example [28, 16, 5, 18] and the references therein.

Uncertain systems with strong nonlinearities are often of interest. If we use ordinary piecewise linear systems to approximate and study such nonlinear systems, we have to shrink the operating region of the linearization. And this results in a large number of linearized models which makes the subsequent analysis and synthesis computationally expensive or even intractable. So we propose to introduce a bundle of linearization, whose convex hull cover the original (maybe uncertain) nonlinear dynamics, instead of approximating with just single linearization. In this way, we may keep the operating region from shrinking, and so we may study uncertain nonlinear systems in a systematic way and with less computational burden (see Figure 4.1.1).

Our control objective is for the closed loop system to exhibit certain desired behavior despite the uncertainty and disturbance. Specifically, given finite number of regions $\{\Omega_0,\ \Omega_1, \cdots,\ \Omega_M\}$ in the state space, our goal is for the closed loop system

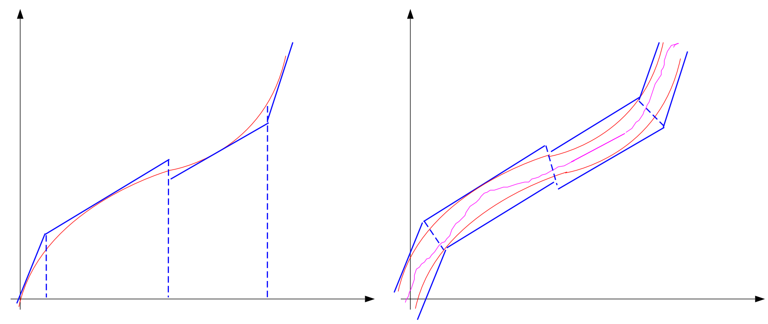

Figure 4.1.1: Piecewise linear approximation and uncertain piecewise linear coverage.

trajectories, starting from the given initial region Ω_0, to go through the sequence of finite number of regions $\Omega_1, \ \Omega_2, \cdots, \ \Omega_M$ in the desired order and finally to reach the final region Ω_M and then remain in Ω_M. This kind of specification is analogous to the ordinary tracking and regulation problem in pure continuous variable dynamical control systems. In addition, it also reflects the qualitative ordering of event requirements along trajectories. One of the main questions is to determine whether there exist admissible control law such that the region-sequence can be followed. If there exists such admissible control law, the region-sequence specification $\{\Omega_0, \ \Omega_1, \cdots, \ \Omega_M\}$ is called *attainable*. The attainability checking is based on backward reachability analysis and symbolic model checking method. The next question is how to design an admissible control law in order to satisfy the closed loop specification. An optimization based method is given in this chapter to design such admissible control law†.

The organization of the chapter is as follows. Section 4.2 defines polytopic uncertain linear hybrid systems, and formulates the tracking and regulation problems. Then, a robust one-step predecessor operator for the uncertain linear hybrid systems is studied in Section 4.3, which serves as the basic tool for analysis that follows. In Section 4.4, the necessary and sufficient conditions for checking the safety, reachability and attainability are given. The robust controller synthesis problem for the polytopic uncertain linear hybrid systems is studied next. The control problem considered is event sequence tracking (servo) control or regulation, and it is formulated and solved in Section 4.5. Networked Control Systems (NCS) are proposed as a possible application field of the theoretic results and tools developed here, and the ultimate boundedness control for NCS is formulated as a regulation problem for uncertain hybrid systems studied in this chapter. Finally, concluding remarks are made.

†This chapter is an extension of our group's previous work [18] to uncertain systems and to more general cases. Earlier work appeared in [21, 22].

4.2 Problem Formulation

We are interested in the following discrete-time uncertain hybrid dynamical systems:

Definition 4.2.1. *The discrete-time* Polytopic Uncertain Linear Hybrid Systems *are defined by*

$$x(t+1) = \tilde{A}_{q(t)}x(t) + \tilde{B}_{q(t)}u(t) + E_{q(t)}d(t) \tag{4.2.1}$$
$$q(t+1) = \delta(q(t), \pi(x(t)), \sigma_c(t), \sigma_u(t)) \tag{4.2.2}$$

where $q \in Q = \{q_1, q_2, \cdots, q_s\}$ and Q is the collection of discrete states (modes); $x \in X \subset \mathbb{R}^n$ and X stands for the continuous state space. For mode q, the continuous control $u \in \mathcal{U}_q \subset \mathbb{R}^m$, and the continuous disturbance $d \in \mathcal{D}_q \subset \mathbb{R}^p$, where $\mathcal{U}_q$, $\mathcal{D}_q$ are bounded convex polyhedral sets. Denote

$$\mathcal{U} = \bigcup_{q\in Q} \mathcal{U}_q, \ \ \mathcal{D} = \bigcup_{q\in Q} \mathcal{D}_q,$$

and

- *$\tilde{A}_q \in \mathbb{R}^{n\times n}$, $\tilde{B}_q \in \mathbb{R}^{n\times m}$, and $E_q \in \mathbb{R}^{n\times p}$ are the system matrices for the discrete state q. The entries in $\tilde{A}_q$ and $\tilde{B}_q$ are unknown, and maybe time-variant, but $[\tilde{A}_q, \tilde{B}_q]$ are contained in a convex hull in $\mathbb{R}^{n\times n} \times \mathbb{R}^{n\times m}$, that is*

$$[\tilde{A}_q, \tilde{B}_q] = \sum_{i=1}^{N_q} \lambda_i [A_q^i, B_q^i], \quad \lambda_i \geq 0, \sum_{i=1}^{N_q} \lambda_i = 1$$

- *$\pi\colon X \to X/E_\pi$ partitions the continuous state space $X \subset \mathbb{R}^n$ into polyhedral equivalence classes.*
- *$q(t+1) \in act(\pi(x(t)))$, where $act\colon X/E_\pi \to 2^Q$ defines the active mode set,*
- *$\delta\colon Q \times X/E_\pi \times \Sigma_c \times \Sigma_u \to Q$ is the discrete state transition function. Here $\sigma_c \in \Sigma_c$ denotes a controllable event and Σ_u the collection of uncontrollable events.*
- *The* guard *$G(q, q')$ of the transition (q, q') is defined as the set of all continuous states x such that $q, q' \in act(\pi(x(t)))$ and there exist controllable event $\sigma_c \in \Sigma_c$ such that $q' = \delta(q, \pi(x), \sigma_c, \sigma_u)$ for every uncontrollable event $\sigma_u \in \Sigma_u$. The guard of the transition describes the region of the continuous state space where the transition can be forced to take place independently of the disturbances generated by the environment.*

Remark: Note that in the above definition, we do not consider "state jumps" (reset) for continuous state x explicitly. However, the reset function can be easily included in our model by adding some auxiliary modes.

In the following we assume the existence of the solution for such uncertain hybrid systems under given initial conditions. And we assume that exact state measurement (q, x) is available. An *admissible control input* (or *law*) is one which satisfies the input constraints $(\Sigma_c, \mathcal{U}_q)$. The elements of an allowable disturbance sequence are contained in $(\Sigma_u, \mathcal{D}_q)$.

We consider specifications that are described with respect to regions of the hybrid state space. Consider a finite number of regions $\{\Omega_0,\ \Omega_1, \cdots,\ \Omega_M\} \subset Q \times X$, where $\Omega_i = (\mathbf{q}_i, P_i)$ are regions in the hybrid state space. Note that the continuous part P_i does not necessarily coincide with the partitions of π in Definition 4.2.1, and this gives us more flexibility. However, it is required that the following consistency condition holds.

$$P_i \subset \bigcap_{q_i \in \mathbf{q}_i} Inv(q_i) \tag{4.2.3}$$

where

$$Inv(q_i) = \{x \in X : q_i \in act(\pi(x))\}.$$

$Inv(q_i)$ is similar to the concept of invariant set of mode q_i in hybrid automata.

