Abstractions of Constrained Linear Systems

Herbert G. Tanner and George J. Pappas

Department of Electrical and Systems Engineering
University of Pennsylvania
Philadelphia, PA 19104-6228

***Abstract*—Simulation relations are powerful abstraction techniques in computer science that reduce the complexity of analysis and design of labeled transition systems. In this paper, we define and characterize simulation relations for discrete-time linear systems in the presence of state and input constraints. Given a discrete-time linear system and the associated constraints, we consider a control-abstract embedding into a transition system. We then establish necessary and sufficient conditions for one constrained linear system to simulate the transitions of the other. Checking the simulation conditions is formulated as a linear programming problem which can be efficiently solved for systems of large dimensions. We provide an example where our approach is applied to the hybrid model of the Electronic Throttle Control (ETC) System.**

I. Introduction

Theoretical computer science, and, in particular, the areas of concurrency theory [12], and computer aided verification [11] have established formal notions of abstraction and model refinement which exploit the hierarchical and compositional nature of large scale systems. In the context of hybrid systems, such notions have been recently considered by [10],[2], and [7]. In the control community, similar ideas have been considered in the hierarchical, supervisory control of discrete event systems [4], [21], and hybrid systems (see surveys [1], [8]).

Simulation relations of labeled transition systems provide such a formal notion of abstraction [12]. Roughly, transition system T_2 simulates transition system T_1, if every transition taken by T_1 can be matched by a similar transition taken by T_2. Simulation relations are used in order to establish modeling consistency between various levels of hierarchical systems, as transitions of the higher level system T_1 can be matched by the lower level system T_2.

As mentioned in [20], simulation relations have escaped the world of purely continuous systems. More recently, a notion of simulation was introduced for continuous-time systems [14]. Given a continuous system and quotient map, a formal construction was provided for extracting quotient systems that simulated the trajectories of the original system. Furthermore, linear maps that preserve control theoretic properties such as controllability [14], and stabilizability [13] were characterized. Similar results have also been established for nonlinear systems [15]. Simulation relations for unconstrained discrete-time linear systems have been established in [18].

In this paper we derive necessary and sufficient conditions for simulation relations between discrete-time linear systems that are subject to state and input constraints. We first embed constrained linear systems into transition systems. Control input information is abstracted away, contrary to model reduction methods in which control inputs are preserved [3]. The simulation relations considered in this paper can capture at least two important cases: complexity reduction and refinement. In the former case, one is concerned with reducing the dimensionality of the system to facilitate analysis. In the latter case, one may be interested in either refining a controller designed at a higher level or substituting the target system with a more complicated. The simulation conditions are expressed as a set-inclusion relationship that can be checked numerically using a linear programming formulation. The structure of the linear programming formulation, naturally reflect the game theoretic interpretation of simulation relations, a subject that has a long and rich history in theoretical computer science.

The outline of this paper is as follows: In Section II we review the definition of simulation relations for transition systems. In Section III we derive necessary and sufficient conditions for simulation relations between constrained, discrete-time, linear systems. Section IV provides a computational framework for checking the simulation conditions and Section V illustrates the application of our approach on a challenge problem, the ETC problem. The conclusions from this work are summarized in Section VI.

II. Simulations of Transition Systems

In this section we review the standard definitions of simulation relations for transition systems [12]. A (labeled) transition system is defined as follows:

Definition II.1 *A labeled transition system is a tuple $T = (Q, \Sigma, \longrightarrow)$ that consists of:*

- *A (possibly infinite) set Q of states,*
- *A (possibly infinite) set Σ of labels,*
- *A transition relation $\longrightarrow \subseteq Q \times \Sigma \times Q$,*

The transition $(q_1, \sigma, q_2) \in \longrightarrow$ is commonly denoted as $q_1 \xrightarrow{\sigma} q_2$. The transition system is called *finite* if Q and Σ are finite, and *infinite* otherwise. A *region* is a subset

$P \subseteq Q$ of the states. The σ-*successor* of a region P is defined as the set that can be reached from P with one σ-transition. More precisely,

$$\text{Post}_\sigma(P) = \{q \in Q \mid \exists p \in P \text{ with } p \xrightarrow{\sigma} q\} \quad (1)$$

Simulation relations between transition systems formally define when one transition system implements another.

