Design of Hybrid System Regulators[1]

Xenofon D. Koutsoukos and Panos J. Antsaklis
Department of Electrical Engineering
University of Notre Dame
Notre Dame, IN 46556
e-mail:xkoutsou,antsaklis.1@nd.edu

Abstract

In this paper, a new approach for modeling and controlling hybrid systems is presented. Discrete abstractions are used to approximate the continuous dynamics and emphasis is placed on the nondeterministic nature of the abstracting models. The regulator problem for hybrid systems is formulated and an example of a robotic manufacturing system is used to illustrate the approach.

1 Introduction

In this paper, a new approach for modeling and controlling hybrid systems is presented. Discrete abstractions of the continuous dynamics are studied and the emphasis is placed on the nondeterministic nature of the abstracting models. Discrete-time models are used to model the continuous dynamics. The class of systems we are particularly interested in is the class of piecewise-linear systems [8]. Piecewise-linear systems model interesting engineering applications and can be studied with existing powerful mathematical tools [6, 7].

A great amount of research work has already been done in the hybrid systems area during the past decade. The approach presented in this paper has been influenced especially by the work in [1, 10] where a feedback architecture of a continuous plant with a discrete-event controller is utilized for hybrid control design. The problem of obtaining discrete abstractions of continuous systems has also been considered in [2, 3, 4].

The main advantage of the proposed approach is that it provides a general framework for hybrid systems not only for analysis, but more importantly for controller synthesis. The notion of quasideterminism is used to characterize discrete abstractions that can be used for control design and is compared with bisimulations of hybrid automata. In order to develop efficient algorithms, we concentrate on piecewise-linear systems. However, the framework is valid for systems with more general dynamics. It should be noted that this framework allows the implementation of the analysis and synthesis algorithms using existing software such as Matlab, Simulink, and Stateflow.

The paper is organized as follows. A robotic manufacturing system which is used to illustrate the approach throughout the paper is presented in Section 2. In Section 3, the proposed modeling formalism is introduced. In Section 4, the deterministic nature of the discrete abstractions is discussed. Algorithms for the computation of the discrete approximations are discussed in Section 5. Finally, the regulator problem for hybrid systems is formulated in Section 6 and it is illustrated using the robotic manufacturing system example.

2 Hybrid System Example

A robotic manufacturing system (RMS) will be used to illustrate our approach. The system shown in Fig. 1 consists of two robots whose task is to move components periodically from a *parts bin* to an *assembly area.* Each robotic arm is driven by an armature-voltage-controlled DC servomotor. Each servomotor is controlled by a local controller and the overall system is monitored and coordinated by a supervisor. The supervisor communicates with the local controllers via a standard computer network. The system described above is an experimental setup in the control lab at the University of Notre Dame.

Our objective is the design of decision and control algorithms that guarantee the safe and efficient operation of the RMS. The control objective is to supervise the actions of the robotic arms to ensure that they will not enter the parts bin at the same time. The parts bin represents the critical section of the system and it is described by $|\theta_i| \leq 0.1$ where θ_i is the angular position of the ith robotic arm. This problem is simple enough to be described here, but also rich enough to demonstrate our approach. The safety requirement can be addressed using numerous approaches based on

[1]The partial financial support of the National Science Foundation (ECS95-31485) and the Army Research Office (DAAG55-98-1-0199)is gratefully acknowledged.

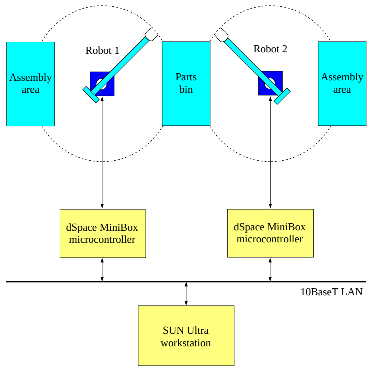

Figure 1: Robotic manufacturing system

logical models only that can result in acceptable solutions. However, there is an important need to investigate additional approaches that take into consideration the continuous dynamics in order to explore all possible solutions.