Our control objective is for the closed loop system trajectories to follow a given sequence of regions $\{\Omega_0,\ \Omega_1, \cdots,\ \Omega_M\} \subset Q \times X$, despite the uncertainty and disturbance. One of the main questions is to check whether there exists an admissible control law, $\sigma_c[q(t), x(t)] \in \Sigma_c$ and $u[q(t), x(t)] \in \mathcal{U}_{q(t)}$, such that the hybrid state trajectory $(q(t), x(t))$ goes through the regions, $\Omega_0, \Omega_1, \Omega_2, \cdots$, in the specified order and the closed loop system satisfies some desired requirements. This involves sequencing of events and eventual execution of actions. If there exist admissible control laws such that the region-sequence starting from Ω_0 can be followed, we call the sequence of regions specification $\{\Omega_0,\ \Omega_1, \cdots,\ \Omega_M\} \subset Q \times X$ attainable. In order to check the attainability for a sequence of regions specification, two different kinds of properties should be checked, that is the direct reachability from region Ω_i to Ω_{i+1} for $0 \le i < M$ and the safety (or controlled invariance) for region Ω_M. The analysis problems for safety and direct reachability are formulated as follows.

- Safety: Given a region $\Omega \subset Q \times X$, determine whether there exist admissible control laws such that the evolution of the system starting from Ω will remain inside the region for all time, despite the presence of dynamic uncertainties and disturbances.

- Reachability: Given two regions $\Omega_1,\ \Omega_2 \subset Q \times X$, determine whether there exist admissible control laws such that all the states in Ω_1 can be driven into Ω_2 in finite number steps without entering a third region.

The safety, reachability and attainability checking are all based on backward reachability analysis and symbolic model checking method. In the next section, we will briefly discuss the backward reachability analysis, which serves as one of the basic tools for the analysis that follows. After answering how to check the attainability of a specification, we will design the admissible control law, $\sigma_c[q(t), x(t)]$ and $u[q(t), x(t)]$, such that the hybrid state trajectory $(q(t), x(t))$ goes through the regions, $\Omega_0, \Omega_1, \cdots$, in the specified order and such that the closed loop system satisfies some desired requirements.

4.3 Robust One-Step Predecessor Set

The basic building block to be used for backward reachability analysis is the *robust predecessor operator*, which is defined below.

Definition 4.3.1. The *robust one-step predecessor set*, $pre(\Omega)$, is the set of states in $Q \times X$ for which an admissible control input exists and guarantees that the system will be driven to Ω in one step, for all allowable disturbances and dynamic uncertainties, i.e.,

$$\begin{aligned} pre(\Omega) = \{&(q(t), x(t)) \in Q \times X | \forall \sigma_u \in \Sigma_u, d(t) \in \mathcal{D}_{q(t)}, \\ &\exists \sigma_c \in \Sigma_c,\ u(t) \in \mathcal{U}_{q(t)},\ s.t.\ \tilde{A}_{q(t)} x(t) + \tilde{B}_{q(t)} u(t) + E_{q(t)} d(t)) \in \Omega\} \end{aligned}$$

where

$$[\tilde{A}_{q(t)}, \tilde{B}_{q(t)}] \in Conv_{i=1}^{N_q} [A^i_{q(t)}, B^i_{q(t)}].$$

The predecessor operator has the following properties.

Proposition 4.3.1. *For all Ω_1 and Ω_2, $\Omega_1 \subseteq \Omega_2 \Rightarrow pre(\Omega_1) \subseteq pre(\Omega_2)$. If Ω is given by the union, $\Omega = \bigcup_i \Omega_i$, then $pre(\Omega) = \bigcup_i pre(\Omega_i)$.*

Next, let's assume that a region of the state space is defined as $\Omega = (\mathbf{q}, P) \subset Q \times X$, where P is a piecewise linear set. Without loss of generality, we assume that P is convex and can be represented by $P = \{x \in X | Gx \leq w\}$, where $G \in \mathbb{R}^{v \times n}$, $w \in \mathbb{R}^v$. Here $a \leq b$ means that all entries in the vector $(a - b)$ are all nonpositive. If P is nonconvex, then it is known that nonconvex piecewise linear set P can be written as finite union of convex piecewise linear set P_i, that is $P = \cup_{i=1}^m P_i$ [27]. And $\Omega = \bigcup_i \Omega_i = \bigcup_i (\mathbf{q}, P_i)$. Because of the above proposition, we have $pre(\Omega) = \bigcup_i pre(\Omega_i)$. Similarly, without loss of generality we assume that the discrete part of Ω contains only one mode, that is $|\mathbf{q}| = 1$ or $\mathbf{q} = \{q\}$.

We are interested in computing the set of all the states that can be driven to $\Omega = (q, \{Gx \leq w\})$ by both continuous and discrete transitions despite the presence of dynamic uncertainties and disturbances. To calculate $pre(\Omega)$, we first calculate the predecessor set for Ω either purely by discrete transition, $pre_d(\Omega)$, or purely by continuous transition at mode q, $pre_c^q(\Omega)$. Then an algorithm is given for $pre(\Omega)$ by considering the coupling between $pre_d(\Omega)$ and $pre_c^q(\Omega)$.

4.3.1 Discrete Transitions

The predecessor operator for discrete transitions is denoted by $pre_d\colon 2^{Q\times X} \to 2^{Q\times X}$, and it is used to compute the set of states that can be driven to the region Ω by a discrete instantaneous transition $q' \to q$ which can be forced by the controller for any uncontrollable event. The predecessor operator in this case is defined as follows:

$$pre_d(\Omega) = \{(q', x) \in Q \times P | \exists \sigma_c \in \Sigma_c, \forall \sigma_u \in \Sigma_u,\ q = \delta(q', x, \sigma_c, \sigma_u)\}$$

For every discrete transition that can be forced by a controllable event we have that

$$pre_d(\Omega) = \bigcup_{q' \in act(P)} \{q'\} \times (G(q', q) \cap P)$$

where $G(q', q)$ is the guard set of transition $q' \to q$.

4.3.2 Continuous Transitions

In the case of continuous transitions, we define the continuous predecessor operator under mode q as $pre_c^q\colon 2^{Q\times X} \to 2^{Q\times X}$. It computes the set of states for which there exists a control input so that the continuous state will be driven into the set P for every disturbance and uncertainty, while the system is at the discrete mode q. The action of the operator is described by

$$\begin{aligned} pre_c^q(\Omega) = \{q\} \times \{x \in X | \exists u \in \mathcal{U}_q, \forall d \in \mathcal{D}_q, \forall [\tilde{A}_q, \tilde{B}_q] \in Conv_{i=1}^{N_q}([A_q^i, B_q^i]), \\ s.t.\ \tilde{A}_q x + \tilde{B}_q u + E_q d \in P\} \end{aligned}$$

4.3.3 Computation of the Predecessor Operator

As it was explained above, the predecessor operator for discrete transitions is given by the union of the guards of those transitions that are feasible and can be forced by a control mechanism. Since the guards are regions of the state space that are included in the description of the model, we concentrate on predecessor operator for the continuous transitions.

Let us denote $pre_{c,i}^q(\Omega)$ the continuous predecessor set of the i-th vertex $[A_q^i, B_q^i]$ for $1 \leq i \leq N_q$. That is

$$pre_{c,i}^q(\Omega) = \{x \in X | \exists u \in \mathcal{U}_q, \forall d \in \mathcal{D}_q,\ s.t.\ A_q^i x + B_q^i u + E_q d \in P\}$$

Because of linearity and convexity, we can derive the relationship between $pre_{c,i}^q(\Omega)$ and $pre_c^q(\Omega)$ as the following proposition.

Proposition 4.3.2. $pre_c^q(\Omega) = \{q\} \times \bigcap_{i=1}^{N_q} pre_{c,i}^q(\Omega)$.

Remark: The significance of the proposition is that the calculation for the continuous predecessor for the polytopic uncertain linear hybrid systems can be boiled down to the finite intersection of continuous predecessor sets corresponding to the dynamic matrix polytope vertices, which have deterministic continuous dynamics. And the predecessor set under deterministic continuous dynamics, $pre^q_{c,i}(\Omega)$, can be computed by Fourier-Motzkin elimination [25] and linear programming techniques, which is given in [18].