Definition II.2 *Let $T_1 = (Q_1, \Sigma, \longrightarrow_1)$ and $T_2 = (Q_2, \Sigma, \longrightarrow_2)$ be two transition systems over the same label set Σ . The relation $S \subseteq Q_1 \times Q_2$ is called a simulation relation if for all $(q_1, q_2) \in S$, the following property holds: if $q_1 \xrightarrow{\sigma} q_1'$, then there exists $q_2' \in Q_2$ with $q_2 \xrightarrow{\sigma} q_2'$ and $(q_1', q_2') \in S$.*

If such a simulation relation exists, then T_2 simulates (or implements) T_1, since every σ-transition taken by T_1 can be matched (or implemented) by a σ-transition of T_2. The label set Σ is common to both transition systems. In general T_2 may have many more transitions, and may be a much more complicated system. Transition system T_1 can also serve as a more abstract description of transition system T_2. If, in addition, T_1 also simulates T_2 with the same relation S, then T_1 and T_2 are called *bisimilar*.

The language of a transition system, denoted $L(T)$, is the collection of label sequences that can be generated by transition system T. It is straightforward to show that if transition system T_2 simulates T_1, then $L(T_1) \subseteq L(T_2)$. Therefore, the behavior of T_1 is contained in that of T_2. Simulation relations, even though sufficient for language inclusion, are preferable to language inclusion since there are much easier to check algorithmically.

III. Simulations of Constrained Linear Systems

We begin by embedding linear systems into a transition system choosing one possible embedding out of a variety of different ones: a transition can occur whenever an admissible control exists, where by admissible control we mean an input that ensures that transitions do not violate the state constraints. Consider discrete-time, constrained linear control systems:

$$\Delta: \qquad x_{k+1} = Ax_k + Bu_k \quad (2)$$

with time $k \in \mathbb{N}_+$, state x_k belonging in a set $X \subseteq \mathbb{R}^n$, control u_k belonging in a set $U \subseteq \mathbb{R}^m$, and matrices A, B of appropriate dimension. From linear systems theory [22], we know that given an initial condition x_0 at time zero, and an input sequence $\{u_i\}_{i=0}^{k-1} = \{u_0, u_1, \ldots, u_{k-1}\}$, then the state x_k at time k is

$$x_k = A^k x_0 + \sum_{i=0}^{k-1} A^{k-i-1} B u_i \quad (3)$$

The embedding of discrete-time systems into transition systems preserves information about the state in which the system is at each single time step, abstracting away the particular control that was used the transition.

Definition III.1 *The transition system $T_\Delta = (Q, \Sigma, \longrightarrow)$ generated by Δ consists of:*

- *State space $Q = X \subseteq \mathbb{R}^n$,*
- *Unique label $\Sigma = \{1\}$,*
- *Transition relation $\longrightarrow \subseteq Q \times \{1\} \times Q$ with*

$$x \xrightarrow{1} x' \Leftrightarrow \exists u \in U : x' = Ax + Bu \wedge Ax + Bu \in X$$

The transitions of the transition system naturally correspond to evolution of the discrete-time system in one time step. Furthermore, the transitions of Definition III.1 are *control abstract* in the sense that the transition system does not care which u is responsible for the transition of the discrete-time system, as long as the states stays in X.

Consider two discrete-time, state and input constrained linear systems:

$$\Delta_1 : x_{k+1} = Ax_k + Bu_k, \; x \in X \subseteq \mathbb{R}^n, u \in U \subseteq \mathbb{R}^m \quad (4)$$

$$\Delta_2 : z_{k+1} = Fz_k + Gv_k, \; z \in Z \subseteq \mathbb{R}^r, v \in V \subseteq \mathbb{R}^s \quad (5)$$

where matrices A,B,F, and G are of appropriate dimension. Linear systems Δ_1 and Δ_2 generate various transition systems T_{Δ_1} and T_{Δ_2} respectively.

The simulation relations we shall consider in this paper are of the form $S \subseteq Q_1 \times Q_2$, with $Q_1 = X \subseteq \mathbb{R}^n$ and $Q_2 = Z \subseteq \mathbb{R}^r$ where

$$(x, z) \in S \subseteq Q_1 \times Q_2 \Leftrightarrow z = Hx + y, \quad y \in Y \quad (6)$$

where $H \in \mathbb{R}^{r \times n}$ is an arbitrary linear map, and $Y \subseteq \mathbb{R}^r$ is a set. Relation S can be thought of as a set valued map assigning to each $x \in Q_1$ an affine set $Hx + Y \subseteq Q_2$.

The structure of the relations (6) considered in this paper captures at least two important cases. In the first case, where $Y = 0$ and the map Hx is surjective, we are interested in simulating the transitions of Δ_1 by a system Δ_2, which should be smaller in size, thus performing complexity reduction. Such a case can be useful in model checking and verification. In the second case, where the map Hx is injective and $Y = \mathcal{R}(H)^\perp$ (the orthogonal complement of the range of H) we are interested in the more complicated system Δ_2 simulating the transitions of the simpler system Δ_1, thus refining the transitions from the simpler to the more complicated model.

