Abstractions of Constrained Linear Systems

Herbert G. Tanner  
*University of Pennsylvania*

George J. Pappas  
*University of Pennsylvania*, pappasg@seas.upenn.edu

Follow this and additional works at: [https://repository.upenn.edu/ese_papers](https://repository.upenn.edu/ese_papers)

**Recommended Citation**


This material is posted here with permission of the IEEE. Such permission of the IEEE does not in any way imply IEEE endorsement of any of the University of Pennsylvania's products or services. Internal or personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution must be obtained from the IEEE by writing to pubs-permissions@ieee.org. By choosing to view this document, you agree to all provisions of the copyright laws protecting it.

This paper is posted at ScholarlyCommons, [https://repository.upenn.edu/ese_papers/102](https://repository.upenn.edu/ese_papers/102)  
For more information, please contact repository@pobox.upenn.edu.
Abstractions of Constrained Linear Systems

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.

Comments

This material is posted here with permission of the IEEE. Such permission of the IEEE does not in any way imply IEEE endorsement of any of the University of Pennsylvania's products or services. Internal or personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution must be obtained from the IEEE by writing to pubs-permissions@ieee.org. By choosing to view this document, you agree to all provisions of the copyright laws protecting it.
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 conditions for one constrained linear system to simulate the constraints, we consider a control-abstract embedding into a

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 \( T = (Q, \Sigma, \rightarrow) \) that consists of:

- A (possibly infinite) set \( Q \) of states,
- A (possibly infinite) set \( \Sigma \) of labels,
- A transition relation \( \rightarrow \subseteq Q \times \Sigma \times Q \),

The transition \((q_1, \sigma, q_2) \in \rightarrow\) is commonly denoted as \( q_1 \xrightarrow{\sigma} q_2 \). The transition system is called finite if \( Q \) and \( \Sigma \) are finite, and infinite otherwise. A region is a subset...
$P \subseteq Q$ of the states. The $\sigma$-successor of a region $P$ is defined as the set that can be reached from $P$ with one $\sigma$-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 $T_1 = (Q_1, \Sigma, \rightarrow_1)$ and $T_2 = (Q_2, \Sigma, \rightarrow_2)$ be two transition systems over the same label set $\Sigma$. 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 $\sigma$-transition taken by $T_1$ can be matched (or implemented) by a $\sigma$-transition of $T_2$. The label set $\Sigma$ 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, $T_2$ also simulates $T_1$ 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:

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

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

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

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 $T_\Delta = (Q, \Sigma, \rightarrow)$ generated by $\Delta$ consists of:

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

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

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:

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

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

where matrices $A, B, F, G$ are of appropriate dimension. Linear systems $\Delta_1$ and $\Delta_2$ generate various transition systems $T_{\Delta_1}$ and $T_{\Delta_2}$, 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 $Q_2 = Z \subseteq \mathbb{R}^p$ where

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

where $H \in \mathbb{R}^{p \times n}$ 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 $Hz + Y \subseteq Q_2$.

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 $Hz$ is surjective, we are interested in simulating the transitions of $\Delta_1$ by a system $\Delta_2$, 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 $Hz$ is injective and $Y = R(H)^1$ (the orthogonal complement of the range of $H$) we are interested in the more complicated system $\Delta_2$ simulating the transitions of the simpler system $\Delta_1$, thus refining the transitions from the simpler to the more complicated model.

**Theorem III.2 (Simulation)** Consider discrete time systems $\Delta_1$ and $\Delta_2$ given by (4)-(5), and a relation $S$ of the form (6). Then $T_{\Delta_1}$ simulates $T_{\Delta_2}$, 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 \( \sigma \) being a one-step transition, \( A_1 \) simulates \( A_2 \) with respect to the relation \( S \) if and only if for all \( (x, z) \in S \) it holds that: \( \forall z \in Post_1(x), \exists z' \in Post_1(z) : (x', z') \in S \). Given (6), the above is rewritten as: \( \forall (x, z) \in S, \forall z' \in Post_1(x), \exists y_1 \in Y : z' = Hx' + y_1 \in Post_1(z) \). Definition III.1 provides explicit expressions for the Post_1 operators \( T_{A_1} \) and \( T_{A_2} \). Substituting, the necessary and sufficient condition for simulation becomes:

\[
\forall (x, z) \in S, \exists y_1 \in U : Ax + Bu \in X, \exists z' \in Y, \exists y_1 \in Y : H(Ax + Bu) + y_1 = Fz + Gu \in Z
\]

