Hybrid Control Systems Using Timed Petri Nets: Supervisory Control Design Based on Invariant Properties*

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 class of timed Petri nets named programmable timed Petri nets is used for supervisory control of hybrid systems. In particular, the transfer of the continuous state to a region of the state space under safety specifications on the discrete and continuous dynamics is addressed. The switching policy is embedded in the dynamics of the underlying Petri net structure and the supervisors are described by Petri nets. The discrete specifications are expressed in terms of linear constraints on the marking vector and are satisfied by applying supervisory control of Petri nets based on place invariants. The hybrid system switches from a subsystem to another, in a way that the state gradually progresses from one equilibrium to another towards the desired target equilibrium. The supervisory control algorithm is designed to allow switchings to occur only on the intersection of the invariant manifolds. Finally, in the case when the continuous dynamics are described by first order integrators, the design algorithm is formulated as a linear programming problem.

1 Introduction

In hybrid systems the behavior of interest is governed by interacting continuous and discrete dynamic processes. Hybrid control systems typically arise from the interaction of discrete planning algorithms and continuous processes, and their study is essential in designing discrete event supervisory controllers for continuous systems, and central in designing intelligent control systems with a high degree of autonomy. The investigation of hybrid systems is creating a new and fascinating discipline bridging control engineering, mathematics and computer science; further information on hybrid systems may be found in references [1–6]; see also the survey paper [7].

This paper considers systems that arise when computers are used to supervise or synchronize the actions of subsystems described by continuous dynamics

* This work was supported in part by the National Science Foundation grant ECS95-31485.

** This author gratefully acknowledges the fellowship of the Center of Applied Mathematics at the University of Notre Dame for the academic year 1997-98.

(that involve continuous variables). Examples of such systems arise in chemical process control, command and control networks, power distribution networks, as well as distributed manufacturing systems. The size and complexity of such systems often requires that the system use a number of distinct operational modes. Consequently, these systems can be viewed as *supervised* systems, in which a high-level discrete (-event) supervisor is used to coordinate the actions of various subsystems so that overall system safety is guaranteed. By *safety*, we mean that pre-specified limits or tolerances on the subsystem states are not violated. In the paper, the sets of safe states are characterized by Lyapunov functionals and are related to stability properties of the subsystems. So, these systems can be viewed as a *hybrid* mixture of systems with continuous dynamics (continuous variables) supervised by a switching law generated by a (discrete-event) supervisor described by discrete dynamics (discrete variables).

Petri nets have been used extensively as a tool for modeling, analysis and synthesis for discrete event systems. For DES control, Petri nets modeling formalism offers some advantages over finite automata, and it is also useful for hybrid systems control. Peleties and DeCarlo [8] presented a model based on the work in [9] on the periodicity of symbolic observations of piecewise smooth discrete-time systems. This hybrid model is suitable for Petri net based symbolic analysis of hybrid systems; the continuous plant is approximated by a Petri net and a supervisor consisting of two communicating Petri nets controls the behavior of the open plant. Lunze *et al.* [10] proposed a model where Petri nets are used as a discrete event representation of the continuous variable system; the system and the interface are represented by a Petri net and the supervisor represents a mapping of the output event sequence into the input event sequence. Several other approaches to modeling of hybrid systems that use Petri nets have also been reported in the literature [11–16].

In this paper, a class of timed Petri nets named *programmable timed Petri nets* [17] is used to model hybrid systems. In particular, it is assumed that the switching policy is embedded in an underlying Petri net structure and that the supervisors are described also by Petri nets. Petri nets are used instead of finite automata because of the following two reasons. The first is the expressiveness of Petri nets. Petri net languages include regular languages described by finite automata and further, they can model switching policies that describe conflict, concurrency, synchronization, and buffer sizes. The second reason is that recent results in the supervisory control of discrete-event systems using ordinary Petri nets [18] have made possible to design supervisors in an efficient and transparent manner; and this methodology is used in this paper.

In the nonlinear control literature, switching has been used to expand the domain of attraction of a control system [19, 20]. Here, it is assumed that the continuous subsystems admit a family of equilibria and each equilibrium has a domain of attraction associated with it. The hybrid system switches from a subsystem to another, in a way that the state gradually progresses from one equilibrium to another towards the desired target equilibrium. For the hybrid systems of interest in this paper, this idea can be formalized using *an invariant*

based approach to the design of hybrid systems [21, 22]. This approach introduces the notion of a *common flow region* , which is defined as the set of states which can be driven to the target region with the same control policy, and gives sufficient conditions for a set of invariant manifolds to bound common flow regions. In this paper, such invariant manifolds are determined by appropriate Lyapunov functions. The switchings are allowed to occur only if the continuous state lies on the intersection of those invariant manifolds. Since the switching logic is described by a Petri net, only sequences of invariant manifolds that satisfy the discrete specifications have to be considered.

The paper is organized as follows. Section 2 presents programmable timed Petri nets which are used in section 3 to model hybrid control systems. In Section 4 we discuss in detail a Petri net approach to hybrid control which emphasizes supervisory control of hybrid systems and we give a simple illustrative example. Note that related work has appeared in [23, 24].

2 Programmable Timed Petri Nets

Programmable timed Petri nets were introduced in [17] and are used to generate the switching logic of the hybrid system. In particular, a programmable timed Petri net (PTPN) is a timed Petri net whose places, transitions, and arcs are all labeled with formulae representing constraints and reset conditions on the rates and times generated by a set of continuous-time systems called *clocks*. The model can seen as an extension of the Alur-Dill hybrid automaton model [25, 26].

An *ordinary Petri net structure* [27–29] is the 4-tuple $\mathcal{N} = (P, T, I, O)$ where P is a finite set of places, T is a finite set of transitions, $I \subset P \times T$ is a set of input arcs (from places to transitions), and $O \subset T \times P$ is a set of output arcs (from transitions to places). The *preset* and *postset* of a place p are defines by $\bullet p = \{t \mid (t,p)\} \in O$ and $p\bullet = \{t \mid (p,t) \in I\}$. The *preset* and *postset* of a transition t are defined similarly as $\bullet t = \{p \mid (p,t) \in I\}$ and $t\bullet = \{p \mid (t,p) \in O\}$.

The *marking* of a Petri net is a mapping $\mu : P \to \mathbb{Z}^+$ from the set of places onto the nonnegative integers which assigns to each place p a number of tokens $\mu(p)$. The marking can be represented also by an n_p-dimensional vector $\mu = (\mu_1, \ldots, \mu_p)$, where $n_p = |P|$. The vector μ gives for each place p_i, the number of tokens in that place, $\mu_i = \mu(p_i)$. To avoid confusion, the marking μ is interpreted as a mapping when it is appeared with an argument and as a vector of nonnegative integers otherwise. The dynamics of ordinary Petri nets are characterized by the evolution of the marking vector which is referred to as the *state* of the net.