Theorem III.2 (Simulation) *Consider discrete time systems Δ_1 and Δ_2 given by (4)-(5), and a relation S of the form (6). Then T_{Δ_2} simulates T_{Δ_1} if and only if*

$$(HA - FH)X + HBU - FY \subseteq GV - Y \quad (7a)$$

$$AX + BU \subseteq X \quad (7b)$$

$$FZ + GV \subseteq Z \quad (7c)$$

Proof: By Definition II.2 and equation (1), with σ being a one-step transition, Δ_2 simulates Δ_1 with respect to the relation S *if and only if* for all $(x, z) \in S$ it holds that: $\forall x' \in \mathrm{Post}_1(x), \exists z' \in \mathrm{Post}_1(z) : (x', z') \in S$. Given (6), the above is rewritten as: $\forall (x, z) \in S, \forall x' \in \mathrm{Post}_1(x), \exists y_1 \in Y : z' = Hx' + y_1 \in \mathrm{Post}_1(z)$. Definition III.1 provides explicit expressions for the Post_1 operators T_{Δ_1} and T_{Δ_2}. Substituting, the necessary and sufficient condition for simulation becomes:

$$\begin{aligned} &\forall\ (x, z) \in S, \forall u \in U : Ax + Bu \in X, \exists y_1 \in Y, \\ &\qquad \exists v \in V : H(Ax + Bu) + y_1 = Fz + Gv \in Z \end{aligned}$$

Since $(x, z) \in S$, z can always be expressed as $z = Hx+y_2$ with $y_2 \in Y$, which makes the above equivalent to:

$$\begin{aligned} &\forall x \in X, \forall y_2 \in Y, \forall u \in U : Ax + Bu \in X, \\ &\exists y_1 \in Y, \exists v \in V : H(Ax+Bu)+y_1 = F(Hx+y_2)+Gv \in Z \end{aligned}$$

Collecting terms, and eliminating the quantifiers we have:

$$(HA - FH)X + HBU - FY \subseteq GV - Y.$$

Thus, the necessary and sufficient condition for simulation can take the form of (7a). The remaining conditions:

$$AX + BU \subseteq X, \qquad FZ + GV \subseteq Z$$

restrict transitions that do not lead to admissible states. ■

IV. Simulation Checking Algorithm

An important question that arises is how to check the simulation conditions of Theorem III.2. We show that when the constrained sets can be expressed as polyhedra, checking the conditions for simulation is equivalent to solving a number of Linear Programming (LP) problems.

A. The Linear Programming Formulation

Consider the linear systems (4) and (5) and assume that the sets X, U, Z, V and Y are given as:

$$\begin{aligned} X &= \{x \in \mathbb{R}^n \mid C_x x \preceq d_x\}, \quad U = \{u \in \mathbb{R}^m \mid C_u u \preceq d_u\}, \\ Z &= \{z \in \mathbb{R}^r \mid C_z z \preceq d_z\}, \quad V = \{v \in \mathbb{R}^s \mid C_v v \preceq d_v\}, \\ Y &= \{y \in \mathbb{R}^r \mid C_y y \preceq d_y\}. \end{aligned}$$

The above constraint sets can be grouped together into two polyhedral regions, each characterizing each side of the simulation condition (7a):

$$\mathcal{P}_l \triangleq \{q = (x, u, y)^T \mid P_l q \preceq d_l\} \tag{8a}$$

$$\mathcal{P}_r \triangleq \{w = (y, v)^T \mid P_r w \preceq d_r\} \tag{8b}$$

where:

$$\begin{aligned} P_l &\triangleq \mathrm{diag}\{C_x, C_u, C_y\}, \qquad d_l \triangleq (d_x, d_u, d_y)^T, \\ P_r &\triangleq \mathrm{diag}\{C_y, C_v\}, \qquad d_r \triangleq (d_y, d_v)^T. \end{aligned}$$

In order for transitions to remain within X and Z, conditions (7b,c) are expressed as:

$$C_x Ax + C_x Bu \preceq d_x, \quad C_z Fy + C_z Gv \preceq d_z - C_z FHx.$$

By defining $C_1 \triangleq [C_x A\ C_x B\ 0]$, $C_2 \triangleq [C_z F\ C_z G]$ and $C_3 \triangleq [C_z FH\ 0\ 0]$, the above can be rewritten as:

$$C_1 q \preceq d_x, \qquad C_2 w \preceq d_z - C_3 q. \tag{9}$$

Now define the linear maps:

$$\begin{aligned} &M_l : \mathcal{P}_l \to \mathcal{P}_1;\ q \mapsto \begin{bmatrix} HA - FH & HB & -F \end{bmatrix} q \\ &M_r : \mathcal{P}_r \to \mathcal{P}_2;\ w \mapsto \begin{bmatrix} -I & G \end{bmatrix} w \end{aligned}$$