In the following, we describe an algorithm for calculating the robust predecessor set $pre(\Omega)$ under both discrete and continuous transitions. Consider the uncertain hybrid systems of Definition 4.2.1 and a region $\Omega = (\mathbf{q}, P)$. We denote $\{P^\pi_i\}$ $(i = 1, \cdots, N)$ the partition of the continuous state space X by the map π as given in Definition 4.2.1. The following algorithm computes all the states of the hybrid system that can be driven to Ω in one time-step.

Algorithm 4.3.1. Predecessor Operator

INPUT: $\Omega = (\mathbf{q}, P)$, $S = \emptyset$;
for $i = 1, \cdots, N$,
  $Q_i = P \cap P^\pi_i$
  if $Q_i \neq \emptyset$
    for $q' \in act(P^\pi_i)$
      $S^{q'}_i = G(q', q) \cap Q_i$
      if $S^{q'}_i \neq \emptyset$
        $V = X$
        for $j = 1, \cdots, N_{q'}$
          $V = V \cap pre^{q'}_{c,j}(q' \times S^{q'}_i)$
        end
        if $V \neq \emptyset$
          $S = S \cup (\{q'\} \times V)$
        end if
      end if
    end
  end if
end
OUTPUT: $pre(\Omega) = S$

Remark: Note that $pre^q_{c,i}(\Omega)$ is piecewise linear and piecewise linear sets are closed under finite intersections, so $\bigcap_{i=1}^{N_q} pre^q_{c,i}(\Omega)$ is also piecewise linear set. Piecewise linear sets are relatively easy to be calculated and efficiently represented in computers.

Let us see an numerical example for the predecessor operator.

Example 4.3.1 (PREDECESSOR OPERATOR). Consider an uncertain hybrid system with two modes, q_0 and q_1. The continuous dynamics are described by:

$$x(t+1) = \begin{cases} \tilde{A}_0 x(t) + \tilde{B}_0 u(t) + E_0 d(t), & q = q_0 \\ \tilde{A}_1 x(t) + \tilde{B}_1 u(t) + E_1 d(t), & q = q_1. \end{cases}$$

where

$$\begin{aligned}
A_0^1 &= \begin{pmatrix} 0.9568 & 0.1730 & 0.2523 \\ 0.5226 & 0.9797 & 0.8757 \\ 0.8801 & 0.2714 & 0.7373 \end{pmatrix}, \; A_0^2 = \begin{pmatrix} 1.0934 & 0.3721 & 0.5367 \\ 0.5343 & 1.2785 & 1.3450 \\ 1.7740 & 0.9329 & 0.8021 \end{pmatrix} \\
B_0^1 &= \begin{pmatrix} 0.9883 \\ 0.5828 \\ 0.4235 \end{pmatrix}, \; B_0^2 = \begin{pmatrix} 1.5038 \\ 0.9167 \\ 0.8564 \end{pmatrix}, \; E_0 = \begin{pmatrix} 0.2259 \\ 0.5798 \\ 0.7604 \end{pmatrix} \\
A_1^1 &= \begin{pmatrix} 0.1509 & 0.8600 & 0.4966 \\ 0.6979 & 0.8537 & 0.8998 \\ 0.3784 & 0.5936 & 0.8216 \end{pmatrix}, \; A_1^2 = \begin{pmatrix} 0.2154 & 0.8942 & 0.5500 \\ 0.7797 & 0.8826 & 0.9725 \\ 0.4444 & 0.6277 & 0.8526 \end{pmatrix} \\
B_1^1 &= \begin{pmatrix} 0.8385 \\ 0.5681 \\ 0.3704 \end{pmatrix}, \; B_1^2 = \begin{pmatrix} 0.9088 \\ 0.6227 \\ 0.4149 \end{pmatrix}, \; E_1 = \begin{pmatrix} 0.6946 \\ 0.6213 \\ 0.7948 \end{pmatrix}
\end{aligned}$$

The partition of the state space is obtained by considering the following hyperplane $h_1(x) = x_1 - 5$, $h_2(x) = x_2 - 5$, $h_3(x) = x_3 - 5$, $h_4(x) = x_1$, $h_5(x) = x_2$ and $h_6(x) = x_3$. Assume $u \in \mathcal{U} = [-1, 1]$, $d \in \mathcal{D} = [-0.1, 0.1]$. Here we assume that $X = \mathbb{R}^3$, and $G_{q_0}^{q_1} = G_{q_1}^{q_0} = X$. Consider region $\Omega = (\{q_0, q_1\}, P)$, where P is the tube with edge 5. In order to calculate $pre(\Omega)$, we first calculate the continuous predecessor sets $pre_c^{q_0}(P)$, $pre_c^{q_0}(P \cap G_{q_0}^{q_1})$, $pre_c^{q_1}(P)$ and $pre_c^{q_1}(P \cap G_{q_1}^{q_0})$. It turns out that $pre_c^{q_0}(P \cap G_{q_0}^{q_1}) = pre_c^{q_0}(P)$, which is shown in the left part of Figure 4.3.1, and $pre_c^{q_1}(P \cap G_{q_1}^{q_0}) = pre_c^{q_1}(P)$ as shown in the right part of Figure 4.3.1. The predecessor operator is given by

$$pre(\Omega) = (q_0, pre_c^{q_0}(P)) \cup (q_1, pre_c^{q_1}(P)).$$

4.4 Safety, Reachability and Attainability

In this section, we first present necessary and sufficient conditions for checking the safety for a given region $\Omega \subset Q \times X$ and the direct reachability between two given

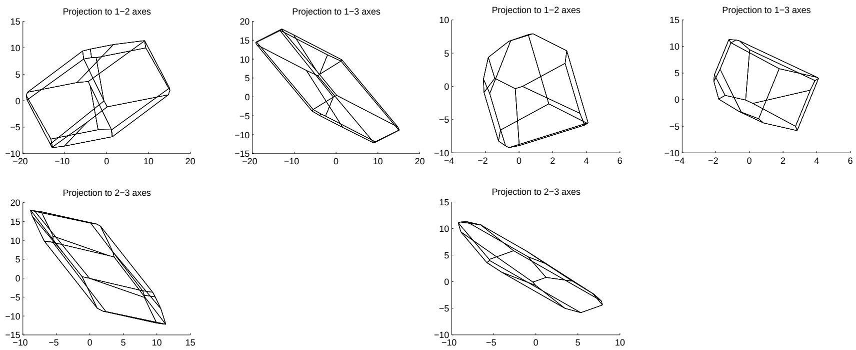

Figure 4.3.1: Illustration for the predecessor sets.

regions Ω_1 and Ω_2. Then a necessary and sufficient condition for checking the attainability of a given specification is presented.

4.4.1 Safety

The following is an important, well-known geometric condition for a set to be safe (controlled invariant).

Theorem 4.4.1. *The set Ω is safe if and only if $\Omega \subseteq pre(\Omega)$.*

The proof follows immediately from the definition of the predecessor set $pre(\Omega)$. Testing for safety need to: compute $pre(\Omega)$, which can be efficiently done by the predecessor operator algorithm described in the former section; test whether $\Omega \subseteq pre(\Omega)$, this can be done by a feasibility of a linear programming problem. So this condition can be efficiently tested by solving a finite number of linear programming problems that depends on the number of regions and discrete states of the system.

4.4.2 Reachability

Consider the reachability problem for uncertain linear hybrid systems in Definition 4.2.1. It should be emphasized that we are interested only in the case when reachability between two regions Ω_1 and Ω_2 is defined so that the state is driven to Ω_2 directly from the region Ω_1 in finite steps without entering a third region. This is a problem of practical importance in hybrid systems since it is often desirable to drive the state to a target region of the state space while satisfying constraints on the state and input during the operation of the system.