3 Hybrid system modeling

In this section, the proposed modeling formalism for hybrid systems is presented. The foundation of the model is the *set-dynamical system* [5]. A *set-dynamical system* (SDS) is denoted as $(X, A, D, Y, M; f, g, m)$ where X is the state set of the system, A is set of control actions, D is the set of disturbances, Y is the output set, M is the measurement set, $f : X \times A \times D \to X$ is the state transition function, $g : X \times A \times D \to Y$ is the output function, and $m : X \times A \times D \to M$ is the measurement function.

A *hybrid dynamical system* (HDS) is an SDS where the constituent sets consist of a continuous and a discrete part. We assume that the continuous part is a subset of a finite dimensional vector space and that the discrete part is finite. The advantage of such a representation is that, although it is simple, it provides the tools for interconnecting heterogeneous systems via input-output maps and for abstracting parts of the processes using equivalence relations.

In this paper, we restrict ourselves to a class of systems that is characterized in the literature as *piecewise-linear systems* [6, 8] to facilitate the development of analysis and synthesis tools. These systems arise when the state set and/or the input set are partitioned into regions described by linear equalities and inequalities and the dynamics at each region are described by linear (or affine) state transitions. Output and measurement maps can be defined also in a similar way. The class of piecewise-linear systems is quite general as it includes linear systems, finite state machines, and their interconnections [8].

Typical control specifications for hybrid systems are safety requirements that are usually formulated with respect to a partition of the state space of the system. A partition can be described by an equivalence relation on the state space and is represented by a projection function that assigns every state to its equivalence class. We assume that the partition defined by this map is appropriate for extraction of important information for the system and it will be called the *primary partition*.

Consider the state set X of a SDS and define the mapping $\pi : X \to \mathbb{P}(X)$ from X into the power set of X. The mapping π defines an equivalence relation E_π on the set X in the natural way $x_1 \, E_\pi \, x_2 \quad \text{iff} \quad \pi(x_1) = \pi(x_2)$. The image of the mapping π is called the *quotient space* of X by E_π and is denoted by X/E_π. Adopting this notation we can write $\pi : X \to X/E_\pi$ where π is understood as the *projection* of X onto X/E_π. The mapping π generates a partition of the state set X into the equivalence classes of E_π and will be called *generator*.

More specifically, in this paper we are interested in the case when $X = \mathbb{R}^n$ and the generator is defined by a set of hyperplanes in $\mathbb{R}^n$. This assumption is of significant practical importance since piecewise-linear regions arise in many applications. First, consider the collection $\{h_i\}_{i=1,2,\ldots,\ell}$, $h_i : \mathbb{R}^n \to \mathbb{R}$ of real-valued functions of the form $h_i(x) = g_i^T x - w_i$, $i = 1, 2, \ldots, \ell$ where $g_i \in \mathbb{R}^n$ and $w_i \in \mathbb{R}$. Let $H_i = \ker(h_i) = \{x \in \mathbb{R}^n : h_i(x) = g_i^T x - w_i = 0\}$ and assume that H_i is an $(n-1)$-dimensional hyperplane ($\nabla h_i(x) = g_i^T \neq 0$). We define the function $\hat{h}_i : \mathbb{R}^n \to \{-1, 0, 1\}$ by

$$\hat{h}_i(x) = \begin{cases} -1 & if \quad h_i(x) < 0 \\ 0 & if \quad h_i(x) = 0 \\ 1 & if \quad h_i(x) > 0 \end{cases} \qquad (1)$$

Then, the generator is defined by $\pi(x) = [\hat{h}_1(x), \ldots, \hat{h}_\ell(x)]^T$. Although the generator has been defined as $\pi : \mathbb{R}^n \to \{-1, 0, 1\}^\ell$ there is a bijection between $\{-1, 0, 1\}^\ell$ and the quotient set X/E_π (they are the same set). Since any other symbols could be used in (1) to identify the region the continuous state lies in, the image of the generator will be denoted as X/E_π. By the previous construction, the hybrid system specifications can be expressed as requirements of the *output* of the hybrid system $y(k) = \pi(x(k)) \in X/E_\pi$ that represents the equivalence class of the state $x(k)$.