The transition t is *enabled* when each one of its input places is marked with at least one token, $\mu(p) > 0$ for all $p \in \bullet t$. An enabled transition may fire. The transition t *fires* by removing one token from each one of its input places and by placing one token to each one of its output places. If $\mu(p)$ and $\mu'(p)$ denote

the marking of place p before and after the firing of enabled transition t, then

$$\mu'(p) = \begin{cases} \mu(p) + 1 \text{ if } p \in t\bullet \setminus \bullet t \\ \mu(p) - 1 \text{ if } p \in \bullet t \setminus t\bullet \\ \mu(p) \qquad \text{otherwise} \end{cases} \tag{1}$$

The firing of the transition t is described by the firing function $q : T \to \{0, 1\}$ such that $q(t) = 1$ if t is firing and $q(t) = 0$ otherwise. In untimed Petri nets one can prohibit controlled transitions from firing, but cannot force the firing of a transition at a particular instant. In a timed Petri net controlled transitions are forced to fire, as this can be accomplished by considering the firing functions to be functions of a global time. For the timed Petri net, the firing of a transition occurs over a time interval $[\tau_0, \tau_f]$. The length of this interval is called the transition's *holding time*. A transition t which starts to fire at time τ_0 is said to be *committed*. During the time that the transition is committed, the network's marking vector is not changed. It is only when the firing is completed at time τ_f that the marking vector is changed according to equation (1) given above.

The holding times can be seen as control variables. They can be controlled by specifying conditions which cause transitions to fire. The conditions that characterize the holding times are represented by logical propositions defined over a set of vector dynamical equations, which can be seen as a set of *local clocks*.

Consider the set, $\mathcal{X}$, of N *local clocks* where the ith clock $\mathcal{X}_i$ is denoted by the triple $(\dot{x}_i, x_{i0}, \tau_{i0})$. $x_{i0} \in \Re^n$ is a real vector representing the clock's offset. τ_{i0} is an initial time (measured with respect to the global clock) indicating when the local clock was started. $\dot{x}_i : \Re^n \to \Re^n$ is a Lipschitz continuous automorphism over $\Re^n$ characterizing the local clock's rate. It is assumed that the clock rate $\dot{x}_i$ is denoted by the automorphism f. The *local time* generated by the ith clock will be denoted as x_i which is a continuous function over $\Re^n$ that is the solution to the following initial value problem for $\tau > \tau_{i0}$,

$$\frac{dx_i}{dt} = f(x_i) \tag{2}$$

$$x_i(\tau_{i0}) = x_{i0}. \tag{3}$$

The *state* of the ith timer is the ordered pair $z_i(\tau) = (x_i(\tau), \dot{x}_i(t))$. The ensemble of all the local clock states will denoted by $z(\tau)$.

The interval $[\tau_0, \tau_f]$ over which a transition t will be firing is characterized by conditions on the "local time" $x_i(\tau)$ and the "clock rate" $\dot{x}_i(t)$ of the ith timer. These conditions are described by formulae in a propositional logic whose atomic formulas are equations over the local times or clock rates of $\mathcal{X}$. In the next section, the local clocks are used to describe the continuous dynamics of the hybrid systems at each operational mode. The hybrid system switches modes based on constraints on the continuous states. The atomic and the propositional formulas which will be used to describe the conditions on the states of the vector dynamical equations are defined next. Their form is general enough to describe a variety of constraints which will be used to characterize the evolution of the hybrid system in the next section.

Definition 1. *An* atomic formula, *p, takes one of the following forms;*

1. *It can be a* time constraint *of the form $h(x_i) = 0$ or $h(x_i) < 0$ where $h : \Re^n \to \Re$ is a real valued function. This formula means that the clock time x_i satisfies the equation.*
2. *The atomic formula p can be a* rate constraint *of the form $\dot{x}_i = f$ which means that the ith clock's rate $\dot{x}_i$ is equal to the vector field $f : \Re^n \to \Re^n$.*
3. *Finally, p can be a* reset equation *of the form $x_i(\tau) = \bar{x}_0$ which says that the ith clock's local time at global time τ is set to the vector $\bar{x}_0$.*

Definition 2. *A* well-formed formula *or* WFF *is defined as any expression generated by a finite number of application of the following rules;*

1. *Any atomic formula is a WFF,*
2. *If p and q are WFFs, then $p \wedge q$ is a WFF.*
3. *If p is a WFF, then $\tilde{}p$ is a WFF.*

The set of all WFFs formed in this manner will be denoted as $\mathcal{P}$. Consider an ordinary Petri net, $\mathcal{N} = (P, T, I, O)$ and a set of logical timers, $\mathcal{X}$. A *programmable timed Petri net* (PTPN) is denoted by the ordered tuple, $(\mathcal{N}, \mathcal{X}, \ell_P, \ell_T, \ell_I, \ell_O)$ where the functions $\ell_P : P \to \mathcal{P}$, $\ell_T : T \to \mathcal{P}$, $\ell_I : I \to \mathcal{P}$, and $\ell_O : O \to \mathcal{P}$ label the places, transition, input arcs, and output arcs (respectively) of the Petri net $\mathcal{N}$ with WFFs in $\mathcal{P}$.

The *syntax* for well formed formulas is defined with respect to the underlying Petri net structure of the form $\mathcal{N} = (P, T, I, O)$ and the set of local clocks $\mathcal{X}$. The local clock state z at time τ is said to *satisfy* a formula $p \in \mathcal{P}$ if p is "true" for the given clock state, $z(\tau)$. The truth of the well-formed formed formulas is understood in the usual sense.

3 Hybrid Control Systems

In this section, programmable timed Petri nets are used to model hybrid systems. The hybrid control systems of interest in this paper are described by the following equations

$$\dot{x} = f_{i(t)}(x(t)) \tag{4}$$

$$i(t) = q(x(t), i(t^-)) \tag{5}$$

where $x(t) : \Re \to \Re^n$ and $i : \Re \to \mathbb{Z}^+$ denote the continuous and discrete states of the system, respectively. The continuous dynamics are controlled by a finite collection of N control strategies

$$\mathcal{D} = \{f_1, f_2, \cdots f_N\} \tag{6}$$

where $f_i : \Re^n \times \Re^m \to \Re^n$ for $i \in \{1, \ldots, N\}$ are locally Lipschitz continuous functions. The discrete state of the system is controlled by a *successor* function

$q : \Re^n \times \mathbb{Z}^+ \to \mathbb{Z}^+$ which determines the next possible discrete state $i(t)$ at time t given the current continuous state and the "previous" discrete state $i(t^-)$.

We introduce now some additional notation that will be useful in formulating the control algorithms later in the section. The firing times of transition t are described by $\sigma^t(n),\ n \in \mathbb{Z}^+$, where $\sigma^t(k) \in \Re$ represents the duration of the k^{th} firing of transition t. During the time interval $\sigma^t(k)$ the tokens of the input places of transition t do not change. These tokens are put into the output places of t upon the completion of firing of the transition, according to the enabling condition of the untimed Petri net. We assume that at each time instant exactly one transition is firing. In addition, we assume that $0 < \Delta \leq \sigma^t(n) < \infty$, for some $\Delta \in \Re$, for all firings n and transitions t. We may easily incorporate in our model instantaneous transitions, but these correspond to jumps in the continuous state and will not be considered here. The assumption $0 < \Delta \leq \sigma^t(n)$ eliminates the possibility of infinitely many switchings in a finite time interval.

