Proceedings of the
46th IEEE Conference on Decision and Control
New Orleans, LA, USA, Dec. 12-14, 2007

Approximate Hierarchies of Linear Control Systems

Antoine Girard and George J. Pappas

Abstract— **Recently, a hierarchical control approach based on the notion of approximate simulation relations has been introduced. The proposed hierarchical control architecture consists of a precise model (the concrete system) of the plant to be controlled and of a rough model (the abstract system) of the plant that is used for control synthesis. In this paper, we consider the problem of computing approximate simulation relations for linear control systems. For stabilizable systems, we give an effective characterization of simulation functions (*i.e.* functions whose level sets define approximate simulation relations) and compute the associated interfaces which allow the refinement of the control inputs of the abstract system into control inputs of the concrete system. Then, we propose a procedure for the computation of abstractions of linear control systems that can be used in our hierachical control framework. Finally, as an example of application, we use our approach to synthesize hierarchical controllers of a class of high-order systems.**

I. INTRODUCTION

Controlling complex (nonlinear and/or high-order) systems in order to achieve sophisticated tasks constitutes one of the great challenges of modern engineering. Handling at once both complexities of the dynamics and of the specification often leads to untractable problems and therefore a hierarchical approach to control synthesis is highly desirable.

A hierarchical control architecture consists of (at least) two layers. The first layer consists of a precise (and complex) model of the plant that need to be controlled and is usually referred to as the *concrete system*. The second layer consists of a coarse (and simple) model of the plant that is used for synthesis and is referred to as the *abstract system* or *abstraction*. The main challenge of such approaches is the refinement of control laws designed for the abstract system in order to control the concrete system.

An approach using a hierarchy of consistent continuous abstractions of continuous systems has been proposed in [11], [15]. It is based on the notion of *simulation relation*, widely used in computer science for discrete systems [1], [9], and extended in several papers to the continuous and hybrid settings [6], [10], [11], [14], [16]. More recently, it has been claimed that approaches based on *approximate simulation relations* [5] would provide more robust control laws while allowing to consider simpler discrete [13] or continuous [3] abstractions for control synthesis. The effectiveness of this approach has been demonstrated in [2] where it is used successfully for the hierarchical synthesis of hybrid controllers for dynamic robots from temporal logic specifications.

In this paper, we continue the development of this hierarchical control framework for linear systems. The paper is organized as follows. In Section II, we briefly present some existing results [11], [16] on abstraction using *exact* simulation relations. In Section III, we recall the main results of [3] and show how the control inputs of the abstraction can be refined into control inputs of the concrete system using an interface associated to an approximate simulation relation. The main contributions of the paper are in Sections IV and V. In Section IV, for stabilizable linear systems, we give an effective characterization of simulation functions (*i.e.* functions whose level sets define approximate simulation relations) and compute the associated interfaces which allow the refinement of the control inputs. In Section V, we propose a procedure for the computation of abstractions of linear control systems that can be used in our hierachical control framework. Finally, in Section VI, as an example of application, we use our approach to synthesize controllers of a class of high-order systems.

II. ABSTRACTION USING SIMULATION RELATIONS

Let us consider two linear control systems:

$$\Sigma : \begin{cases} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) \end{cases} \tag{1}$$

where $x(t) \in \mathbb{R}^n$, $x(0) = \bar{x}_0$, $u(t) \in \mathbb{R}^p$, $y(t) \in \mathbb{R}^k$ and

$$\Sigma' : \begin{cases} \dot{z}(t) &= Fz(t) + Gv(t) \\ w(t) &= Hz(t) \end{cases} \tag{2}$$

where $z(t) \in \mathbb{R}^m$, $z(0) = \bar{z}_0$, $v(t) \in \mathbb{R}^q$, $w(t) \in \mathbb{R}^k$. Note that both systems have the same observation space (*i.e.* $\mathbb{R}^k$), but may have different input spaces. We assume, without loss of generality, that $\text{rank}(B) = p$ and $\text{rank}(C) = k$. In this paper, we will refer to Σ as the *concrete system*, that is a (complex) system that we actually want to control. Σ' is referred to as the *abstract system*, that is a (simple) system we want to use to design a controller for Σ. We therefore assume throughout the paper that $m \leq n$.