Suppose that at time k we have that $y(k) = \pi(x(k)) \in X/E_\pi$. The signals $x(k)$ and $y(k)$ represent the state and the output of the system respectively at the kth successive iteration of the system. If it is agreed that the granularity of the partition generated by the mapping π is appropriate for the extraction of useful information regarding the system's behavior, then it is desirable to uniquely determine the state at the next iteration up to its membership on an equivalence class $y(k+1) = \pi(x(k+1)) \in X/E_\pi$. This can be accomplished by considering a finer partition than the partition defined by the generator π to obtain better estimates for the continuous state. This partition will be called the *final partition*.

The generator π_F is defined in a similar way as the output function π. Given a partition defined by a finite set of $(n-1)$-dimensional hyperplanes the generator $\pi_F : X \to X/E_{\pi_F}$ separates the state space into a finite number of equivalence classes which correspond to polyhedral regions in $\mathbb{R}^n$. The function $z = \pi_F(x)$ can be seen as a *measurement function* that provides some information about the continuous state. Intuitively, our ability to make decisions to influence the behavior of the system depend on the amount of information contained in the measurement signal.

Example The transfer function of the servomotor and the lever arm is $\frac{\theta(s)}{v_{in}(s)} = \frac{1}{s(0.0026s+0.1081)}$. The output of the system is the angular position of the robotic arm with respect to a fixed reference system. The parts bin corresponds to $\theta = 0$ and the assembly area to $\theta = \pi$.

The open loop position response of the servomotor is unstable due to the pole at the origin. First, we design local controllers for each servomotor so that each task is performed in an acceptable manner. Another control objective for the specific system is to protect the servomotor for high frequency voltages that can eventually damage the gearbox or the brushes. Conventional control methods are applied to force each robotic arm to follow a reference trajectory with satisfactory performance. The control algorithms are implemented in the microcontrollers. The closed loop system consisting of the servomotor and its local controller is described by a sampled data system with sampling period $T = 0.01s$.

There are three available reference inputs with corresponding controllers associated with the commands of the supervisor. A command `goto_parts_bin` issued by the supervisor is translated into the reference signal $r_1 = 0$ representing the angular position when the arm is at the parts bin. The controller used for this task allows a fast response of the system. Similarly, for the second task `goto_assembly_area`, the corresponding reference signal is $r_2 = \pi$ and a more conservative controller is used to guarantee an overdamped response in order to protect sensitive workpieces. The last command available to the supervisor is `stop`. Here, it is assumed that no brake command is available and that the arm can stop only because of its natural damping.

In order to coordinate the actions of the robotic arms to ensure that they will not enter their critical section (parts bin) simultaneously, the mathematical model of the system must include the dynamics of the two servomotors and take into consideration the effects of the computer network. For that, we consider that the model of the physical process seen by the supervisor is evolving at a slower time rate ($T_d = 0.1s$). In addition, a disturbance term has been included in the state space representation of each subsystem to take into account the stochastic nature of the time delays, since, practically, when the message carrying the angular position of the arm reaches the workstation over the network, the actual position will have changed.

The RMS is described by the hybrid system with states $x_1 = \theta_1, x_2 = \dot{\theta}_1, x_3 = \theta_2$, and $x_4 = \dot{\theta}_2$. The control actions available to the supervisor are $A = \{\texttt{goto\_parts\_bin}, \texttt{goto\_assembly\_area}, \texttt{stop}\}_{robot1} \times \{\texttt{goto\_parts\_bin}, \texttt{goto\_assembly\_area}, \texttt{stop}\}_{robot_2}$. For simplicity, we will represent this set by $A = \{a_{11}, a_{12}, a_{13}, a_{21}, a_{22}, a_{23}, a_{31}, a_{32}, a_{33}\}$ where a_{ij} corresponds to the ith task for robot 1 and the jth task for robot 2 (in the above order). $D \subset \mathbb{R}^4$ is a bounded polytope described by $|d_i| \leq 0.1,\ i = 1, 2, 3, 4$. For fixed control action a_{ij}, the state transition is

$$x(k+1) = \begin{bmatrix} A_i & 0 \\ 0 & A_j \end{bmatrix} x(k) + \begin{bmatrix} B_i & 0 \\ 0 & B_j \end{bmatrix} \begin{bmatrix} r_i \\ r_j \end{bmatrix} + B_d d(k) \tag{2}$$