The control algorithm for the mode selection problem will be based on structural information associated with the places of the Petri net. Routing policies for timed Petri nets are used usually for resolution of conflicts and were introduced in [30]. In our case, we define a mapping $\nu^p(n) :\ \mathbb{Z}^+ \to T$ for each place $p \in P$, where $\nu^p(k)$ identifies the particular transition $t \in p\bullet$ to which the k^{th} token to enter place p is to be routed. Note that more than one transition is enabled but only one is allowed to actually fire. If the k^{th} token is routed to $t \in p\bullet$, then the transition t wins the token, which after a firing time of $\sigma^t(k)$ is routed to $t\bullet$, the output places of the transition.

Next, a firing event is defined as the pair (t, τ) which denotes that the transition t starts firing at time τ. Consider the sequence of firing events

$$s = (t_{i_0}, \tau_0), (t_{i_1}, \tau_1), \dots\ ,\ \ i_j \in \{1, \dots, N\},\ \text{for } j = 0, 1, 2, \dots \tag{7}$$

where j denotes the ordering of the transitions that fire. For example $s = (t_1, \tau_0), (t_3, \tau_1), \dots$ denotes that t_1 fires at τ_0, next t_3 fires at τ_1 and so on. The firing time intervals are defined by the equation

$$\sigma^{t_i}(k) = \tau_{k+1} - \tau_k \tag{8}$$

At the k^{th} firing of the network, the transition t_i starts firing (at time τ_k) for $\sigma^{t_i}(k)$ time units (until τ_{k+1}). The continuous state of the system during this interval evolves according to

$$\dot{x}(\tau) = f_i(x(\tau)),\ \text{for } \tau_k \leq \tau < \tau_{k+1}. \tag{9}$$

The *event projection* and the *timed projection* of the sequence s are defined as

$$\pi_1(s) = i_0, i_1, i_2, \dots \tag{10}$$

$$\pi_2(s) = \sigma^{t_{i_0}}(k_0), \sigma^{t_{i_1}}(k_1), \dots \tag{11}$$

These are used later in this section.

4 Supervisor

The supervisor has two main tasks. The first task is to allow only sequences of events that satisfy specifications imposed on the discrete-event part of the hybrid plant. In particular, consider the net $\mathcal{N}$ of the hybrid system. The objective here is to restrict the possible mode switches of the systems to satisfy additional logical constraints (for example mutual exclusion constraints) that have not been taken into consideration in the modeling phase of the hybrid plant. This can be accomplished without any information about the continuous dynamics. The differential equations of the continuous subsystems associated with the transitions are used to label these transitions. The second task of the supervisor is to enforce firing times that satisfy specifications on the continuous state of the plant. In untimed Petri nets one can prohibit transitions from firing, but cannot force the firing of a transition at a particular instant. In a timed Petri net controlled transitions are forced to fire, as this can be accomplished by considering the firing vectors to be functions of the global time τ. We will show that for a special class of problems, we can first determine the routing policy and then the firing times that will not violate certain conditions imposed on the continuous dynamics. These conditions will be expressed as well formed formulas labeling the input and output arcs of the Petri net.

4.1 Supervisor Control of Petri Nets Based on Place Invariants

The first step is to satisfy the discrete specifications of the hybrid plant by applying DES control methods. We assume that the discrete specifications are described by linear inequalities on the marking vector of the Petri net. A methodology for DES control based on Petri net place invariants has been proposed in [18]. A feedback controller based on place invariants is implemented by adding control places and arcs to existing transitions in the Petri net structure. Although the method was developed for ordinary Petri nets, the introduction of time delays associated to each transition will not affect the controlled behavior of the Petri net with respect to the discrete specifications. The supervisor is used to enforce a set of linear constraints on the discrete state of the hybrid plant. These constraints can describe a broad variety of problems including forbidden state problems, mutual exclusion problems, a class of logical predicates on plant behavior [31], conditions involving the concurrence of events, and the modeling of shared resources.

The system to be controlled is the untimed Petri net $\mathcal{N} = (P, T, I, O)$, which is called the plant net. We assume that the plant net has n places and m transitions and its incidence matrix is D_p. The controller net is a Petri net with incidence matrix D_c made up of the transitions of the plant net and a separate set of places. The controlled net is the Petri net with incidence matrix D made up of both the plant and the controller net. The control objective is to enforce the discrete state to satisfy constraints of the form

$$L\mu_p \leq b \tag{12}$$

where μ_p is the marking vector of the plant net, L is an $n_c \times n$ integer matrix, b is an $n_c \times 1$ integer vector, and n_c is the number of 1-dimensional constraints of the type $\sum_{i=1}^{n_p} l_i \mu_i \leq \beta$.

This inequality constraint can be transformed to an equality by introducing an external Petri net controller that contains places representing nonnegative slack variables. Then

$$L\mu_p + \mu_c = b \tag{13}$$

where μ_c is an n_c integer vector which represents the marking of the controller places. The structure of the controller net will be computed by observing that the introduction of the slack variables forces a set of place invariants on the controlled system. A *place invariant* is defined by an integer vector x that satisfies

$$x^T \mu = x^T \mu_0 \tag{14}$$

where μ_0 is the initial marking and μ any reachable subsequent marking. The place invariants of a net are elements of the kernel of the net's incidence matrix, and they can be computed by finding integer solutions to

$$x^T D = 0 \tag{15}$$

where D is an $n \times m$ incidence matrix. The matrix D_c contains the arcs that connect the controller places to the transitions of the plant net. The incidence matrix D of the closed loop system is given by

$$D = \begin{bmatrix} D_p \\ D_c \end{bmatrix} \tag{16}$$

and the marking vector μ and the initial marking μ_0 are given by

$$\mu = \begin{bmatrix} \mu_p \\ \mu_c \end{bmatrix} \quad \mu_0 = \begin{bmatrix} \mu_{p_0} \\ \mu_{c_0} \end{bmatrix} \tag{17}$$

Note that equation (13) is in the form of (14), thus the invariants defined by equation (13) on the system (16),(17) must satisfy equation (15).

$$X^T D = [L, I] \begin{bmatrix} D_p \\ D_c \end{bmatrix} = 0 \tag{18}$$

$$LD_p + D_c = 0 \tag{19}$$

If D_c is chosen as the solution of equation (19), then the rows of $[L, I]$ are elements of the kernel of the net's incidence matrix. Therefore, they represent place invariants of the closed loop systems and equation (13) is satisfied. Since $\mu(p) > 0$ for all $p \in P$, inequality (12) holds componentwise. The above analysis leads to the following proposition presented in [32].

Proposition 1. *The Petri net controller with incidence matrix D_c and initial marking μ_{c_0}, which enforces the constraints $L\mu_p \leq b$ when included in the closed loop system (16) with marking (17) is defined by*

$$D_c = -LD_p \tag{20}$$

with initial marking

$$\mu_{c_0} = b - L\mu_{p_0} \tag{21}$$

assuming that the transitions with arcs from D_c are controllable, observable, and that $\mu_{c_0} \geq 0$.