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:

\[
\forall z \in X, \exists y_2 \in Y, \forall u \in U : Ax + Bu \in X, \exists y_1 \in Y, \exists u' \in V : H(Ax + Bu) + y_1 = F(Hx + y_2) + Gu \in Z
\]

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:

\[
AX + BU \subseteq X,
FZ + GV \subseteq Z
\]

restrict transitions that do not lead to admissible states. \( \square \)

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:

\[
X = \{ x \in \mathbb{R}^n | C_2 x \leq d_z \}, \quad U = \{ u \in \mathbb{R}^m | C_u u \leq d_u \},
Z = \{ z \in \mathbb{R}^r | C_z z \leq d_z \}, \quad V = \{ v \in \mathbb{R}^r | C_v v \leq d_v \},
Y = \{ y \in \mathbb{R}^r | C_y y \leq d_y \}.
\]

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

\[
P_1 = \{ q = (x, u, y)^T | P_1q \leq d_1 \} \quad (8a)
\]

\[
P_r = \{ w = (y, v)^T | P_r w \leq d_r \} \quad (8b)
\]

where:

\[
P_1 \triangleq \text{diag} \{ C_2, C_u, C_y \}, \quad d_1 \triangleq (d_z, d_u, d_y)^T,
P_r \triangleq \text{diag} \{ C_y, C_v \}, \quad d_r \triangleq (d_y, d_v)^T.
\]

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

\[
C_x Ax + C_z Bu \leq d_z, \quad C_y Fy + C_v Gu \leq d_v - C_z FH x.
\]

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

\[
C_1 q \leq d_z, \quad C_2 w \leq d_v - C_3 q. \quad (9)
\]

Now define the linear maps:

\[
M_1 : P_1 \rightarrow P_1; \quad q \mapsto [HA - FH \ 0] q
M_r : P_r \rightarrow P_2; \quad w \mapsto [-I \ G] w
\]

Clearly, the image of a polyhedron under a linear map is itself a polyhedron. The simulation condition (7a) then requires the inclusion \( P_1 \subseteq P_2 \). The following theorem offers a computational means of checking this inclusion:

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

\[
\begin{align*}
\min \quad & p_k^x (I - M_r^+ M_r) s \\
\text{s.t.} \quad & P_1 (I - M_r^+ M_r) s \leq d_r - P_r M_r^+ M_l q_k^* \\
& C_3 (I - M_r^+ M_r) s \leq d_v - (C_3 + C_2 M_r^+ M_l) q_k^*
\end{align*}
\]

where \( p_k^x \) is the \( k \)th row of \( P_1 \), \( M_r^+ \) is the pseudoinverse of \( M_r \) and \( q_k^* = (x^*, u^*, y^*)_k^* \) is the solution of

\[
\begin{align*}
\max \quad & p_k^x M_r^* M_l q \\
\text{s.t.} \quad & P_l q \leq d_l, \quad C_1 q \leq d_z.
\end{align*}
\]

Proof: If \( P_1 \) and \( P_2 \) are given as:

\[
P_1 = \{ t | P_1 t \leq d_1 \}, \quad P_2 = \{ t | P_2 t \leq d_2 \}
\]

then the checking condition \( P_1 \subseteq P_2 \) is equivalent to verifying that \( p_k^2 t^* \leq d_2^* \) with \( j \) ranging over the number of rows of \( P_2 \), where \( t^* \) is the solution of the LP problem:

\[
\begin{align*}
\max \quad & p_k^x t^* M_r^+ M_l q \\
\text{s.t.} \quad & P_1 t \leq d_1, \quad C_1 q \leq d_z
\end{align*}
\]

The explicit description of \( P_1 \) and \( P_2 \) requires vertex representation of \( P_1 \) and \( P_r \), which is generally difficult. Thus, a problem formulation in the original space where \( P_1 \) and \( P_r \) 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

\[
\begin{align*}
\max \quad & p_k^x M_r^+ M_l q \\
\text{s.t.} \quad & P_1 q \leq d_1, \quad C_1 q \leq d_z
\end{align*}
\]
where \( j \) ranges over the number of rows of \( P_t \).