The problem of deciding whether a region Ω_2 is directly reachable from Ω_1 can be solved by recursively computing all the states that can be driven to Ω_2 from Ω_1 using the predecessor operator. Given Ω_1 and Ω_2, define N-step directly reachable set from Ω_1 to Ω_2,

$$PRE^N_{\Omega_1}(\Omega_2) = \underbrace{pre(\cdots pre(pre(\Omega_2) \cap \Omega_1) \cap \Omega_1 \cdots)}_{N times}$$

assume $PRE^0_{\Omega_1}(\Omega_2) = \Omega_1 \cap \Omega_2$. With the introduction of the directly reachable set from Ω_1 to Ω_2 by taking union of all the finite step directly reachable set from Ω_1 to Ω_2, that is

$$CR_{\Omega_1}(\Omega_2) = \bigcup_{N=0}^{\infty} PRE^N_{\Omega_1}(\Omega_2),$$

the geometric condition to check the direct reachability can be given as follows.

Theorem 4.4.2. *Consider an uncertain hybrid systems described by Definition 4.2.1 and the regions Ω_1 and Ω_2. The region Ω_2 is directly reachable from Ω_1 if and only if $\Omega_1 \subseteq CR_{\Omega_1}(\Omega_2)$.*

In general, the reachability problem for hybrid systems is undecidable. So the above procedure is *semi-decidable* [1], because the termination of the procedure is not guaranteed. In order to formulate a constructive procedure for reachability, two approaches maybe employed. First, we consider an upper bound on the time horizon and we examine the reachability only for the predetermined finite horizon. Second, we formulate a termination condition for the reachability algorithm based on a grid-based approximation of the piecewise linear regions of the state space [18].

4.4.3 Attainability

Given finite number of regions $\{\Omega_0,\ \Omega_1, \cdots,\ \Omega_M\} \subset Q \times X$, the attainability for this sequence of regions specification is equivalent to the following two different kinds of properties, that is the direct reachability from region Ω_i to Ω_{i+1} for $0 \leq i < M$ and the safety (or controlled invariance) for region Ω_M. Therefore the attainability checking can be expressed as follows.

Theorem 4.4.3. *The specification $\{\Omega_0,\ \Omega_1, \cdots,\ \Omega_M\} \subset Q \times X$ is attainable* if and only if *the following conditions hold: First, Ω_M is safe; and secondly the region Ω_{i+1} is directly reachable from Ω_i, for $i = 0, 1, \cdots, M-1$.*

4.5 Hybrid Regulation

The hybrid regulation problem considered in this section is to design the admissible control law, $\sigma_c[q(t), x(t)]$ and $u[q(t), x(t)]$, such that the hybrid state trajectory $(q(t), x(t))$ goes through the regions, $\Omega_0, \Omega_1, \Omega_2, \cdots$, in the specified order and so that the closed loop system satisfies some desired requirements. The requirements include sequencing of events and eventual execution of actions. In Section 4.4, we have specified the conditions for the existence of such control laws so that the closed loop system satisfies the specifications, like safety, reachability and attainability. A *regulator* is designed as a dynamical system to implement the desired control policy. Here we design such control laws based on optimization techniques. From the discussion in the previous section, the attainability regulation problem can be divided into two basic problems, that is safety regulation and direct reachability regulation. The two basic regulation problems are formulated as follows.

• Safety Regulation: Given a safe set $\Omega \subset Q \times X$, determine the admissible control laws such that the evolution of the system starting from Ω will remain inside the set for all time, despite the presence of dynamic uncertainties and disturbances.

• Reachability Regulation: Given two regions $\Omega_1,\ \Omega_2 \subset Q \times X$, where Ω_2 is directly reachable from Ω_1, determine the admissible control laws such that all the states in Ω_1 can be driven into Ω_2 in finite number steps without entering a third region.

In the following we present a systematic procedure for the regulator design for these two basic cases based on optimization techniques. Then a procedure for attainability regulation is given.

4.5.1 Safety Regulator

First, we consider terminating safety regulation for the terminating region, $\Omega_M = (\mathbf{q}_M, P_M)$, of the state space. Let's assume that $P_M = \{x \colon G_M x \leq w_M\}$. We define the cost functional $J_M \colon Q \times [0,1]^{N_q} \times \mathcal{U}_q \to \mathbb{R}$

$$J_M(q, \lambda, u) = \|G_M \sum_{i=1}^{N_q} [\lambda_i A_q^i, \lambda_i B_q^i] \begin{pmatrix} x(t) \\ u(t) \end{pmatrix} \|_\infty$$

where $\|\cdot\|_\infty$ stands for the infinite norm. The control signal is selected as the solution to the following minmax optimization problem:

$$\min_{u \in \mathcal{U}_q} \max_{\lambda \in [0,1]^{N_q}} J_M(q, \lambda, u)$$
$$s.t. \begin{cases} G_M[A_q^1 x(t) + B_q^1 u(t)] \leq w_M - \delta \\ G_M[A_q^2 x(t) + B_q^2 u(t)] \leq w_M - \delta \\ \cdots\cdots \\ G_M[A_q^{N_q} x(t) + B_q^{N_q} u(t)] \leq w_M - \delta \\ u \in \mathcal{U}_q \end{cases}$$

where δ is a vector whose components are given by $\delta_j = \max_{d \in \mathcal{D}_q} g_j^T E_q d$, and g_j^T is the j-th row of matrix G_M. The optimal action of the controller is one that tries to minimize the maximum cost, and try to counteract the worst disturbance and the worst model uncertainty. This kind of solution is referred to as *Stackelberg solution*. The above minmax optimization problem can be equivalently transformed to the following semi-infinite programming problem [6],

$$\min_{u\in\mathcal{U}_q} z$$
$$s.t. \begin{cases} J_M(q,\lambda,u) \le z \\ G_M[A_q^1 x(t) + B_q^1 u(t)] \le w_M - \delta \\ G_M[A_q^2 x(t) + B_q^2 u(t)] \le w_M - \delta \\ \cdots\cdots \\ G_M[A_q^{N_q} x(t) + B_q^{N_q} u(t)] \le w_M - \delta \\ u \in \mathcal{U}_q. \end{cases}$$

In addition, because of the special form of J_M, the above optimization problems can be reduced to the following form:

$$\min_{u\in\mathcal{U}_q} z$$
$$s.t. \begin{cases} G_M[A_q^1 x(t) + B_q^1 u(t)] \le z \\ G_M[A_q^2 x(t) + B_q^2 u(t)] \le z \\ \cdots\cdots \\ G_M[A_q^{N_q} x(t) + B_q^{N_q} u(t)] \le z \\ G_M[A_q^1 x(t) + B_q^1 u(t)] \le w_M - \delta \\ G_M[A_q^2 x(t) + B_q^2 u(t)] \le w_M - \delta \\ \cdots\cdots \\ G_M[A_q^{N_q} x(t) + B_q^{N_q} u(t)] \le w_M - \delta \\ u \in \mathcal{U}_q. \end{cases}$$

The above problem can be solved very efficiently by solving a linear programming problem for each possible discrete mode. The following algorithm describe the procedure for the synthesis of safety regulator for an given initial condition (q_0, x_0) containing in a specified region $\Omega_M = (\mathbf{q}_M, P_M)$.

Algorithm 4.5.1. Safety Regulator

INPUT: $\Omega_M = (\mathbf{q}_M, P_M)$, (q_0, x_0);
if $\min_u \max_\lambda J_M(q_0, x_0, \lambda, u)$ feasible
  $u^* = arg\min_u J_M(q_0, x_0, \lambda, u)$
  $q^* = q_0$
else
  for $i = 1, \cdots, |\mathbf{q}_M|$,
    $q_i = \mathbf{q}_M(i)$
    if $x_0 \in G_{q_0}^{q_i}$
      $J_M^{q_i} = \min_u \max_\lambda J_M(q_i, x_0, \lambda, u)$
    end
  end
  $q^* = arg\min_{q_i\in\mathbf{q}_M} J_M^{q_i}$
  $u^* = arg\min_u J_M^{q_i}$
end
OUTPUT: u^*, q^*

In the procedure, we first try to remain the mode and avoid switching, simply because switching maybe costly. However, sticking to mode q_0 may be not a good choice, and there may not exist feasible control signal. So the procedure try to take possible mode switchings into consideration and choose the mode that can make the next continuous state farthest from the boundary. It is claimed that there must exist at least one mode $q \in \mathbf{q}_M$ such that the above optimization problem is feasible. Otherwise, it leads to a contradiction to the safety of the region $\Omega_M = (\mathbf{q}_M, P_M)$. Here q^* stands for the mode that corresponds to the minimum cost value J_M, then the candidate control input is selected as $(\sigma_c(t), u^*(t))$ where $q^* = \delta(q(t), \pi(x(t)), \sigma_c(t), \varepsilon)$ and u^* is the solution of the above optimization procedure. In a formal way we can express it as the following proposition.