Clearly, the image of a polyhedron under a linear map is itself a polyhedron. The simulation condition (7a) then requires the inclusion $\mathcal{P}_1 \subseteq \mathcal{P}_2$. The following theorem offers a computational means of checking this inclusion:

Theorem IV.1 *The necessary and sufficient conditions conditions for simulation, (7), are satisfied iff each of the following LP problems is feasible:*

$$\begin{aligned} \min_s \quad & p_r^k (I - M_r^+ M_r) s \\ \text{s.t.} \quad & P_r (I - M_r^+ M_r) s \preceq d_r - P_r M_r^+ M_l q_k^* \\ & C_2 (I - M_r^+ M_r) s \preceq d_z - (C_3 + C_2 M_r^+ M_l) q_k^*. \end{aligned}$$

where p_r^k is the k^{th} row of P_r, M_r^+ is the pseudoinverse of M_r and $q_k^* = (x^*, u^*, y^*)_k^T$ is the solution of

$$\begin{aligned} \max_q \quad & p_r^k M_r^+ M_l q, \\ \text{s.t.} \quad & P_l q \preceq d_l \qquad C_1 q \preceq d_x. \end{aligned}$$

Proof: If $\mathcal{P}_1$ and $\mathcal{P}_2$ are given as:

$$\mathcal{P}_1 = \{t \mid P_1 t \preceq d_1\} \qquad \mathcal{P}_2 = \{t \mid P_2 t \preceq d_2\}$$

then the checking condition $\mathcal{P}_1 \subseteq \mathcal{P}_2$ is equivalent to verifying that $p_2^j t^* \leq d_2^j$ with j ranging over the number of rows of P_2, where t^* is the solution of the LP problem:

$$\max_t \quad p_2^j t, \qquad \text{s.t.} \quad P_1 t \preceq d_1. \tag{10}$$

The explicit description of $\mathcal{P}_1$ and $\mathcal{P}_2$ requires vertex representation of $\mathcal{P}_l$ and $\mathcal{P}_r$, which is generally difficult. Thus, a problem formulation in the original space where $\mathcal{P}_l$ and $\mathcal{P}_r$ are expressed in edge representation (8a) is preferable. Since M_r is a linear surjective map, the solutions of (10) are a subset of the solutions of:

$$\max_q \quad p_r^j M_r^+ M_l q \tag{11a}$$

$$\text{s.t.} \quad P_l q \preceq d_l, \qquad C_1 q \preceq d_x. \tag{11b}$$

Proceedings of the American Control Conference
Denver, Colorado June 4-6, 2003

where j ranges over the number of rows of P_r.

Let z_j^* be the solution of (11). Then, z_j^* is the point in $\mathcal{P}_l$, with an image under $M_r^{-1}M_l$, (M^{-1} denoting the inverse mapping), which is the "worst" among all points on hyperplane $p_r^j w = c$, with respect to containment in $\mathcal{P}_2$. For that point to be contained in $\mathcal{P}_2$, the LP problem:

$$\begin{aligned} \min_s \quad & p_r^j\big(M_r^+ M_l z_j^* + (I - M_r^+ M_r)s\big) \\ \text{s.t.} \quad & P_r\big(M_r^+ M_l z_j^* + (I - M_r^+ M_r)s\big) \preceq d_r \\ & C_2\big(M_r^+ M_l z_j^* + (I - M_r^+ M_r)s\big) \preceq d_z - C_3 z_j^* \end{aligned}$$

should have a feasible solution. And since optimization is only with respect to s, the above simplifies to:

$$\min_s \quad p_r^j (I - M_r^+ M_r)s \tag{12a}$$

$$\text{s.t.} \quad P_r(I - M_r^+ M_r)s \preceq d_l - P_r M_r^+ M_l z_j^*, \tag{12b}$$

$$C_2(I - M_r^+ M_r)s \preceq d_z - (C_3 + C_2 M_r^+ M_l) z_j^* \tag{12c}$$

■

Theorem IV.1 reveals the game-theoretic interpretation of simulation condition (7a), where system Δ_1 first picks the worst transition by maximizing (x^*, u^*, y^*), which must then be matched by Δ_2 by choosing v^*. Figures 1-2 provide a pictorial description of the procedure followed in the proof of Theorem IV.1.