where $B_d = I_4$. $Y = X/E_\pi$ is the quotient space induced by the generator π. As it was described earlier, the mapping π is determined by the control specification, which for the RMS can be expressed using the inequalities $|\theta_i| < 0.1$. We define the affine functions $h_1(x) = x_1 + 0.1, h_2(x) = x_1 - 0.1, h_3(x) = x_3 + 0.1$, and $h_4(x) = x_3 - 0.1$. The generator $\pi : X \to X/E_\pi$ is then defined by the functions h_i. The final partition will be computed in Section 5.

4 Quasideterminism

In order to analyze hybrid systems and design control algorithms, it is desirable to induce dynamical systems in finite quotient spaces that preserve the properties of interest and then study the simplified models. Consider the state set X of an SDS and define the mapping $\pi : X \to X/E_\pi$. Let f be the state transition function of an SDS and assume that the inputs are fixed. Consider the diagram in Fig.2-(a). Intuitively, the map π is used to coarsen the state set of the system. The question that arises is whether the system f can follow this abstraction. This question is concerned with the existence of a mapping $\tilde{f} : X/E_\pi \to X/E_\pi$ that makes the diagram commute. It is shown in [5] that $\tilde{f}$ exists if and only if

$$x_1\, E_\pi\, x_2 \;\Rightarrow\; (\pi \circ f)(x_1) = (\pi \circ f)(x_2) \tag{3}$$

(where $\circ$ denotes function composition) and moreover, if (3) is satisfied then $\tilde{f}$ is unique. Note that the above result does not require any structure on the set X or the mappings π and f. Using equivalence relations on the state set X, it is possible to define new dynamical systems in the derived quotient spaces. These systems are called *induced dynamical systems*.

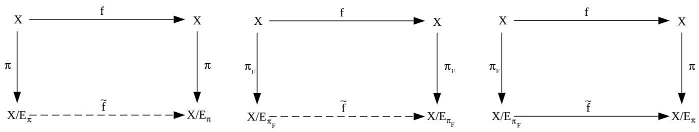

Figure 2: Quasideterminism and the partitions of the state space

In the hybrid system case, consider two states $x_1, x_2 \in X$, $x_1 \neq x_2$ such that $\pi(x_1) = \pi(x_2) = y \in X/E_\pi$. The states x_1 and x_2 may be driven under the mapping f to different equivalence classes of the quotient space X/E_π. Therefore, in general we have that $(\pi \circ f)(x_1) \neq (\pi \circ f)(x_2)$ and a mapping $\tilde{f}$ that makes the diagram in Fig. 2-(a)commute does not exist. The induced system defined by the mapping $\tilde{f} : X/E_\pi \to X/E_\pi$ can be viewed as a nondeterministic system. The nondeterminism of the approximating system has been identified as the main drawback in discretization methods for hybrid systems. Efforts to relax the commutativity requirement and still obtain useful partitions have been made in [1] and [9] and this has led to the concept of quasideterminism. Here, a similar idea is followed for piecewise-linear systems.

Suppose that at time k, $\pi(x(k)) = y(k) \in X/E_\pi$. In the case when the estimates of the state at time k provide sufficient information to uniquely determine the membership of the state of the induced system at time $k+1$ on an equivalence class of E_π, the system is said to be *quasideterministic*.

Recently, considerable attention has focused on the study of partitions of the continuous state space that preserve reachability properties. For example in [3] conditions for hybrid systems to admit finite bisimulations are formulated and finite bisimulations are computed for certain classes of systems. Bisimulations are essentially equivalence relations on the state set of the system that preserve the reachability properties and are computed by refining an initial partition. In our case, the final partition that makes the system quasideterministic is also computed by refining the primary partition, but it does not preserve the reachability properties. This can be seen from the diagram of Fig. 2-(b) which does not commute. Quasideterminism describes a weaker requirement where only the membership of the continuous state in an equivalence class of the primary partition in the next time step can be determined by examining the quotient system. Therefore, the best thing we can do by studying the finite quotient system is to determine the membership of the state in an equivalence class at a prescribed time interval (multiple of the sampling period). However, using only this information we can solve some interesting problems and design controllers for hybrid systems as illustrated in the remaining of the paper. In addition, it can be shown that a final partition with these properties can be always computed for discrete-time piecewise linear systems.