This proposition designs a controller that enforces the linear constraints $L\mu \leq b$ under the assumption that the controller will enable or inhibit only controllable and observable transitions. These results have been extended for handling uncontrollable and unobservable transitions in [33]. In the hybrid systems case, we have associated transitions to continuous subsystems described by differential equations. It is assumed that the supervisor can force and observe the firing of the transitions. This is accomplished by imposing conditions described by well-formed formulas on the input and output arcs of the transitions, as described in the next section.

4.2 Hybrid Strategy based on Equilibria

In the nonlinear control literature, switching has been used to expand the domain of attraction of control systems [19, 20]. In the hybrid systems case, we assume that the continuous part admits a family of equilibria corresponding to different symbolic inputs generated by the discrete event part. Each equilibrium has a domain of attraction associated with it. The idea is to switch at discrete time instants from one symbolic input to another in a way that the system gradually progresses from one equilibrium to another towards the final equilibrium.

This idea can be formalized using an *invariant based approach* for hybrid systems proposed in [21, 22]. A *common flow region* for a given target region, is defined as a set of states which can be driven to the target region with the same control policy. The approach as described by Stiver *et al.* considers common flow regions which are bounded by *invariant hypersurfaces*, *cap boundaries* and an *exit boundary*. Invariant hypersurfaces and cap boundaries which are described next in the section, form manifolds to bound a region so that the state trajectory can leave the region only through the exit boundary. In [21] sufficient conditions for a set of hypersurfaces to form a common flow region were established. Here, a Lyapunov approach is followed to efficiently compute hypersurfaces that form common flow regions for each control policy. Each common flow region is identified as a subset of an invariant manifold defined by a Lyapunov functional and is associated with a control policy. Since the switching function is generated by the underlying Petri net only sequences of invariant manifolds that correspond to control policies which satisfy the discrete specifications have to be examined.

Definition 3. *For the continuous part of the hybrid plant, the set B is a common flow region for a given region R if*

$$\forall x(t_0) \in B,\ \exists t_1, t_2,\ t_0 < t_1 < t_2$$

such that

$$x(t) \in B,\ t \leq t_1$$

and

$$x(t) \in R,\ t_1 < t < t_2$$

subject to

$$\dot{x}(t) = f_i(x(t))$$

In [21] two proposition are given which provide sufficient conditions for a set of hypersurfaces to form a common flow region. These hypersurfaces can be either *invariant* under the vector field of the given control policy or *cap boundaries* for the given vector field. Invariant hypersurfaces and cap boundaries form manifolds to bound a common flow region, so that the state trajectories cannot cross those manifolds.

Definition 4. *A set $M \subset X$ is said to be invariant with respect to the system $\dot{x} = f(x)$ if $x(t_0) \in M \Rightarrow x(t) \in M,\ \forall t \in \Re$.*

Consider the set of smooth hypersurfaces, $\{h_i,\ i \in I_B \subset 2^I\}$. The hypersurfaces is a set of smooth functionals $\{h_i :\ \Re^n \to \Re,\ i \in I_B\}$, defined on the state space of the plant. Each functional must satisfy the condition

$$\nabla_x h_i(\xi) \neq 0,\ \forall \xi \in \mathcal{N}(h_i) \tag{22}$$

which ensures that the null space of the functional $\mathcal{N}(h_i) = \{\xi \in \Re^n :\ h_i(\xi) = 0\}$ forms an $n-1$ dimensional manifold separating the state space.
Invariant hypersurfaces For a hypersurface h_i to be invariant under the vector field f of the given control policy, the following condition must be satisfied

$$\nabla_\xi h_i(\xi) \cdot f(\xi) = 0 \tag{23}$$

The set of all invariant hypersurfaces can be found in terms of $n-1$ functionally independent mappings which form the basis for the desired set of functionals, $\{h_i\}$. This basis is obtained by solving the characteristic equation

$$\frac{dx_1}{f_1(x)} = \frac{dx_2}{f_2(x)} = \ldots \frac{dx_n}{f_n(x)} \tag{24}$$

where $f_j(x)$ is the jth element of $f(x)$ ($f(x)$ is used rather than $f_i(x)$ to avoid subscript confusion).

Cap boundaries For a hypersurface h_c to form a cap boundary for a given vector field f and common flow region B, the following condition must be satisfied

$$\nabla_\xi h_c(\xi) \cdot f(\xi) < 0,\ \forall \xi \in B \cap \mathcal{N}(h_c) \tag{25}$$

Consider the hypersurface $h_c(x)$ that forms a cap boundary for the common flow region B. Assume that there exists an appropriate Lyapunov function $V(x)$ for the vector fields f such that

$$\begin{gathered} V(x) > 0,\ \forall x \in B \\ V(x) \to \infty \text{ as } \|x\| \to \infty \\ \dot{V}(x) < 0,\ \forall x \in B \end{gathered} \tag{26}$$

then $\Omega_c = \{x \in \Re^n | \ V(x) \leq c\}$ is bounded and the hypersurface $h_c(x) = V(x) - c$ is a cap boundary candidate. The constant parameter c can be selected appropriately so that the hypersurface $h_c(x)$ which bounds the common flow region B satisfies certain safety constraints.

As it was recognized in [21], the task of determining suitable invariant hypersurfaces is very difficult in general; for special cases (e.g. integrator systems), the differential equation (24) was solved analytically, otherwise an algorithm which was computationally inefficient was used. Here, we introduce a Lyapunov approach to determine cap boundaries. This approach is more efficient and can be applied to a larger class of systems; furthermore, the design based on Lyapunov functions exhibits desirable robustness properties. However, by assuming that the common flow regions are bounded by manifolds defined by Lyapunov functionals, we impose restrictive conditions on the dynamics of the continuous subsystems. In most of the cases, these conditions are quite restrictive but they suggest a systematic way to compute common flow regions. For example, we require stability (in the region of interest) for the continuous subsystems because then we can systematically approximate the region of attraction of an equilibrium. The next proposition gives sufficient conditions for the state to progress from one equilibrium to another.

Proposition 2. *Let $f_{i_1}, f_{i_2} \in \Sigma$ satisfy the following assumptions*

1. *Each f_i is globally Lipschitz and admits an isolated equilibrium point $\bar{x}_i$, and $\bar{x}_i$ is asymptotically stable w.r.t. f_i.*
2. *For each f_i there exists an appropriate Lyapunov function $V_i : \Re^n \to \Re$ and $\Omega_i = \{x \in \Re^n | \ V_i(x) \leq c_i\}$ such that*

$$\begin{aligned} V(x) > 0, \ \forall x \in \Omega_i \\ V(x) \to \infty \ as \ \|x\| \to \infty \\ \dot{V}(x) < 0, \ \forall x \in \Omega_i \end{aligned} \tag{27}$$

In addition, assume that $\Omega_{i_1} \cap \Omega_{i_2} \neq \emptyset$ and $\bar{x}_{i_1} \in R' = int(\Omega_{i_1} \cap \Omega_{i_2})$, then for every $x_0 \in \Omega_{i_1}$ there exists a switching sequence

$$s(x_0, t_0) = (i_1, \sigma^{t_{i_1}}(k_0)), (i_2, \sigma^{t_{i_2}}(k_1))$$

which drives the state to a region R of the equilibrium point $\bar{x}_{i_2}$.