The number of LP problems that need to be solved is at most $2n_r$ where n_r is the number of faces describing $\mathcal{P}_r$. In other words, the complexity of checking (7) is proportional to the complexity of the polyhedra describing the admissible regions for state and input.

V. A Challenge Problem

This approach was applied to an instance of the Electronic Throttle Control (ETC) problem: a throttle controls the amount of air-fuel mixture that is sent to the engine of a car. The throttle is electronically controlled by a PWM driven motor. In the main mode of operation of the system, the PWM signal is produced based on the output of a sliding mode controller, which takes as input the accelerator pedal position after being filtered by a fifth order linear filter. In the closed loop system, the throttle is tracking the reference signal produced by the driver. The ETC is modeled as a hybrid system with six different modes, distinguishing between the cases where the motor is receiving an input pulse or not and in which direction the throttle is moving. In each mode the states consists of nine continuous variables expressing the current and voltage of the motor, the angle and rotational velocity of the throttle, and the five states of a filter.

Such a system should meet certain specifications, some of which can be formalized in terms of overshoot, rise time and steady error for the throttle angle. However, verifying these properties on the original system is too computationally expensive due to the relatively high dimension

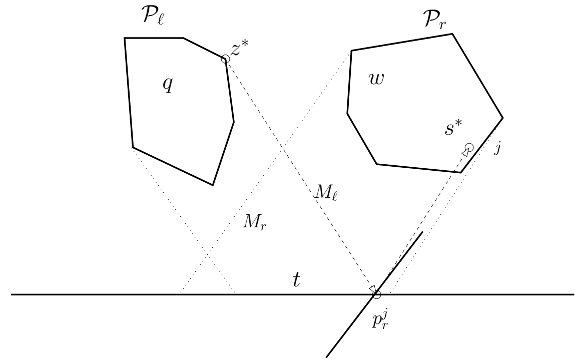

Fig. 1. The image of $\mathcal{P}_\ell$ is contained in $\mathcal{P}_2$. The abstract system can simulate the original.

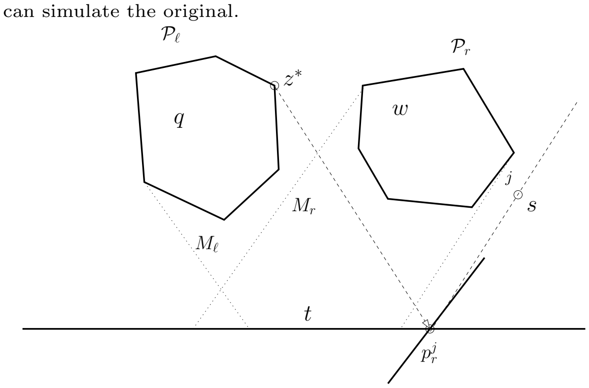

Fig. 2. The image of $\mathcal{P}_\ell$ is not contained in $\mathcal{P}_2$. The abstract system cannot simulate the original.

of the continuous state vector which inhibits reachability computations. Thus, the system dimension in each mode is reduced using the proposed methodology and verification can proceed using a lower dimensional system (Figure 4). If the property is verified on the abstract system, then it will also hold for the original system, since by the definition of simulation, the abstract system includes all the behaviors of the original.

The dynamics of the original system in each mode, is described by:

$$x[k+1] = A_i x[k] + B_i u[k] \tag{13a}$$

$$C_x^i x \preceq d_x^i,\ C_u^i u \preceq d_u^i,\ i = 1, \ldots, 6 \tag{13b}$$

where $A_i,\ i = 1, \ldots, 6$ are 9×9 matrices and $B_j,\ j = 1, \ldots, 6$ are 9×3 matrices. Due to lack of space, only the numerical expressions for A_1 and B_1 are given:

$$B_1 = \begin{bmatrix} 2.58\cdot10^{-1} & 2.07\cdot10^{-5} & 0 \\ 2.86\cdot10^{-1} & -3.67\cdot10^{-4} & 0 \\ 1.16\cdot10^{-5} & 7.00\cdot10^{-5} & 0 \\ 3.33\cdot10^{-2} & 1.40\cdot10^{-1} & 0 \\ 0 & 0 & -8.13\cdot10^{-9} \\ 0 & 0 & -1.44\cdot10^{-9} \\ 0 & 0 & -3.77\cdot10^{-10} \\ 0 & 0 & -7.84\cdot10^{-11} \\ 0 & 0 & 1.00\cdot10^{-3} \end{bmatrix}$$