Proposition 4.5.1. *If the region $\Omega_M = (\mathbf{q}_M, P_M)$ is safe, then the procedure described in algorithm (4.5.1) can solve the safety problem.*

4.5.2 Reachability Regulator

Next, we consider the reachability specification between the regions $\Omega_k = (\mathbf{q}_k, P_k)$ and $\Omega_{k+1} = (\mathbf{q}_{k+1}, P_{k+1})$. The control objective is to drive every state in Ω_k to Ω_{k+1}. Let the convex polyhedral set $P_k = \{x \colon Gx \le w\}$. For a pair of modes $q_k \in \mathbf{q}_k$ and $q_k' \in \mathbf{q}_{k+1}$, assume the intersection of the guard set for (q_k, q_k'), $G_{q_k}^{q_k'}$, with the common region of P_k and P_{k+1}, is not empty. Let's denote this polytope as $P_C^{(q_k, q_k')} = P_k \cap P_{k+1} \cap G_{q_k}^{q_k'} = \{x \colon G_C x \le w_C\}$. We define the cost functional, $J_C \colon Q \times Q \times [0,1]^{N_{q_k}} \times \mathcal{U}_{q_k} \to \mathbb{R}$

$$J_C(q_k, q_k', \lambda, u) = \|G_C \sum_{i=1}^{N_{q_k}} [\lambda_i A_{q_k}^i, \lambda_i B_{q_k}^i] \begin{pmatrix} x(t) \\ u(t) \end{pmatrix} \|_\infty .$$

The control signal is selected as the solution to the following minmax optimization problem:

$$\begin{aligned} &\min_{u \in \mathcal{U}_{q_k}} \max_{\lambda \in [0,1]^{N_{q_k}}} J_C(q_k, q_k', \lambda, u) \\ &\qquad s.t. \begin{cases} G[A_{q_k}^1 x(t) + B_{q_k}^1 u(t)] \le w - \delta \\ G[A_{q_k}^2 x(t) + B_{q_k}^2 u(t)] \le w - \delta \\ \cdots\cdots \\ G[A_{q_k}^{N_{q_k}} x(t) + B_{q_k}^{N_{q_k}} u(t)] \le w - \delta \\ u \in \mathcal{U}_{q_k}. \end{cases} \end{aligned}$$

Similarly, the above minmax optimization problem can be equivalently transformed to the following linear programming problem:

$$\min_{u \in \mathcal{U}_{q_k}} z$$

$$s.t. \begin{cases} G_C[A^1_{q_k} x(t) + B^1_{q_k} u(t)] \leq z \\ G_C[A^2_{q_k} x(t) + B^2_{q_k} u(t)] \leq z \\ \cdots\cdots \\ G_C[A^{N_{q_k}}_{q_k} x(t) + B^{N_{q_k}}_{q_k} u(t)] \leq z \\ G[A^1_{q_k} x(t) + B^1_{q_k} u(t)] \leq w - \delta \\ G[A^2_{q_k} x(t) + B^2_{q_k} u(t)] \leq w - \delta \\ \cdots\cdots \\ G[A^{N_{q_k}}_{q_k} x(t) + B^{N_{q_k}}_{q_k} u(t)] \leq w - \delta \\ u \in \mathcal{U}_{q_k}. \end{cases}$$

The following algorithm design the regulator to guarantee the direct reachability.

Algorithm 4.5.2. Reachability Regulator

INPUT: $\Omega_k = (\mathbf{q}_k, P_k)$, $\Omega_{k+1} = (\mathbf{q}_{k+1}, P_{k+1})$, (q_0, x_0), $\text{feasibility} = 0$;
for $j = 1, \cdots, |\mathbf{q}_{k+1}|$,
  $q'_j = \mathbf{q}_{k+1}(j)$
  if $\min_u \max_\lambda J_C(q_0, q'_j, x_0, \lambda, u)$ feasible
    $J_C^{(q_0, q'_i)} = \min_u \max_\lambda J_C(q_0, q'_j, x_0, \lambda, u)$
    $\text{feasibility} = 1$
  end
end
if $\text{feasibility} == 1$
  $ind = arg \min_{q'_j \in \mathbf{q}_{k+1}} J_C^{(q_0, q'_j)}$
  $u^* = arg \min_u J_C^{(q_0, ind)}$
  $q^* = q_0$
else
  for $i = 1, \cdots, |\mathbf{q}_k|$,
    $q_i = \mathbf{q}_k(i)$
    if $x_0 \in G^{q_i}_{q_0}$
      for $j = 1, \cdots, |\mathbf{q}_{k+1}|$,
        $q'_j = \mathbf{q}_{k+1}(j)$
        $J_C^{(q_i, q'_j)} = \min_u \max_\lambda J_C(q_i, q'_j, x_0, \lambda, u)$
      end
    end
  end
  $[q^*, q'] = arg \min_{q_i \in \mathbf{q}_k; q'_j \in \mathbf{q}_{k+1}} J_M^{(q_i, q'_j)}$
  $u^* = arg \min_u J_M^{q_i}$
end
OUTPUT: u^*, q^*

Similarly, we have the following proposition.

Proposition 4.5.2. *If the region $\Omega_k = (\mathbf{q}_k, P_k)$ is directly reachable to $\Omega_{k+1} = (\mathbf{q}_{k+1}, P_{k+1})$, then the procedure described in algorithm (4.5.2) can solve the reachability problem.*

4.5.3 Attainability Regulator

The following algorithm design the regulator to guarantee the directly attainability for the specification described by $\{\Omega_0, \Omega_1, \cdots, \Omega_M\}$.

Algorithm 4.5.3. Attainability Regulator

INPUT: $\{\Omega_1, \cdots, \Omega_M\}$, (q_0, x_0);
for *n =1,$\cdots$,M-1,*
  while $x_0 \in \Omega_n$ and $x_0 \notin \Omega_{n+1}$
  Design Reachability Regulator from Ω_n to $\Omega_\mathrm{n+1}$
  end
end
Design Safety Regulator for Ω_M
OUTPUT: u^*, q^*

From Theorem 4.4.3 and the previous two propositions on safety and direct reachability regulation, It is immediate to conclude the following proposition.

Proposition 4.5.3. *If the region sequence $\{\Omega_0, \Omega_1, \cdots, \Omega_M\}$ is attainable, then the procedure described in algorithm (4.5.3) can solve the attainability problem.*

Let us turn to an example to illustrate the regulation method.