Proof. Let $\Omega = \Omega_{i_1} \setminus \Omega_{i_2}$ and define the hypersurface $h_c(x) = \partial\Omega \cap \partial\Omega_{i_1}$ and $h_e(x) = \partial\Omega \cap \partial\Omega_{i_2}$. Since $\bar{x}_{i_1} \in R'$ is an asymptotically equilibrium point for f_{i_1}, Ω is a common flow region for $R' = int(\Omega_{i_1} \cap \Omega_{i_2})$. Let $\Omega' = \Omega_{i_2} \setminus R$ and define the hypersurface $h_c(x) = \partial\Omega_{i_2}$ and $h_e(x) = \partial R$, then Ω' is a common flow region for the target region R.

If we assume that the switching logic of the hybrid system is described by a DES (here a programmable timed Petri net), only sequences of vector fields that correspond to control policies that satisfy the discrete specifications have

to be considered. The control policies that satisfy the discrete specifications are exactly those that are accepted by the controlled Petri net which includes also the supervisor designed using the methodology based on place invariants (see above). They can be determined by identifying the periodic behavior of the Petri net. Several methods have been proposed to determine the periodic behavior of Petri nets (reachability tree, transition invariants, unfolding). For this problem, the size of the Petri net depends on the number of available symbolic inputs, hence we can efficiently identify all the periodic behaviors by computing the fundamental cycles of the reachability tree.

The underlying Petri net structure, which generates the switching policy offers two important advantages. First, it makes possible to efficiently design the supervisor that satisfy specifications that frequently appear in complex systems such as generalized mutual exclusion constraints. Second, it reduces considerably the search for common flow regions, since only desirable switching strategies generated by the controlled Petri net have to be examined.

The following corollary gives sufficient conditions for a switching sequence generated by the controlled Petri net to drive the continuous state x_0 to a target region of the state space. It is assumed that the initial conditions belong to the region of attraction Ω_{i_0} of the first control policy and that it progresses towards $\bar{x}_{i_m} \in \Omega_{i_m}$ by allowing switchings to occur on the intersection $\Omega_{i_j} \cap \Omega_{i_{j+1}}$ of consecutive invariant manifolds. In the case when all the pairs of control policies satisfy Proposition 2, the set $\Omega_{i_j} \cap \Omega_{i_{j+1}}$ will be nonempty and the proof is clear.

Corollary 1. *Suppose there exists a switching sequence with event projection $\pi_1(s) = i_0, i_1, \ldots, i_m$ accepted by the controlled Petri net such that every pair $(f_{i_j}, f_{i_{j+1}})$ satisfies Proposition 1 . Given a target region R such that $\bar{x}_{i_m} \in int(R)$, there exists routing policy $\nu^p(n)$ to drive the continuous state from any initial condition $x_0 \in \Omega_{i_0}$ to the region R in finite time. The firing time intervals $\sigma^t(n)$ will be chosen so that the switchings occur while $x \in int(\Omega_{i_j} \cap \Omega_{i_{j+1}})$.*

Remark The condition that every pair $(f_{i_j}, f_{i_{j+1}})$ satisfies Proposition 2 can be relaxed by allowing intermediate transitions which will keep the continuous state in the domain of attraction of $\bar{x}_{i_{j+1}}$.

The supervisor is implemented by assigning well-formed formulas to the places, transitions, input and output arcs of the controlled Petri net. Let $\{h_i\}$, $i = 1, \ldots, n$ be the set of hypersurfaces that bound a region M of the state space X. We can use the following well-formed formulas to describe that $x \in M$.

$$\ell = p_1 \wedge p_2 \wedge \ldots \wedge p_n \tag{28}$$

where p_i is a constraint of the form $h_i(x) < 0$. Consider a pair of vector fields $(f_{i_j}, f_{i_{j+1}})$ that satisfy Proposition 2 and let B_{i_j}, R_{i_j} and $B_{i_{j+1}}, R_{i_{j+1}}$ be the corresponding common flow and target regions. From Proposition 2 we have that the target region R_{i_j} coincides with the common flow region $B_{i_{j+1}}$. The switching algorithm is implemented by the following labeling functions, where $p \in P$ is the place to connect the output arc of t_{i_j} to the input arc of $t_{i_{j+1}}$

1. $\ell_P(p)$ is chosen to be a tautology.
2. $\ell_T(t_{i_j}), \ell_T(t_{i_{j+1}})$ are chosen to be the atomic rate formulas $\dot{x} = f_{i_j}(x)$ and $\dot{x} = f_{i_{j+1}}(x)$ respectively.
3. $\ell_O((t_{i_j}, p))$ is chosen to be a WFF of the form (28) representing that $x \in R_{i_j}$.
4. $\ell_I((p, t_{i_{j+1}}))$ is chosen to be a WFF of the form (28) representing that $x \in B_{i_{j+1}}$

Assuming that transition t_{i_j} is firing, the next transition to fire, $t_{i_{j+1}}$ is determined by the routing policy $\nu^p(n)$ so that the pair $(f_{i_j}, f_{i_{j+1}})$ satisfies Proposition 2. Transition $t_{i_{j+1}}$ will fire only when the firing time intervals $\sigma^t(n)$ assign true values to the logic formulas $\ell_O((t_{i_j}, p))$ and $\ell_I((p, t_{i_{j+1}}))$. For the initialization of the hybrid system we assume that $\ell_I((p, t_{i_0}))$ is a tautology.

Remark In the case when the Petri net is live and the event projection generated by the controlled Petri net $\pi_1(s)$ is an infinite sequence that satisfies Corollary 1, the hybrid system exhibits a periodic behavior in the sense that the continuous state is visiting periodically neighborhoods of the equilibria.

Affine Systems A class of systems that satisfy the conditions for supervisory control design of the previous section is the affine systems. They represent physical systems that are described by linear ordinary differential equations with one additional assumption. The input is allowed to take a finite number of prespecified constant values.

Consider the case the continuous dynamics are described by

$$\dot{x} = Ax + c_i, \quad T_k \leq t < T_{k+1}$$

where $c_i \in W \subset \Re^n$ a finite set of control vectors and the matrix $A \in \Re^{n \times n}$ is Hurwitz.

Let $f_i(x) = Ax + c_i$, then $\bar{x}_i = -A^{-1}c_i$ is a globally asymptotically stable equilibrium point for $\dot{x} = f_i(x)$. In view of the global asymptotic stability of each equilibrium point, it is clear that Proposition 2 holds for every pair of control inputs. The values for the control input can be selected so that the continuous state can be driven to prescribed regions of the state space.

Example: Hybrid System Describing Resource Contention Consider the case of two different processes that use the same resource to carry out their operations. This is a conflict situation which stems from the resource contention. More specifically, assume that each process consists of two different operations which are described by ordinary differential equations and the switching policy is represented by the Petri net in Fig. 1. This situation arises frequently in physical systems when different processes share the same resources. We will use this Petri net to describe the switching policy for two examples that follow. The first example is a temperature control system where the continuous dynamics are described by affine systems. In the second example, we consider a hybrid system with continuous dynamics described by first order integrators.