Proceedings of the American Control Conference
Denver, Colorado June 4-6, 2003

$$A_1 = \begin{bmatrix} 7.09\cdot10^{-1} & -7.36\cdot10^{-2} & -5.76\cdot10^{-5} & 3.88\cdot10^{-4} & 0 & 0 & 0 & 0 & 0 \\ 7.30\cdot10^{-2} & 2.46\cdot10^{-2} & 1.02\cdot10^{-3} & -4.28\cdot10^{-3} & 0 & 0 & 0 & 0 & 0 \\ 1.07\cdot10^{-4} & -4.43\cdot10^{-6} & 1.00 & 1.00\cdot10^{-4} & 0 & 0 & 0 & 0 & 0 \\ 2.02\cdot10^{-1} & -1.16\cdot10^{-2} & -3.90\cdot10^{-1} & 1.00 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 6.14\cdot10^{-1} & -1.26\cdot10^{-1} & -1.06\cdot10^{-2} & -4.45\cdot10^{-4} & -1.50\cdot10^{-5} \\ 0 & 0 & 0 & 0 & 4.09\cdot10^{-1} & 9.66\cdot10^{-1} & -2.90\cdot10^{-3} & -1.23\cdot10^{-4} & -4.16\cdot10^{-6} \\ 0 & 0 & 0 & 0 & 2.26\cdot10^{-1} & 1.01 & 9.99\cdot10^{-1} & -4.34\cdot10^{-5} & -1.47\cdot10^{-6} \\ 0 & 0 & 0 & 0 & 8.01\cdot10^{-2} & 5.21\cdot10^{-1} & 1.02 & 1.00 & -3.86\cdot10^{-7} \\ 0 & 0 & 0 & 0 & 1.05\cdot10^{-2} & 8.91\cdot10^{-2} & 2.62\cdot10^{-1} & 5.12\cdot10^{-1} & 1.00 \end{bmatrix},$$

As it can be seen in the following Tables, the original state bounds (especially for states $x_5, \ldots, x_9$) are quite conservative. This is due to the absence of any particular physical constraint for this part of the state vector. The conservative nature of the original state bounds will eventually be reflected upon the control authority that is necessary in the abstracted system. This implies that constraints are actually useful in abstraction: the use of constraint information can lead to more specific system description and less conservative abstractions.

Concrete State and Input Constraints						
	x_1	x_2	x_3	x_4	x_5	x_6
max	7.4	27.3	1.82	19.7	1.57	$1.57\cdot10^8$
min	0	0	0.25	0	0	$-1.57\cdot10^8$
	u_1	u_2	u_3	x_7	x_8	x_9
max	12	-1	1.57	$1.57\cdot10^8$	$1.57\cdot10^8$	$1.57\cdot10^8$
min		12	-1	0 $-1.57\cdot10^8$	$-1.57\cdot10^8$	$-1.57\cdot10^8$

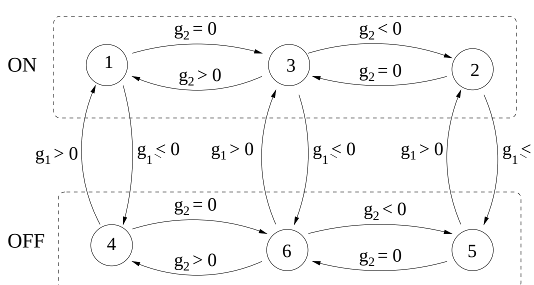

Fig. 3. The hybrid system modeling the original ETC System.

The switching conditions are expressed by the guards of the hybrid system. These are algebraic expressions involving the continuous states (Figure 3):

$$\begin{aligned} g_1 =& -x_1 + 1.64x_3 - 8.36\cdot10^{-2}x_4 + 0.59\text{sign}(x_4) \\ &+ 332x_7 + 6.49x_8 \\ &\quad - 2\text{sign}(20x_3 + x_4 - 77.2x_8 - 3.02x_9 - 5) \\ g_2 =& x_4 \end{aligned}$$

The specifications that the ETC system should meet concern the steady state error of the throttle angle, x_3 as well as the rise time and overshoot. For a hybrid system with continuous dynamics of that size, reachability computation is beyond the limits of state-of-the art computational tools [17], [6], [9], [16], [19], [5].