Example 4.5.1 (TEMPERATURE CONTROL SYSTEM**).** *The system consists of a furnace that can be switched on and off. The control objective is to control the temperature at a point of the system by applying the heat input at a different point. So, the discrete mode only contains two states, which are the furnace is "off", q_0, and the furnace is "on", q_1. The continuous dynamics are described by* ‡

$$x(t+1) = \begin{cases} \tilde{A}_0 x(t) + \tilde{B}_0 u(t) + E_0 d(t), & q = q_0 \\ \tilde{A}_1 x(t) + \tilde{B}_1 u(t) + E_1 d(t), & q = q_1. \end{cases}$$

‡ using zero-order hold sampling with $T = 1$s.

where

$$A_0^1 = \begin{pmatrix} 0.825 & 0.135 \\ 0.68 & 1 \end{pmatrix}, \; A_0^2 = \begin{pmatrix} 1 & 0.35 \\ 0.068 & 0.555 \end{pmatrix}$$
$$B_0^1 = \begin{pmatrix} 1.7 \\ 0.06 \end{pmatrix}, \; B_0^2 = \begin{pmatrix} 1.9 \\ 0.08 \end{pmatrix}, \; E_0 = \begin{pmatrix} 0.0387 \\ 0.3772 \end{pmatrix}$$
$$A_1^1 = \begin{pmatrix} -0.664 & 0.199 \\ 0.199 & 0.264 \end{pmatrix}, \; A_1^2 = \begin{pmatrix} -0.7 & 0.32 \\ 0.32 & 0.44 \end{pmatrix}$$
$$B_1^1 = \begin{pmatrix} 0.8 \\ 0.1 \end{pmatrix}, \; B_1^2 = \begin{pmatrix} 0.9 \\ 0.2 \end{pmatrix}, \; E_1 = \begin{pmatrix} 0.1369 \\ 0.5363 \end{pmatrix}.$$

The state space is partitioned by the following hyperplane $h_1(x) = x_1 - 20$, $h_2(x) = x_2 - 5$, $h_3(x) = x_2$ and $h_4(x) = x_1$. Assume $u \in \mathcal{U} = [-1, 1]$, $d \in \mathcal{D} = [-0.1, 0.1]$. Consider region $\Omega_1 = (\{q_0, q_1\}, P_1)$ and $\Omega_2 = (\{q_0, q_1\}, P_2)$, where $P_1 = \{x \in \mathbb{R}^2 | (0 \leq x_1 \leq 20) \wedge (-20 \leq x_2 \leq 0)\}$, and $P_2 = \{x \in \mathbb{R}^2 | (0 \leq x_1 \leq 20) \wedge (0 \leq x_2 \leq 5)\}$. Our control object is that for every initial sate (q_0, x_0) within region Ω_1 there exist control $u \in \mathcal{U}$ and $\sigma_c \in \Sigma_c$ so that from (q_0, x_0) the state can be driven to Ω_2 without entering a third region, then the state will stay inside Ω_2, no matter what the dynamic uncertainty, continuous and discrete disturbances are. Let's check the attainability. *We first calculate $pre(\Omega_2)$, which cover the region Ω_2, so Ω_2 is safe. By recursively using $pre(.)$, we find that Ω_1 can be driven to Ω_2 in three steps, i.e. Ω_2 reachable from Ω_1. So the attainability of the specification is satisfied. Then, we design the regulator and plot the simulation result for nominal plant (here we choose the epicenter of the state matrix , i.e. $\frac{1}{2}(A_q^1 + A_q^2)$). in Figure 4.5.1. Also the control signal output (σ_c, u) of the regulator is plotted in Figure 4.5.1.*

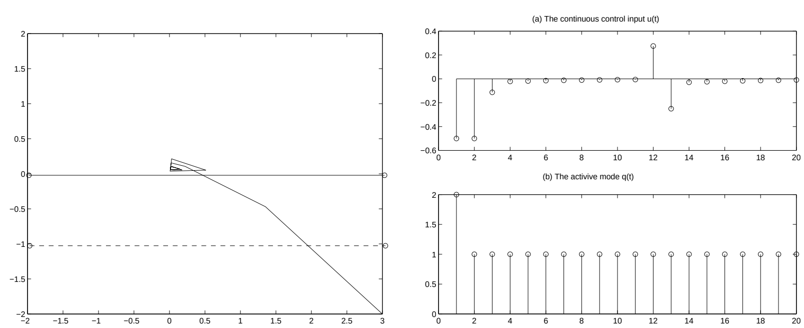

Figure 4.5.1: *Left:* Simulation for close loop nominal plant (assuming d = 0). *Right:* The control signals output (u, σ_c) of the Regulator.

4.6 Networked Control Systems

By Networked Control Systems (NCS), we mean feedback control systems where networks, typically digital band-limited serial communication channels, are used for the connections between spatially distributed system components like sensors and actuators to controllers. And these channels may be shared by other feedback control loops. In traditional feedback control systems these connections are through point-to-point cables. Compared with the point-to-point cables, there are many attractive advantages of introducing serial communication networks, like high system testability and resource utilization, as well as low weight, space, power and wiring requirements. These advantages make the use of networks in control systems connecting sensors/actuators to controllers more and more popular in many applications, including traffic control, satellite clusters, mobile robotics etc. Recently modeling and control of networked control systems with limited communication capability has emerged as a topic of significant interest to control community.

In NCS, the real-time requirement is critical, and time delay usually has negative effects on the NCS stability and performance. There are several situations where time delay may arise. First, transmission delay is caused by the limited bit rate of the communication channels. Secondly, the channel in NCS is usually shared by multiple sources of data, and the channel is usually multiplexed by time-division method. Therefore, there are delays caused by a node waiting to send out a message through a busy channel, which is usually called accessing delay and serves as the main source of delays in NCS. There are also some delays caused by processing and propagation, which are usually negligible for NCS. Another interesting problem in NCS is the package-dropout issue. Because of the uncertainties and noise in the communication channels, there may exist unavoidable errors in the transmitted package or even loss. If this happens, the corrupted package is dropped and the receiver (controller or actuator) uses the package that it received most recently. In addition, package-dropout may occur when one package, say sampled values from the sensor, reaches the destination later than its successors. In such situation, the old package is dropped, and its successive package is used instead. Finally, according to the finite bit rate constrain, only quantized signals can be transmitted through network. So quantization scheme and its effects have to be considered in real NCS. The primary objective of NCS is to efficiently use the finite channel capacity while maintaining good closed loop control system performance, including stability, disturbance attenuation, rising time, overshoot and other design criteria. Therefore, quantization and limited bit rate issues have attracted many researchers' attention, see for example [8, 12, 11]. In this section, we will formulate a NCS with uncertain time delay, package-dropout and quantization effects into the framework of polytopic uncertain hybrid (switched) systems. Then the methods developed here and existing methods for hybrid (switched) systems can be employed to study NCS.

4.6.1 The Delay and Package-Dropout

The NCS model discussed in this section is shown in Figure 4.6.1. For simplicity, but without loss of generality, we may combine all the time delay and package-dropout effects into the sensor to controller path and assume the controller-actuator communicates ideally.

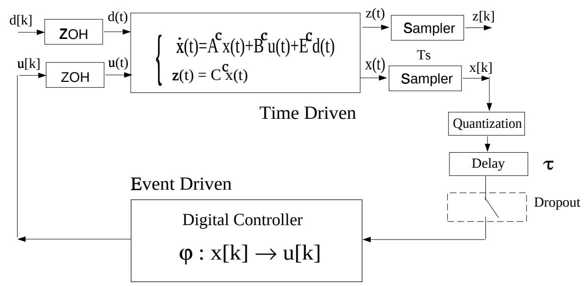

Figure 4.6.1: The networked control systems' model.

We assume that the plant can be modeled as a continuous-time linear time-invariant system described by

$$\begin{cases} \dot{x}(t) = A^c x(t) + B^c u(t) + E^c d(t) \\ z(t) = C^c x(t) \end{cases}$$

where $x(t) \in \mathbb{R}^n$ is the state variable, and $z(t) \in \mathbb{R}^p$ is the controlled output. $u(t) \in \mathbb{R}^m$ is control input. The disturbance input $d(t)$ is contained in $\mathcal{D} \subset \mathbb{R}^r$. $A^c \in \mathbb{R}^{n\times n}$, $B^c \in \mathbb{R}^{n\times m}$ and $E^c \in \mathbb{R}^{n\times r}$ are state matrices, and $C^c \in \mathbb{R}^{p\times n}$ is the output matrix.

It is assumed that the sensors work in a time-driven fashion. After each clock cycle (sampling time T_s), the output node (sensor) attempts to send a package containing the most recent state (output) samples. If the communication bus is idle, then the package will be transmitted to the controller. Else, if the bus is busy, then the output node will wait for, say $\varpi < T_s$, and try again. After several attempts or time elapses, if the transmission still can not be completed, then the package is discarded. The controller and actuator are event driven and work in a simpler way. The controller, as a receiver, has a receiving buffer which contains the most recently received data package from the sensors (the overflow of the buffer may be dealt with as package-dropout). The controller reads the buffer periodically at a higher frequency than the sampling frequency, say every $\frac{T_s}{N}$ for some integer N large enough. Whenever there is new data in the buffer, then the controller will calculate the new control signal and transmit it to the actuators instantly. Upon the new control signal arrival, the actuators update the output of the zero order holder (ZOH) to the new value.