Let \( z_j^* \) be the solution of (11). Then, \( z_j^* \) is the point in \( P_t \) with an image under \( M_t^{-1} \) \( M_t \), \( (M_t^{-1} M_t) \) denoting the inverse mapping), which is the "worst" among all points on hyperplane \( p_t w = c \), with respect to containment in \( P_2 \). For that point to be contained in \( P_2 \), the LP problem:

\[
\min_s \; p_t^T (M_t^T M_t z_j^* + (I - M_t^T M_t) s)
\]

s.t. \( P_t (M_t^T M_t z_j^* + (I - M_t^T M_t) s) \leq d \),

\( C_2 (M_t^T M_t z_j^* + (I - M_t^T M_t) s) \leq d_2 - C_3 z_j^* \)

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

\[
\min_s \; p_t^T (I - M_t^T M_t) s \quad (12a)
\]

s.t. \( P_t (I - M_t^T M_t) s \leq d_t - P_t M_t z_j^* \),

\( C_2 (I - M_t^T M_t) s \leq d_2 - (C_3 + C_2 M_t^T M_t) z_j^* \) \quad (12c)

Theorem IV.1 reveals the game-theoretic interpretation of simulation condition (7a), where system \( \Delta_t \) first picks the worst transition by maximizing \((x^*, u^*, y^*)\), which must then be matched by \( \Delta_2 \) 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 \( n_r \), where \( n_r \) is the number of faces describing \( P_t \). 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 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:

\[
x[k+1] = A_i x[k] + B_i u[k]
\]

\[
C_s^T x \leq d_s^T, \quad C_u^T u \leq d_u^T, \quad i = 1, \ldots, 6 \quad (13b)
\]

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

\[
B_1 = \begin{bmatrix}
2.58 \times 10^{-1} & 2.07 \times 10^{-5} & 0 \\
2.91 \times 10^{-1} & 3.79 \times 10^{-4} & 0 \\
1.10 \times 10^{-5} & 7.00 \times 10^{-5} & 0 \\
3.33 \times 10^{-2} & 1.40 \times 10^{-1} & 0 \\
0 & 0 & -3.12 \times 10^{-9} \\
0 & 0 & -3.44 \times 10^{-10} \\
0 & 0 & -3.77 \times 10^{-10} \\
0 & 0 & -7.84 \times 10^{-11} \\
0 & 0 & 1.00 \times 10^{-3}
\end{bmatrix}
\]
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.

<table>
<thead>
<tr>
<th>Concrete State and Input Constraints</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$x_4$</th>
<th>$x_5$</th>
<th>$x_6$</th>
</tr>
</thead>
<tbody>
<tr>
<td>max</td>
<td>7.4</td>
<td>27.3</td>
<td>1.62</td>
<td>19.7</td>
<td>1.57</td>
<td>1.57</td>
</tr>
<tr>
<td>min</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$u_1$ $u_2$ $u_3$ $x_7$ $x_8$ $x_9$</td>
<td>max</td>
<td>12</td>
<td>-1</td>
<td>1.57</td>
<td>1.57</td>
<td>1.57</td>
</tr>
<tr>
<td>min</td>
<td>12</td>
<td>-1</td>
<td>0</td>
<td>-1.57</td>
<td>1.57</td>
<td>1.57</td>
</tr>
</tbody>
</table>

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:

$$H = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \ -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -7 \ 0.6387 & -0.0836 & 0.332 & -0.3585 & 0.4914 & 0.0 \
\end{bmatrix}$$

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

$$z[k+1] = F_1 z[k] + G_1 u[k]$$

$$C_i z \leq d_i$$

where $F_1 = H A_i H^T$ and $G_1 = [H B_i, H A_i K_i (H)]$, 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:

$$F_1 = \begin{bmatrix} 1.00 & 10^{-3} & -8.10^{-10} & -1.50 & 10^{-10} \\
\end{bmatrix}$$

$$G_1 = \begin{bmatrix} 1.00 & 10^{-3} & -8.10^{-10} & -1.50 & 10^{-10} \\
\end{bmatrix}$$

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 system...

![Fig. 3. The hybrid system modeling the original ETC System.](image)

![Fig. 4. The abstract hybrid system for ETC.](image)
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):

<table>
<thead>
<tr>
<th>Abstract State and Input Constraints</th>
<th>$2_1$</th>
<th>$2_2$</th>
<th>$2_3$</th>
<th>$2_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>max</td>
<td>3.36</td>
<td>19.74</td>
<td>207.27</td>
<td>120.80</td>
</tr>
<tr>
<td>min</td>
<td>0</td>
<td>0</td>
<td>-180.17</td>
<td>-132.66</td>
</tr>
<tr>
<td>$1_1$</td>
<td>$1_2$</td>
<td>$1_3$</td>
<td>$1_4$</td>
<td></td>
</tr>
<tr>
<td>max</td>
<td>25.986</td>
<td>28.524</td>
<td>27.455</td>
<td>5.8554 x 10^{-2}</td>
</tr>
<tr>
<td>min</td>
<td>-35.858</td>
<td>-33.543</td>
<td>-18.124</td>
<td>-5.8944 x 10^{-2}</td>
</tr>
</tbody>
</table>

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