The abstraction map is designed to preserve the information that is crucial for verification (x_3 state), as well as for the discrete transitions between the modes (g_1, g_2 guards), while compressing the state as much as possible. This is done by aggregating the states that appear in the guards into abstract states in a way that all transitions can still be detected:

$$H = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 20 & 1 & 0 & 0 & 0 & -77.2476 & -3.0175 \\ -1 & 0 & 1.6387 & -0.0836 & 0 & 0 & 332.3595 & 6.4914 & 0 \end{bmatrix}$$

The abstracted dynamics in each mode is obtained according to [14]:

$$z[k+1] = F_i z[k] + G_i w[k] \tag{14a}$$

$$C_z^i z \preceq d_z^i,\ C_w^i w \preceq d_w^i,\ i = 1, \ldots, 6 \tag{14b}$$

where $F_i = HA_iH^+$ and $G_i = \begin{bmatrix} HB_i & HA_i\text{Ker}(H) \end{bmatrix}$, and matrices G_i being replaced by the minimum set of column vectors that span the range of each G_i. This procedure yields the following abstracted dynamics for mode 1:

$$F_1 = \begin{bmatrix} 1.00 & 10^{-3} & -8.10\cdot10^{-11} & -9.65\cdot10^{-10} \\ -3.90\cdot10^{-1} & 1.00 & -1.53\cdot10^{-7} & -1.82\cdot10^{-6} \\ 4.05\cdot10^{-3} & -1.50\cdot10^{-4} & 1.00 & -2.40\cdot10^{-1} \\ -3.43\cdot10^{-2} & 1.08\cdot10^{-3} & 1.78\cdot10^{-3} & 1.00 \end{bmatrix}$$

$$G_1 = \begin{bmatrix} -2.13\cdot10^{-10} & -1.30\cdot10^{-6} & 5.16\cdot10^{-4} & 1.00 \\ -3.74\cdot10^{-7} & -2.61\cdot10^{-3} & 1.00 & 5.16\cdot10^{-4} \\ 1.18\cdot10^{-1} & -9.93\cdot10^{-1} & -2.59\cdot10^{-3} & -4.49\cdot10^{-8} \\ -9.93\cdot10^{-1} & -1.17\cdot10^{-1} & -3.05\cdot10^{-4} & -5.26\cdot10^{-9} \end{bmatrix}$$

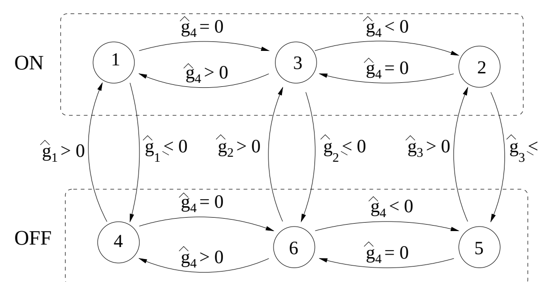

Fig. 4. The abstract hybrid system for ETC.

The guards for the abstract ETC system now take the form (Figure 4):

$$\hat{g}_1 = z_4 + 0.59 - 2\text{sign}(z_3 - 5), \quad \hat{g}_2 = z_4 - 2\text{sign}(z_3 - 5),$$
$$\hat{g}_3 = z_4 - 0.59 - 2\text{sign}(z_3 - 5), \quad \hat{g}_4 = z_2$$

Theorem IV.1 can be used to compute the input and state constraint sets for the abstract system. The linear programming formulation indicates that the abstract

Proceedings of the American Control Conference
Denver, Colorado June 4-6, 2003

dynamics in mode 1 of the hybrid system (14) with input and state constraints given below can simulate the dynamics of mode 1 in the original hybrid system (13):

Abstract State and Input Constraints				
	z_1	z_2	z_3	z_4
max	3.36	19.748	307.27	125.80
min	0	0	−180.17	−132.66
	w_1	w_2	w_3	w_4
max	25.986	28.234	27.4551	$5.8554 \cdot 10^{-2}$
min	−35.858	−33.643	−18.134	$-5.8844 \cdot 10^{-2}$

The simulation relation between (14) and (13) implies a containment of trajectories: the image of all trajectories of (13) under the linear abstraction map H, is a subset of the trajectories that can be generated by (14). Therefore, if all trajectories of the abstraction (14) satisfy the specification, so will the trajectories of the original system (13). The problem then reduces to verifying the specifications on the lower dimensional hybrid system (14), a task that is within the computational capabilities of available tools.

VI. Conclusions

In this paper we establish necessary and sufficient conditions for simulation relations between two constrained, discrete-time linear systems. The simulation conditions derived are expressed in a set-inclusion form since constraints do not allow simple algebraic descriptions. We provide efficient computational means of checking those conditions based on a linear programming formulation which in addition reveals the intrinsic game-theoretic nature of simulation relations. Our computational approach gives a tool for appropriately constraining one of the two systems in order to achieve the desired simulation relation. Furthermore, the computational tool provided by the algorithm allows one to actually measure how close any two systems are to being similar and help addressing issues such as robustness of simulation relations, which is an area for further research.

Acknowledgment. This research is partially supported by DARPA MICA Contract Number N66001-01-C-8076.