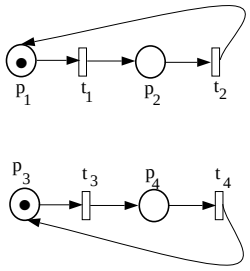

Fig. 1. Petri net describing the switching policy of the hybrid plant.

The incidence matrix of the plant net is

$$D_p = \begin{bmatrix} -1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix} \tag{29}$$

and the initial condition of the marking vector $\mu_{p_0} = [1, 0, 1, 0]^T$. We consider the mutual exclusion constraint $L\mu_p \leq b$, where $L = [0, 1, 0, 1]$ and $b = 1$. Using Proposition 1 the closed loop system has the incidence matrix

$$D = \begin{bmatrix} -1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 1 & -1 \\ -1 & 1 & -1 & 1 \end{bmatrix} \tag{30}$$

and initial condition $\mu_0 = [1, 0, 1, 0, 1]$. The last row of the incidence matrix D represents the Petri net supervisor.

The controlled Petri net is shown in Fig. 2 and describes the switching policy for the hybrid system which satisfies the mutual exclusion constraint.

Temperature Control System Let a temperature control system be described by the electrical circuit shown in figure 3. Here, an electrical analog of the temperature control system is used by considering the temperature being analogous to electric voltage, heat quantity to current, heat capacity to capacitance, and thermal resistance to electrical resistance. The control objective is to control the temperature at a point at the system by applying the heat input at a different point. The temperature control example is used in [34] to illustrate PID control design. Here, we assume that only discrete levels are available for the current (heat) input (u).

Let x_1 and x_2 denote the voltages across the capacitors C_1 and C_2 respectively. Suppose that the (voltages) temperatures x_1 and x_2 are to be controlled

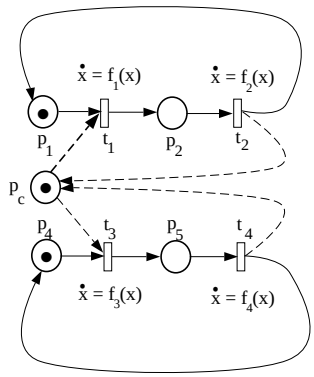

Fig. 2. The controlled Petri net of the resource contention example

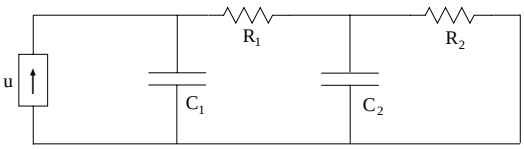

Fig. 3. The temperature control system

by changing the (current) heat input u, which is allowed to take finitely many discrete values. Consider that the numerical values of the electrical elements are

$$R_1 = R_2 = 1$$
$$C_1 = C_2 = 1$$

Then (using Kirchhoff's laws) the system is described by the state-space equation

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -1 & 1 \\ 1 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u \tag{31}$$

Assume that $q_i,\ i = 1, \ldots, N$ are the available discrete levels for the input that correspond to the available control policies. It is easily verified that the matrix A is Hurwitz and therefore $\bar{x}_i = -A^{-1}c_i$ where

$$A = \begin{bmatrix} -1 & 1 \\ 1 & -2 \end{bmatrix}, \quad c_i = \begin{bmatrix} q_i \\ 0 \end{bmatrix}$$

is a globally asymptotically stable equilibrium point for the system $\dot{x} = Ax + c_i$.

We assume that the discrete levels of the heat input are $q_1 = -10, q_2 = 10, q_3 = -20$, and $q_4 = 20$ and that the switching policy is described by the Petri net in Fig. 2 representing resource contention, where transition t_i corresponds to the control input c_i. In this case, the supervisor will determine the routing policy and the firing time intervals so that the continuous state of the hybrid

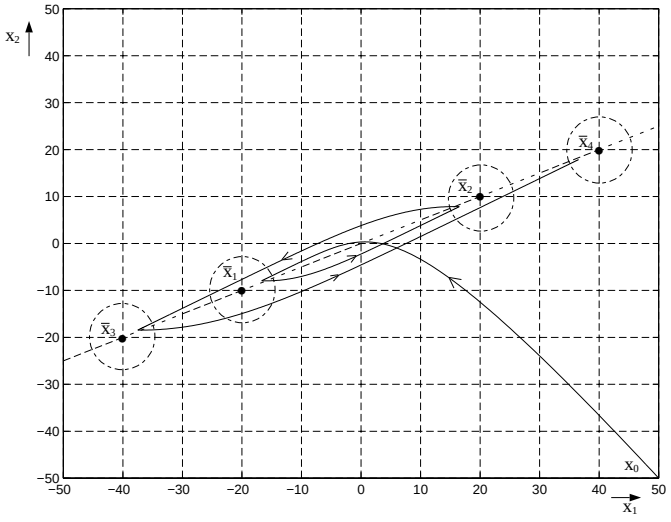

Fig. 4. Periodic behavior of the hybrid system

system visits periodically neighborhoods of the equilibria (Fig. 4). We consider a ball $B_i(\bar{x}_i, r)$ of radius r centered at $\bar{x}_i$, $i = 1, 2, 3, 4$ and we label the input and output arcs of the transitions with the WFFs $\ell_O((t_i, p)) = \ell_I((p, t_{i+1})) = x \in B_i(\bar{x}_i, r)$. Then the supervisor allows switchings to occur only when the above logical propositions are true.

5 Supervisory Control Design for First Order Integrators

Recently, attention has been focused on a particular class of hybrid systems in which the continuous dynamics are governed by the differential equation $\dot{x}(t) = c$, where $c \in \Re^n$ [35–38]. In [36] hybrid systems with continuous dynamics described by first order integrators are used for the control of batch processes. In the case when the continuous dynamics are described by first order integrators, the previous algorithm cannot be applied since the continuous part does not admit any isolated equilibria. In the following, we present an algorithm based on a similar idea as in the case of multiple equilibria. For each symbolic input there exist a family of invariant sets. The switching from a symbolic input to another occur at discrete time instants in a way that the system gradually progresses towards the target region. We determine a sequence of events that drive the state to the prescribed target region by solving a linear programming problem as in [38].

5.1 Hybrid Plant

The continuous state of the hybrid plant evolves in $X \in \Re^n$ and is described by first order integrators

$$\dot{x}(t) = c_i, \;\; T_k \leq \tau_i \leq T_{k+1}$$

where $c_i \in W \subset \Re^n$ a finite set of control vectors and T denotes the global time.

It is assumed as in the section 3 that the switching strategy is embedded in a programmable timed Petri net. Each transition is associated with a control policy. This assignment is defined by the labeling function $\ell_T(t) : T \to \Sigma$ which is chosen to be an atomic rate formula of the form $\dot{x} = c_i$.

Definition 5. *A set $\mathcal{C}$ is said to be a* finitely generated cone *if it has the form*

$$\mathcal{C} = \left\{ x : \; x = \sum_{j=1}^{r} \tau_j c_j, \; \tau_j \geq 0, \; c_j \in \Re^n, \; j = 1, \ldots, r \right\}$$

In the following, we assume that the finitely generated cone by the set W coincides with the continuous state space X. This assumption guarantees that continuous specifications such as state targeting can be satisfied everywhere in the state space X.