4.6.2 Models for NCS

In this section, we will consider the sampled-data model of the plant. Because we do not assume synchronization between the sampler and the digital controller, the control signal is no longer of constant value within a sampling period. Therefore the control signal within a sampling period has to be divided into subintervals corresponding to the controller's reading buffer period, $T = \frac{T_s}{N}$. Within each subinterval, the control signal is constant in view of the assumptions in the previous section.

Hence the continuous-time plant may be discretized into the following sampled-data system using lifting method.

$$x[k+1] = Ax[k] + \underbrace{[B\ B\ \cdots B]}_{N} \begin{bmatrix} u^1[k] \\ u^2[k] \\ \vdots \\ u^N[k] \end{bmatrix} + Ed[k] \tag{4.6.1}$$

where $A = e^{A^c T_s}$, $B = \int_0^{\frac{T_s}{N}} e^{A^c\eta} B^c d\eta$ and $E = \int_0^{T_s} e^{A^c\eta} E^c d\eta$. Note that for linear time-invariant plant and constant-periodic sampling, the matrices A, B and E are constant.

During each sampling period, several different cases may arise. They are listed below.

a) Delay $\tau = h \times \frac{T_s}{N}$, where $h = 0, 1, 2, \cdots, D_{max}$. For this case $u^1[k] = u^2[k] = \cdots u^h[k] = u[k-1]$, $u^{h+1}[k] = u^{h+2}[k] = \cdots u^N[k] = u[k]$, and Equation (4.6.1) can be written as:

$$\begin{aligned} x[k+1] &= Ax[k] + [B\ B\ \cdots B] \begin{bmatrix} u[k-1] \\ \vdots \\ u[k-1] \\ u[k] \\ \vdots \\ u[k] \end{bmatrix} + Ed[k] \\ &= Ax[k] + h \cdot Bu[k-1] + (N-h) \cdot Bu[k] + Ed[k] \end{aligned}$$

If we let $\hat{x}[k] = \begin{bmatrix} u[k-1] \\ x[k] \end{bmatrix}$, then the above equations can be written as:

$$\hat{x}[k+1] = \begin{bmatrix} 0 & 0 \\ hB & A \end{bmatrix} \hat{x}[k] + \begin{bmatrix} I \\ (N-h)B \end{bmatrix} u[k] + \begin{bmatrix} 0 \\ E \end{bmatrix} d[k]$$

where $h = 0, 1, 2, \cdots, D_{max}$. Note that $h = 0$ implies $\tau = 0$, which corresponds to the "no delay" case. And the controlled output $z[k]$ is given by

$$z[k] = \begin{bmatrix} 0 & C \end{bmatrix} \hat{x}[k]$$

where $C = C^c$.

b) In the case of package-dropout, due to corrupted package or delay $\tau > D_{max} \times \frac{T_s}{N}$, the actuator will implement the previous control signal, i.e. $u^1[k] = u^2[k] = \cdots u^N[k] = u[k-1]$. The state transition equation (4.6.1) for this case can be written as follows.

$$\begin{aligned} x[k+1] &= Ax[k] + [B\ B\ \cdots B] \begin{bmatrix} u[k-1] \\ u[k-1] \\ \vdots \\ u[k-1] \end{bmatrix} + Ed[k] \\ &= Ax[k] + N \cdot Bu[k-1] + Ed[k] \end{aligned}$$

By introducing new state variables,

$$\hat{x}[k+1] = \begin{bmatrix} 0 & 0 \\ NB & A \end{bmatrix} \hat{x}[k] + \begin{bmatrix} I \\ 0 \end{bmatrix} u[k] + \begin{bmatrix} 0 \\ E \end{bmatrix} d[k]$$

The controlled output $z[k]$ is given by

$$z[k] = \begin{bmatrix} 0 & C \end{bmatrix} \hat{x}[k]$$

where $C = C^c$.

4.6.3 NCS Hybrid Model

In this section, we reformulate the NCS problem of the previous section as a hybrid (switched) system with $D_{max} + 2$ different modes. In particular,

$$\left\{ \begin{array}{rcl} \hat{x}[k+1] &=& A_h \hat{x}[k] + B_h u[k] + E_h d[k] \\ z[k] &=& C_h \hat{x}[k] \end{array} \right. \tag{4.6.2}$$

where $A_h = \begin{bmatrix} 0 & I \\ hB & A \end{bmatrix}$, $B_h = \begin{bmatrix} I \\ (N-h)B \end{bmatrix}$, $E_h = \begin{bmatrix} 0 \\ E \end{bmatrix}$ and $C_h = \begin{bmatrix} 0 & C \end{bmatrix}$ for $h = 0, 1, 2, \cdots D_{max}, N$. And the set of modes Q is given by $Q = \{0, 1, 2, \cdots D_{max}, N\}$. Note that $h = 0$ implies $\tau = 0$, which corresponds to the "no delay" case, while $h = N$ corresponds to the "package-dropout" case. The discrete dynamics (switching signal) may be modeled as a finite state machine (FSM), whose discrete states correspond to the $D_{max} + 2$ modes of NCS. The discrete transition rule of the FSM should reflect the delay and package dropout pattern of the NCS, and it may be either specified in average dwell-time sense or in stochastic sense.

There is another important issue which is not considered in the above NCS model, that is the quantization effect. With finite bit-rate constraints, quantization has to be taken into consideration in NCS. It has been known that an exponential data representation scheme is most efficient [8, 12]. Here we focus on the floating point representation. Floating point quantization can be viewed as a nonlinear operation described by a time variant sector gain, i.e. $Q(x) = k(x)$, $k \in [1-\epsilon, 1]$, with ϵ depending on the mantissa length, and for this reason the quantization effect can be dealt with as model parameter uncertainties. Now we can model the NCS with quantization effects as a switched system with parameter uncertainty, which is a specific subclass of the polytopic uncertain hybrid systems defined in Section 4.2. In particular,

$$\hat{x}[k+1] = \tilde{A}_h \hat{x}[k] + \tilde{B}_h u[k] + E_h d[k]$$

where the parameter uncertainty in $\tilde{A}_h$ and $\tilde{B}_h$ reflect the quantization effects in NCS. Next we consider the robust stabilization problem for the NCS based on such hybrid model. Because of the parameter uncertainties in the NCS hybrid model and exteriors disturbances, the convergence of all the closed loop trajectories to the origin (assumed to be the equilibrium) may not be achievable. Instead, we consider the convergence to a small region containing the origin, and it is required that the closed loop trajectories of NCS be driven to a small region containing the origin for all bounded initial conditions. This is usually called in the literature ultimate boundedness control or practical stability problem. In the following, we will show that the ultimate boundedness control problem for the NCS with uncertain delay, package dropout and quantization effects may be formulated as a regulation problem for the uncertain hybrid (switched) systems.

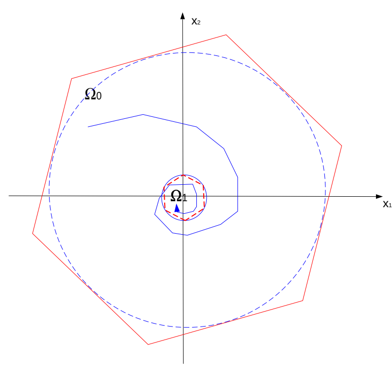

Figure 4.6.2: The networked control systems ultimate boundedness control as a regulation problem.