References

[1] R. Alur, T. Henzinger, G. Lafferriere, and G.J. Pappas. Discrete abstractions of hybrid systems. *Proceedings of the IEEE*, 88(7):971–984, July 2000.

[2] Rajeev Alur, Radu Grosu, Insup Lee, and Oleg Sokolsky. Compositional refinements for hierarchical hybrid systems. In *Hybrid Systems : Computation and Control*, volume 2034 of *Lecture Notes in Computer Science*, pages 33–48. Springer Verlag, 2001.

[3] M. Aoki. Control of large scale dynamic systems by aggregation. *IEEE Transactions on Automatic Control*, 13(3):246–253, June 1968.

[4] P.E. Caines and Y.J. Wei. The hierarchical lattices of a finite state machine. *Systems and Control Letters*, 25:257–263, 1995.

[5] T. Dang and O. Maler. Reachability analysis via face lifting. In T. Henzinger and S. Sastry, editors, *Hybrid Systems : Computation and Control*, volume 1386 of *Lecture Notes in Computer Science*, pages 96–109. Springer Verlag, Berlin, 1998.

[6] T.A. Henzinger, P.-H. Ho, and H. Wong-Toi. A user guide to HYTECH. In E. Brinksma, W.R. Cleaveland, K.G. Larsen, T. Margaria, and B. Steffen, editors, *TACAS 95: Tools and Algorithms for the Construction and Analysis of Systems*, volume 1019 of *Lecture Notes in Computer Science 1019*, pages 41–71. Springer-Verlag, 1995.

[7] Thomas A. Henzinger, Marius Minea, and Vinayak Prabhu. Assume-guarantee reasoning for hierarchical hybrid systems. In *Hybrid Systems : Computation and Control*, volume 2034 of *Lecture Notes in Computer Science*, pages 275–290. Springer Verlag, 2001.

[8] X. Koutsoukos, P. Antsaklis, J.Stiver, and M. Lemmon. Supervisory control of hybrid systems. *Proceedings of the IEEE*, 88(7):1026–1049, July 2000.

[9] K. Larsen, P. Pettersson, and W. Yi. UPPAAL in a nutshell. *Springer International Journal of Software Tools for Technology Transfer*, 1(1), 1997.

[10] Nancy Lynch, Roberto Segala, and Frits Vaandrager. Hybrid i/o automata revisited. In *Hybrid Systems : Computation and Control*, volume 2034 of *Lecture Notes in Computer Science*, pages 403–417. Springer Verlag, 2001.

[11] Z. Manna and A. Pnueli. *Temporal Verification of Reactive Systems: Safety.* Springer Verlag, New York, 1995.

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

[13] G. J. Pappas and G. Lafferriere. Hierarchies of stabilizability preserving linear systems. In *Proceedings of the 40th IEEE Conference in Decision and Control*, Orlando, FL, December 2001.

[14] G. J. Pappas, G. Lafferriere, and S. Sastry. Hierarchically consistent control systems. *IEEE Transactions on Automatic Control*, 45(6):1144–1160, June 2000.

[15] G. J. Pappas and S. Simic. Consistent abstractions of affine control systems. *IEEE Transactions on Automatic Control*, 2001. Submitted.

[16] B. Silva, K. Richeson, B. Krogh, and A. Chutinan. Modeling and verifying hybrid dynamic systems using CheckMate. In *Proceedings of the* 4^{th} *International Conference of Mixed Processes: Hybrid Dynamic Systems*, pages 323–328, 2000.

[17] B. I. Silva, O. Stursberg, B. H. Krogh, and S. Engell. An assessment of the current status of algorithmic approaches to the verification of hybrid systems. In *Proceedings of the* 40^{th} *IEEE Conference in Decision and Control*, pages 2867–2874, Orlando, Florida, December 2001.

[18] Herbert G. Tanner and George J. Pappas. Simulation relations for discrete-time linear systems. In *Proceedings of the 15th IFAC World Congress*, Barcelona, Spain, July 2002.

[19] F.D. Torrisi, A. Bemporad, and D. Mignone. Hysdel - a tool for generating hybrid models. Technical Report AUT00-03, ETP, Automatic Control Laboratory, October 2000.

[20] A. J. van der Schaft and J. M. Schumacher. Compositionality issues in discrete, continuous, and hybrid systems. *International Journal of Robust and Nonlinear Control*, 11(5):417–434, April 2001.

[21] K.C. Wong and W.M. Wonham. Hierarchical control of discrete-event systems. *Discrete Event Dynamic Systems*, 6:241–273, 1995.

[22] W.M. Wonham. *Linear Multivariable Control : A Geometric Approach*, volume 10 of *Applications of Mathematics*. Springer-Verlag, New York, 1985.