The system $\dot{x} = c$ admits a family of invariant sets described by the equation $x = ct + x_0$, which represents a family of parallel lines parameterized in t. The path from the initial state to the target region will be found by solving a linear programming problem to determine the time interval the state of the system evolves in each particular invariant set.

Supervisor The procedure for the design of the supervisor is similar to that of the previous section. First, applying DES control methods we construct a controlled Petri net which satisfies the discrete specifications. Assume now that the control objective is to drive the state to the target region $\mathcal{T} \subset X$. The task of the supervisor is to select at time T_k a control policy accepted by the Petri net, and to decide for how long it should be active. A systematic way to determine the temporal and routing data is first to identify all the periodic behaviors the controlled Petri net can generate and then assign temporal constraints to the switching times.

Proposition 3. *Consider the switching sequence*

$$s(t_0, x_0) = \left\{ (i, \sigma^{t_i}(k)) \right\}_{i=1,\ldots,N \; , \; k=0,1,\ldots,m}, \quad k_0 = t_0$$

such that

1. *The event projection $\pi_1(s) = i_0, i_1, i_2, \ldots, i_m$ consists of the transitions which form all the fundamental cycles.*

2. *The timed projection* $\pi_2(s) = \sigma^{t_{i_0}}(k_0), \sigma^{t_{i_1}}(k_1), \ldots, \sigma^{t_{i_m}}(k_m)$ *satisfies the following conditions* $\sum_k \sigma^{t_i}(k) = \tau_i$ *and* $0 < \Delta \leq \sigma^t(n) < \infty$ *for all transitions and firings;* $\tau = [\tau_1, \ldots, \tau_N]$ *is the solution of the following linear programming problem*

$$min\, a^T \tau$$
$$subject\, to: \begin{cases} x_f = x_0 + \sum_{i=1}^{N} \tau_i c_i \in \mathcal{T} \\ \tau_i \geq \Delta,\ i = 1, \ldots, N \end{cases}$$

where τ *is the vector of operation times to be determined,* a *is a weighting vector,* x_f *is the response of the continuous part at the time instant when the switching sequence* $s(t_0, x_0)$ *is completed.*

Then, the continuous state $x \in \Re$ *is driven to the target region in finite time* $t = \sum_{i=1}^{N} \tau_i$.

Proof. By integrating the state equation (5.1) all the reachable states x_f from initial state x_0 are given by

$$x_f = x_0 + \sum_{i=1}^{N} \tau_i c_i \tag{32}$$

where τ_i represents the total time the corresponding control policy c_i is active. Although there is no unique switching policy to satisfy the convex constrain $x_f \in \mathcal{T}$, the solution of the linear programming problem is unique and gives the control law that drives the state to the target region in minimum time.

Additionally, the necessary number of switches can be minimized by considering one firing of each transition. Additional safety conditions expressed as convex constraints can be incorporated in the linear programming problem. The supervisor solves the above linear programming problem and labels the places, transitions, input and output arcs of the controlled Petri net as follows

1. $\ell_P(p)$ is chosen to be a tautology.
2. $\ell_T(t_i)$ is chosen to be the atomic rate formulas $\dot{x} = c_i$.
3. $\ell_I((p, t_i))$ and $\ell_O((t_i, p))$ are chosen to be time constraints implement by a local timer so that $\sum_k \sigma^{t_i}(k) = \tau_i$.

Example: First Order Integrators We assume now that the continuous part of the hybrid plant consists of a set of first order integrators

$$\dot{x} = c_i \in \Re^2,\ \ i = 1, 2, 3, 4.$$
$$C = [c_1, c_2, c_3, c_4] = \begin{bmatrix} 0.5 & -1 & 1 & -1 \\ 1 & 1 & 1 & -0.4 \end{bmatrix}$$

and we associate the differential equation $\dot{x} = c_i$ with the transition t_i of the Petri net in Fig. 1. The control objective is to drive the state from the initial condition $x_0 = [1, -1]^T$ to the convex region $\mathcal{T}$ of the state space where

$$\mathcal{T} = \left\{ x \in \Re^2 :\ \begin{bmatrix} 1 \\ 1 \end{bmatrix} \leq x \leq \begin{bmatrix} 1.1 \\ 1.1 \end{bmatrix} \right\}$$

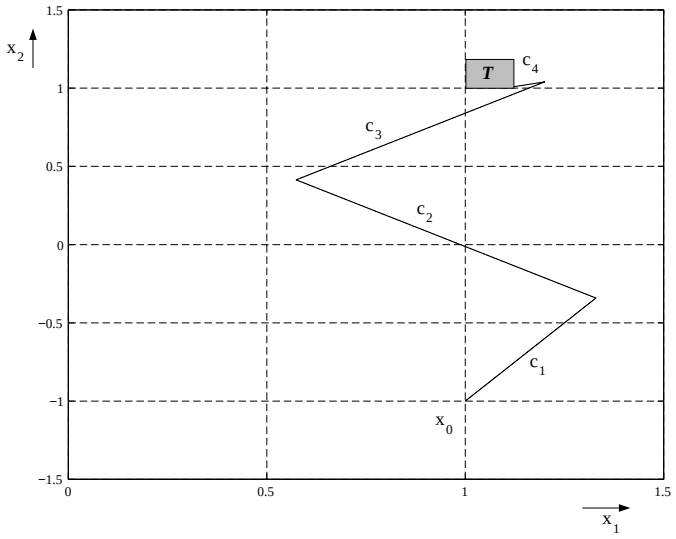

Fig. 5. The trajectory of the continuous state.

According to Proposition 3 we formulate the following linear programming problem

$$min\ (\tau_1 + \tau_2 + \tau_3 + \tau_4)$$

$$subject\ to: \begin{cases} x_f = x_0 + \tau_1 c_1 + \tau_2 c_2 + \tau_3 c_3 + \tau_4 c_4 \in \mathcal{T} \\ \tau_i \geq \Delta \end{cases}$$

where $\Delta = 0.1$. The solution of the linear programming problem gives

$$\tau_1 = 0.6585, \quad \tau_2 = 0.7554; \quad \tau_3 = 0.6261, \quad \tau_4 = 0.1$$

and we can drive the state from x_0 to $\mathcal{T}$ with one firing of each transition by setting $\sigma^{t_i} = \tau_i,\ i = 1, 2, 3, 4$. The trajectory of the continuous state is shown in Fig. 5.

6 Conclusions

In this paper, supervisory control of hybrid systems was addressed using a class of timed Petri nets named programmable timed Petri nets. New methodologies were introduced and algorithms were derived to address these issues. Sufficient conditions for supervisory control design were presented. For the case when the plant is a collection of affine systems or first order integrators with switching logic generated by a programmable timed Petri net, efficient algorithms for supervisory control synthesis were developed.