5 Computation of the Final Partition

In this section, we present the mathematical tools for the computation of the final partition. Consider the hybrid system $(X, A, D, Y, M; F, \pi, \pi_F)$ where $X = \mathbb{R}^n$, A is a finite set of control actions (or control modes) determining which subsystem is active, $D \subset \mathbb{R}^m$ is a polytope, and $F : X \times A \times D$ is the state transition function, which for fixed control action is described by $x(k+1) = \mathbf{A}x(k) + \mathbf{B}x(k) + \mathbf{B}_d d(k)$. The outputs and the measurements are as defined in Section 3.

Let $E(X)$ be the set of all equivalence relations on X. A partial order relation $\leq$ on $E(X)$ can be defined as $E_1 \leq E_2$ if $x_1 E_1 x_2 \Rightarrow x_1 E_2 x_2$. A lattice structure can be developed on the set of all equivalence relations on X by introducing meet and join operations and is called the *equivalence lattice* (for more details see [5]). It can be shown that the set $E_P(X)$ of all equivalence relations on X induced by mappings $\pi : X \to X/E_\pi$ which are defined using finite collections of $(n-1)$-dimensional hyperplanes and thus, they separate the state space X into polyhedral equivalence classes, is a sublattice of the equivalence lattice $E(X)$, and moreover that $E_P(X)$ is not complete.

A partition defined by the mapping π' is finer than the partition defined by π, if the induced equivalence relations considered as elements of the equivalence lattice satisfy the condition $E_{\pi'} \leq E_\pi$. The meet operation of the equivalence lattice will be used in order to refine the state space. Since $E_P(X)$ is a sublattice, the refinement of the state space will result in polyhedral equivalence classes. However, the fact that $E_P(X)$ is not complete implies that the algorithms for the computation of the final partition can use the meet operation only finite number of times.

In the following, it is shown how the refinement of the partition can be implemented in the case we want to ensure that the hybrid system satisfies a safety requirement. Given a set of safe states described by the set $P \subset \mathbb{R}^n$ and an initial condition $x_0 = x(0) \in P$, we say that the system is *safe* if $x(k) \in P$ for every k. In order to refine the state space, we define the predecessor operator $pre : \mathbb{P}(X) \rightarrow \mathbb{P}(X)$ as

$$pre(P) = \{\exists a \in A, \forall d \in D, f_a(x,d) = \bar{A}_a x + \bar{B} r + \bar{B}_{d_a} d \in P\}.$$

The set $pre(P)$ represents all the states x for which there is a control action that will enforce the state to remain in P for any disturbance d. If the set P is piecewise-linear (PL), that is P is the union of a finite number of sets defined by (finitely many) linear equations and linear inequalities, then the set $pre(P)$ is also piecewise-linear and can be defined using only propositional connectives. For a proof of this claim the reader is referred to [7], where it is shown that every PL set in this framework defined using quantifiers, can be also defined using only propositional connectives.

The set of states for which the safety requirement will not be violated is calculated using the following algorithm.

$P^0 = P;\ Q^0 = pre(P);\ i = 0;$
while $P^i \setminus Q^i \neq \emptyset$ do
    $P^{i-1} = P_i \cap Q^i;\ Q^{i-1} = pre(P^{i-1});\ i = i - 1;$
end

At each step of the algorithm, the set Q^i contains all the states for which there exists control action which will ensure that the system's state is in P^i. If the algorithm terminates in a finite number of N steps and $P^{-N} \cap Q^N \neq \emptyset$ the hybrid system is safe with respect to the PL set P. The described procedure, which given a hybrid system and a piecewise-linear region checks if the system is safe, is *semi-decidable*, that is if it terminates it gives the correct answer, but it's termination is not guaranteed.