Consider the semiglobal asymptotic practical stabilization problem by assuming bounded initial states. If we outer-approximate the bounded region containing all the initial conditions with a polytope Ω_0, and inner-approximate the small region containing the origin with another polytope Ω_1 as illustrated in Figure 4.6.2, then the ultimate boundedness problem of NCS can be transformed into a regulation problem. And the ultimate boundedness of NCS can be checked by checking the attainability of the appropriately chosen $\{\Omega_0, \Omega_1\}$. The ultimate boundedness control law maybe designed by solving the optimization based regulator synthesis problem developed in the previous section.

4.7 Conclusions

In this chapter, the regulation problem for the polytopic uncertain linear hybrid systems was formulated and solved. Using the optimization based regulator introduced, the closed loop system exhibits the desired behavior under dynamic uncertainties, continuous disturbances and uncontrollable events. The existence of a controller such that the closed loop system follows desired sequence of regions under uncertainty and disturbance was studied first. Then, based on the novel notion of attainability for the desired behavior of piecewise linear hybrid systems, we presented a systematic procedure for controller design by using finite automata and linear programming techniques. The design procedure may be seen as a one-step moving horizon optimal control. Our future work is intended to generalize the procedure into finite N-step moving horizon optimal control, and the main issue for this generalization is the feasibility of the optimization problems. The predecessor operator, attainability checking and robust regulator design methods have been implemented in a Matlab toolbox called HyStar [20].

The procedures given in this chapter for checking the reachability and attainability, which were based on the backward reachability analysis, are semi-decidable and their termination is not guaranteed. Future work includes specifying the class of polytopic uncertain linear hybrid systems which makes the procedure decidable. One way to obtain such decidable class is to simplify the continuous dynamics, see for example [1]. However, this approach may not be attractive to control applications, where simple continuous dynamics may not be adequate to capture the system's dynamics. Alternatively, one may simplify the discrete dynamics instead of the continuous dynamics, on which our current research effort is focused.

Finally, we proposed Networked Control Systems (NCS) as one of the promising application areas of the methods developed in this chapter. A NCS hybrid model was developed and the ultimate boundedness control problem for NCS was formulated as a hybrid regulation problem. The advantage from formulating the ultimate boundedness control problem of NCS with uncertain delay, package dropout and quantization effects as a regulation problem for uncertain hybrid switched systems comes from the systematic methods developed in this chapter and the existence of rich results in the field of hybrid (switched) systems, jump linear systems etc. Our future work includes researches into other promising application areas for the hybrid regulation methods developed here, such as network congestion control, chemical industrial process control, traffic management, manufacturing systems and robotics.

Bibliography

[1] R. Alur, Tom Henzinger, Gerardo Lafferriere, and George J. Pappas, "Discrete abstractions of hybrid systems," in [2]: 971–984, July 2000.

[2] P. Antsaklis, Ed., *Proceedings of the IEEE*, Special Issue on Hybrid Systems: Theory and Applications, vol. 88 (7), July 2000.

[3] E. Asarin, O. Maler, and A. Pnueli, "Reachability analysis of dynamical systems having piecewise-constant derivatives," *Theoretical Computer Science*, 138: 35–65, 1995.

[4] A. Bemporad, and M. Morari, "Control of systems integrating logic, dynamics, and constraints," *Automatica*, vol. 35 (3): 407–427, 1999.

[5] A. Bemporad, G. Ferrari-Trecate, and M. Morari, "Observability and controllability of piecewise affine and hybrid systems," *IEEE Transactions on Automatic Control*, vol. 45 (10): 1864–1876, 2000.

[6] D. P. Bertsekas, *Nonlinear Programming*, 2nd edition, Athena Scientific, 1999.

[7] M. S. Branicky, V. S. Borkar, and S. K. Mitter, "A unified framework for hybrid control: Model and optimal control theory," *IEEE Transactions on Automatic Control*, vol. 43 (1): 31–45, 1998.

[8] R. W. Brockett, and D. Liberzon, "Quantized feedback stabilization of linear systems," *IEEE Transactions on Automatic Control*, vol. 45 (7): 1279–1289, 2000.

[9] A. Chotinan, and B. H. Krogh, "Verification of infinite-state dynamic systems using approximate quotient transition systems," *IEEE Transactions on Automatic Control*, vol. 46 (9): 1401–1410, 2001.

[10] R. A. Decarlo, M. S. Branicky, S. Pettersson, and B. Lennartson, "Perspectives and results on the stability and stabilizability of hybrid systems," in [2]: 1069–1082, July 2000.

[11] D. F. Delchamps, "Stabilizing a linear system with quantized state feedback," *IEEE Transactions on Automatic Control*, vol. 35 (8): 916–924, 1990.

[12] N. Elia, and S. K. Mitter, "Stabilization of linear systems with limited information," *IEEE Transactions on Automatic Control*, vol. 46 (9): 1384–1400, 2001.

[13] J. Hespanha, D. Liberzon, A. S. Morse, B. D. O. Anderson, T. S. Brinsmead, and F. De Bruyne, "Multiple model adaptive control. Part 2: Switching," *International Journal of Robust and Nonlinear Control*, vol. 11 (5): 479–496, 2001.

[14] J. P. Hespanha, "Computation of Root-Mean-Square Gains of Switched Linear Systems," HSCC 2002, pp. 239–252, 2002.

[15] C. Horn, and P. J. Ramadge, "Robustness issues for hybrid systems," in *Proceedings of the 34th CDC*, pp. 1467–1472, 1995.

[16] M. Johansson, *Piecewise Linear Control Systems*, Ph.D. Thesis, Lund Institute of Technology, Sweden, 1999.

[17] U. T. Jönsson, "On Reachability Analysis of Uncertain Hybrid Systems," in *Proceedings of the 41th CDC*, 2397–2402, 2002.

[18] X. D. Koutsoukos, and P. J. Antsaklis, "Hierarchical Control of Piecewise Linear Hybrid Dynamical Systems Based on Discrete Abstractions," *ISIS Technical Report*, ISIS-2001-001, Feb. 2001.

[19] D. Liberzon, and A. S. Morse, "Basic problems in stability and design of switched systems," *IEEE Control Systems Magazine*, vol. 19 (15): 59–70, 1999.

[20] H. Lin, X. D. Koutsoukos, and P. J. Antsaklis, "HySTAR: A Toolbox for Hierarchical Control of Piecewise Linear Hybrid Dynamical Systems," in *Proceedings of the 2002 ACC*, Anchorage, Alaska, 2002.

[21] H. Lin, X. D. Koutsoukos, and P. J. Antsaklis, "Hierarchical Control of a class Uncertain Piecewise Linear Hybrid Dynamical Systems," *IFAC'2002*, Barcelona, Spain, July 2002.

[22] H. Lin, and P. J. Antsaklis, "Controller Synthesis for a class of Uncertain Piecewise Linear Hybrid Dynamical Systems," in *Proceedings of the 41th CDC*, 2002.

[23] J. Lygeros, C. Tomlin, and S. Sastry, "Controllers for reachability specifications for hybrid systems," *Automatica*, vol. 35 (3): 349–370, 1999.

[24] T. Moor, and J. M. Davoren, "Robust controller synthesis for hybrid systems using modal logic," HSCC 2001, pp. 433–446, 2001.

[25] T. Motzkin, *The theory of linear inequalities*, Rand Corp., Santa Monica, CA, 1952.

[26] A. van der Schaft, and H. Schumacher, *An Introduction to Hybrid Dynamical Systems*, Springer, 2000.

[27] E. Sontag, "Remarks on piecewise-linear algebra," *Pacific Journal of Mathematics*, 92 (1): 183–210, 1982.

[28] E. Sontag, "Interconnected automata and linear systems: A theoretical framework in discrete-time," *Hybrid Systems III*, volume 1066 of Lecture Notes in Computer Science, pp. 436–448. Springer, 1996.

[29] G. Zhai, B. Hu, K. Yasuda, and A. N. Michel, "Disturbance attenuation properties of time controlled switched systems," *Journal of the Franklin Institute*, 338: 765–779, 2001.

[30] G. Zhai, B. Hu, K. Yasuda, and A. N. Michel, "Qualitative analysis of discrete-time switched systems," in *Proceedings of the 2002 ACC*, vol. 3: 1880–1885, 2002.