References

1. R.L. Grossman, A. Nerode, A.P. Ravn, and H. Rischel, editors. *Hybrid Systems*, volume 736 of *Lecture Notes in Computer Science*. Springer-Verlag, 1993.
2. P. Antsaklis, W. Kohn, A. Nerode, and S. Shastry, editors. *Hybrid Systems II*, volume 999 of *Lecture Notes in Computer Science*. Springer, 1995.
3. R. Alur, T.A. Henzinger, and E.D. Sontag, editors. *Hybrid Systems III, Verification and Control*, volume 1066 of *Lecture Notes in Computer Science*. Springer, 1996.
4. P. Antsaklis, W. Kohn, A. Nerode, and S. Shastry, editors. *Hybrid Systems IV*, volume 1273 of *Lecture Notes in Computer Science*. Springer, 1997.
5. A.Stephen Morse, editor. *Control using logic-based switching*, volume 222 of *Lecture Notes in Control and Information Sciences*. Springer, 1997.
6. P.J. Antsaklis and A. Nerode, editors. *IEEE Transactions on Automatic Control, Special Issue on Hybrid Systems*, 1998.
7. P.J. Antsaklis and X.D. Koutsoukos. On hybrid control of complex systems: A survey. In *3rd International Conference ADMP'98, Automation of Mixed Processes: Dynamic Hybrid Systems*, pages 1–8, Reims, France, March 1998.
8. P. Peleties and R. DeCarlo. Analysis of hybrid systems using symbolic dynamics and Petri nets. *Automatica*, 30(9):1421–1427, 1994.
9. P.J. Ramadge. On the periodicity of symbolic observations of piecewise smooth discrete-time systems. *IEEE Transactions on Automatic Control*, 35(7):807–813, 1990.
10. J. Lunze, B. Nixdorf, and H. Richter. Hybrid modelling of continuous-variable systems with application to supervisory control. In *Proceedings of the European Control Conference 97*, Brussels, Belgium, July 1997.
11. J. Le Bail, H. Alla, and R. David. Hybrid Petri nets. In *Proceedings 1st European Control Conference*, Grenoble, France, July 1991.
12. S. Pettersson and B. Lennartson. Hybrid modelling focused on hybrid Petri nets. In *Proceedings 2nd European Workshop on Real-time and Hybrid systems*, pages 303–309, Grenoble, France, June 1995.
13. A. Giua and E. Usai. High-level hybrid Petri nets: A definition. In *Proceeding of the 35th Conference on Decision and Control*, Kobe, Japan, December 1996.
14. D. Vibert, C. Valentin-Roubinet, and E. Neil. A modelling method to take into account fluctuations of continuous variables in a class of hybrid systems. In *Proceedings of the European Control Conference 97*, Brussels, Belgium, July 1997.
15. J-M. Flaus and H. Alla. Structural analysis of hybrid systems modelled by hybrid flow nets. In *Proceedings of the European Control Conference 97*, Brussels, Belgium, July 1997.
16. I. Demongodin and N.T. Koussoulas. Differential Petri nets: Representing continuous systems in a discrete-event world. *Transactions on Automatic Control*, 43(4):573–579, 1998.
17. M.D. Lemmon, K.X. He, and C.J. Bett. Modeling hybrid control systems using programmable Petri nets. In *3rd International Conference ADMP'98, Automation of Mixed Processes: Dynamic Hybrid Systems*, pages 177–184, Reims, France, March 1998.
18. J.O. Moody. *Petri Net Supervisors for Discrete Event Systems*. PhD thesis, Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN, 1997.
19. N.H. McClamroch, C. Rui, I. Kolmanovsky, and M. Reyhanoglu. Hybrid closed loop systems: A nonlinear control perspective. In *Proceedings of the 36th IEEE Conference on Decision and Control*, pages 114–119, 1997.

20. John Guckenheimer. A robust hybrid stabilization strategy for equilibria. *IEEE Transactions on Automatic Control*, 40(2):321–326, 1995.
21. J.A. Stiver, P.J. Antsaklis, and M.D. Lemmon. Interface and controller design for hybrid control systems. In Panos Antsaklis, Wolf Kohn, Anil Nerode, and Shankar Sastry, editors, *Hybrid Systems II*, volume 999 of *Lecture Notes in Computer Science*, pages 462–492. Springer, 1995.
22. J.A. Stiver, P.J. Antsaklis, and M.D. Lemmon. An invariant based approach to the design of hybrid control systems. In *IFAC 13th Triennial World Congress*, volume J, pages 467–472, San Francisco, CA, 1996.
23. X.D. Koutsoukos and P.J. Antsaklis. An approach to hybrid systems control applied to clocks. In *Fifth International Hybrid Systems Workshop*, Notre Dame, USA, September 1997.
24. X.D. Koutsoukos, K.X. He, M.D. Lemmon, and P.J. Antsaklis. Timed petri nets in hybrid systems: Stability and supervisory control. *DEDS: Theory and Applications*, 8(2), 1998.
25. R. Alur and D. Dill. The theory of timed automata. *Theoretical and Computer Science*, 126:183–235, 1994.
26. R. Alur, T. Henzinger, and P-H Ho. Automatic symbolic verification of embedded systems. *IEEE Transactions on Software Engineering*, 22:181–201, 1996.
27. J.L. Peterson. *Petri Net Theory and the Modeling of Systems.* Prentice-Hall, Englewood Cliffs, NJ, 1981.
28. W. Reisig. *Petri Nets.* Springer-Verlag, 1985.
29. T. Murata. Petri nets: Properties, analysis and applications. *Proceedings of IEEE*, 77(4):541–580, 1989.
30. F. Baccelli, G. Cohen, and B. Gaujal. Recursive equations and basic properties of timed Petri nets. *DEDS: Theory and Applications*, 1(4):415–439, 1992.
31. K. Yamalidou and J.C. Kantor. Modeling and optimal control of discrete-event chemical processes using Petri nets. *Computers in Chemical Engineering*, 15(7):503–519, 1991.
32. J.O. Moody and P.J. Antsaklis. Supervisory control using computationally efficient linear techniques: A tutorial introduction. In *Proceedings of the 5th International IEEE Mediterranean Conference on Control and Systems*, Paphos, Cyprus, July 1997.
33. J.O. Moody and P.J. Antsaklis. Supervisory control of Petri nets with uncontrollable/unobservable transitions. In *Proceedings of the 35th Conference on Decision and Control*, pages 4433–4438, Kobe, Japan, December 1996.
34. Bernard Friedland. *Advanced Control System Design.* Prentice-Hall, 1996.
35. R. Alur, C. Courcoubetis, T.A. Henzinger, and P-H. Ho. Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems. In Robert L. Grossman, Anil Nerode, Anders P. Ravn, and Hans Rischel, editors, *Hybrid Systems*, volume 736 of *Lecture Notes in Computer Science*, pages 209–229. Springer-Verlag, 1993.
36. Michael Tittus. *Control Synthesis for Batch Processes.* PhD thesis, Control Engineering Lab., Chalmers University of Technology, Göteborg, Sweden, 1995.
37. J.A. Stiver, P.J. Antsaklis, and M.D. Lemmon. An invariant based approach to the design of hybrid control systems containing clocks. In Rajeev Alur, Thomas A. Henzinger, and Eduardo D. Sontag, editors, *Hybrid Systems III, Verification and Control*, volume 1066 of *Lecture Notes in Computer Science*, pages 464–474. Springer, 1996.
38. B. Lennartson, M. Tittus, B. Egardt, and S. Petterson. Hybrid systems in process control. *Control Systems Magazine*, 16(5):45–56, October 1996.