An important advantage of the approach is that it allows to formulate conditions on the control actions, the system description, and the safe set that guarantee the termination of the above algorithm. This is useful when we have the ability to change each control mode, for example by selecting different continuous controllers for each subsystem. At each iteration both sets P^i and Q^i are PL sets and therefore can be described by formulas without quantifiers. Therefore, for a given N the set $P^{-N} \cap Q^N$ can be described without quantifiers and we can enforce the termination condition by selecting appropriately the control actions. However, note that for fixed control actions, the existence of a finite number N for which the algorithm will terminate is not guaranteed.

Similar computational procedures can be formulated for a region to be reachable or for a system to be deadlock-free. Continuous inputs at control mode of the system and/or uncontrolled discrete inputs can be also incorporated in the framework by modifying appropriately the predecessor operator. Additional details are omitted due to length limitations.

Example Let P denote the safe set for the RMS, which all the states except those in the critical section. The hyperplanes that bound $pre(P)$ can be computed by the following algorithm where equation (2) is denoted by $x(k+1) = \mathbf{A}x + \mathbf{B}r + B_d d,$.

INPUT: $f(x,d) = \mathbf{A}x + \mathbf{B}r + B_d d,$
    $H_i = \{x \in \mathbb{R}^n : h_i(x) = g_i^T x - w_i = 0\}$
for $i = 1 \ldots, \ell$
    $\underline{d} = argmin_{d \in D}(-g_i^T \mathbf{B} d);$
    $g_i^{'T} = g_i^T \mathbf{A};$
    $w_i' = w_i - -g_i^T \mathbf{B} r - g_i^T B_d \underline{d};$
end
OUTPUT: $H_i' = \{x \in \mathbb{R}^n : h_i'(x) = g_i^{'T} x - w_i' = 0\}$

In order to explain the previous algorithm, assume first that the disturbance d is fixed. Then it can be shown that

$$H_i' = \{x \in \mathbb{R}^n : h_i'(x) = g_i^{'T} x - w_i' = 0\} \qquad (4)$$

where $g_i^{'T} = g_i^T \mathbf{A}$ and $w_i' = w_i - g_i^T \mathbf{B} r - g_i^T B_d d$. From equation (4) it follows that for fixed arbitrary inputs the hyperplanes H_i' are parallel. Selecting the input as $\underline{d} = argmin_{d \in D}(-g_i^T B_d d)$ corresponds to the worst case in view of the effect of the input $d \in D$. The above procedure is repeated for every hyperplane that bounds the critical section to compute the hyperplanes H_i'.

The above algorithm is repeated for every control action and every hyperplane of the generator π of the primary partition. The generator of the final partition π_F is defined by the original hyperplanes and all the computed ones by the above algorithm. For the robotic manufacturing system the safety algorithm terminated in only one iteration by selecting appropriately the control modes (practically by reducing the overshoot of each controller). Note that in the general case repetitive applications of the above procedure may be necessary.

6 Hybrid System Regulator

In this section, the regulator problem for hybrid systems is formulated. In general, a regulator requests certain types of outputs from the plant so that these are attained in the presence of disturbances. The desired outputs are characterized by a regulation condition and they can be described as the outputs of another SDS, called the *exosystem*. The plant and the exosystem are linked by a controller to form a regulator as shown in Fig. 3. A feedback controller can be designed to regulate the system. The main characteristic of the controller is that it contains a copy of the exosystem in accordance to the "internal model principle".

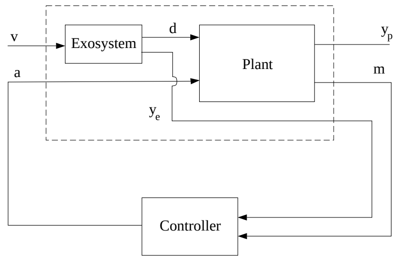

Figure 3: Hybrid system regulator

Example The exosystem, represented an SDS, is specified by the designer in such a way as to characterize the output requests and the disturbances from the environment. For example, the exosystem for the robotic manufacturing system is defined as follows. Denote $\tilde{x}_d$ the equivalence class in X/E_π that corresponds to the unsafe region. The exosystem is described by the SDS $\mathcal{E} = (X_e, D, Y; f_e, g_e, m_e)$ where $X_e = X/E_\pi, Y = X/E_\pi, f_e : X/E_\pi \to X/E_\pi$ such that for every $\tilde{x}_e \in X/E_\pi,\ f_e(\tilde{x}_e) \neq \tilde{x}_d$, and $m_e : X/E_\pi \to X/E_\pi$ such that $m_e(\tilde{x}_e) = \tilde{x}_e$. The disturbance function is $g_e : X/E_\pi \to D$ and for every $\tilde{x}_e \in X/E_\pi$ returns an arbitrary disturbance signal in D. The exosystem corresponds to a finite state machine and therefore is a PL system.

Since the output set $Y = X/E_\pi$ is defined by the generator of the primary partition which characterizes the control specifications, the regulation condition can be described by $y = \pi(x) = m_e(x_e, v)$. The objective is to construct a controller to satisfy this condition. The output function of the controller establishes a feedback link from the measurements of the extended plant to its control actions.

The controller for the RMS is described by $\mathcal{C} = (X_c, A \times M, A; f_c, g_c)$ where $X_c = X/E_{\pi_F}, Y \times M = E_\pi \times E_{\pi_F}$, and $A = \{1, 2, 3\}$. The current controller state $\tilde{x}_c \in X_c$ implies the membership of the continuous state of the plant to the corresponding region of the final partition. This information is updated at every time step using the measurements from the plant. Since we assume that the system is quasideterministic, for every controller state $\tilde{x}_c \in X_c$ and every control action $a \in A$ the membership of the continuous state to a region of the primary partition can be uniquely defined. The set of states for which the safety requirement will not be violated has been computed for the RMS in Section 5. If the initial state belongs to that set, then it is guaranteed that for every region in X/E_{π_F} there exists an appropriate control action that will satisfy the safety specification. The output function of the controller $g_c : X_c \times (Y \times M) \to A$ returns exactly this control action. The controller can be combined with the plant and the exosystem to form a regulator as shown in Fig. 3, that guarantees that the regulation $y = \pi(x) = m_e(x_e, v)$ is satisfied. The regulator for the RMS has been implemented using Simulink and Stateflow. Here, the output request of the exosystem is constant. Problems where the output request function changes with time or upon occurrence of events can be still formulated in the proposed framework of the hybrid system regulator and they are topics for current research.

7 Conclusions

The proposed approach provides a general framework for hybrid systems for analysis and controller synthesis. The notion of quasideterminism is used to characterize discrete abstractions that can be used for control design and the regulator problem for hybrid systems is formulated.

References

[1] P. Antsaklis, J. Stiver, and M. Lemmon. Hybrid system modeling and autonomous control systems. In R. L. Grossman et al. editors, *Hybrid Systems*, Vol. 736 of *LNCS*, p. 366–392. Springer-Verlag, 1993.

[2] P. Caines and Y.-J. Wei. Hierarchical hybrid control systems: A lattice formulation. *IEEE Transactions on Automatic Control*, 43(4):501–508, 1998.

[3] G. Lafferriere, G. Pappas, and S. Sastry. Reachability analysis of hybrid systems using bisimulations. In *Proc. of the 37th IEEE Conference on Decision and Control*, p. 1623–1628, Tampa, FL, 1998.

[4] J. Raisch and S. O'Young. Discrete approximation and supervisory control of continuous systems. *IEEE Transactions on Automatic Control*, 43(4):568–573, 1998.

[5] M. Sain. *Introduction to Algebraic System Theory*. Academic Press, 1981.

[6] E. Sontag. Nonlinear regulation: The piecewise linear approach. *IEEE Transactions on Automatic Control*, 26(2):346–358, 1981.

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

[8] E. Sontag. Interconnected automata and linear systems: A theoretical framework in discrete-time. In R. Alur et al. editors, *Hybrid Systems III, Verification and Control*, Vol. 1066 of *LNCS*, p. 436–448. Springer, 1996.

[9] J. Stiver. *Analysis and design of hybrid control systems*. PhD thesis, Department of Electrical Engineering, Univ. of Notre Dame, Notre Dame, IN, 1995.

[10] J. Stiver, P. Antsaklis, and M. Lemmon. A logical DES approach to the design of hybrid control systems. *Mathl. Comput. Modelling*, 23(11/12):55–76, 1996.