Abstraction of linear control systems using simulation relations has been studied in [11] and subsequently in [16]. In this section, we summarize some results of these works.

Definition 2.1: A relation $\mathcal{W} \subseteq \mathbb{R}^n \times \mathbb{R}^m$ is a simulation relation of Σ by Σ' if for all $(x_0, z_0) \in \mathcal{W}$,

1) $Cx_0 = Hz_0$
2) For all state trajectories $x(.)$ of Σ such that $x(0) = x_0$, there exists a state trajectory $z(.)$ of Σ' such that

This research has been partially supported by the National Science Foundation ITR Grant 0121431.

A. Girard is with the Laboratoire Jean Kuntzmann, Université Joseph Fourier, B.P. 53, 38041 Grenoble Cedex 9, France. Antoine.Girard@imag.fr

George J. Pappas is with the Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA 19104, USA. pappasg@seas.upenn.edu

$z(0) = z_0$ and satisfying for all $t \geq 0$, $(x(t), z(t)) \in \mathcal{W}$.

If $(\bar{x}_0, \bar{z}_0) \in \mathcal{W}$ then we say that Σ' simulates Σ (denoted $\Sigma \preceq \Sigma'$).

Clearly, if $\Sigma \preceq \Sigma'$, it follows that for all observed trajectories $y(.)$ of Σ there is an observed trajectory $w(.)$ of Σ' that satisfies for all $t \geq 0$, $w(t) = y(t)$. This property makes Σ' a suitable abstraction of Σ since it captures all its possible behaviors. For linear systems, interesting abstractions can be obtained through a surjective linear abstraction map Π (*i.e.* for all $t \geq 0$, $z(t) = \Pi x(t)$). This is captured by the notion of Π related systems introduced in [11].

Definition 2.2: Let $\Pi : \mathbb{R}^n \to \mathbb{R}^m$ be a surjective linear map. Σ' is Π-related to Σ if

1) $C = H\Pi$
2) For all $x \in \mathbb{R}^n$, $u \in \mathbb{R}^p$, there exists $v \in \mathbb{R}^q$ such that

$$\Pi(Ax + Bu) = F\Pi x + Gv.$$

If Σ' is Π-related to Σ, then it is easy to show that the relation given by

$$\mathcal{W} = \{(x, z) \in \mathbb{R}^n \times \mathbb{R}^m |\ \Pi x = z\}$$

is a simulation relation of Σ by Σ'. The following proposition is straightforward:

Proposition 2.3: If Σ' is Π-related to Σ and $\bar{z}_0 = \Pi \bar{x}_0$, then $\Sigma \preceq \Sigma'$.

III. Refinement using Approximate Simulation Relations

In the previous section, we presented an abstraction framework for a linear control system. Control design is usually much simpler for the abstract system Σ' which is typically of lower dimension. The main challenge that remains is how to refine the synthesized control inputs so that they can be used to control the concrete system Σ.

The fundamental problem is that all the observed trajectories of Σ are observed trajectories of Σ' but the converse does not hold (except if Σ and Σ' are bisimilar [10]). Thus, generally, we cannot refine an input $v(.)$ of Σ' into an input $u(.)$ of Σ such that the observed trajectory $y(.)$ of Σ track exactly the observed trajectory $w(.)$ of Σ' (*i.e.* for all $t \geq 0$, $y(t) = w(t)$). However, sometimes, this can be done approximately with a guaranteed error bound δ (*i.e.* for all $t \geq 0$, $\|y(t) - w(t)\| \leq \delta$). In [3], we presented an approach based on the notion of approximate simulation relation. The main results are reviewed in this section.

A. Approximate simulation relations

Let Σ'_ν denote the system Σ' where the inputs $v(.)$ are uniformely bounded by ν (*i.e.* for all $t \geq 0$, $\|v(t)\| \leq \nu$).

Definition 3.1: A relation $\mathcal{W}_\delta \subseteq \mathbb{R}^m \times \mathbb{R}^n$ is an approximate simulation relation of precision δ of Σ'_ν by Σ if for all $(z_0, x_0) \in \mathcal{W}_\delta$,

1) $\|Cx_0 - Hz_0\| \leq \delta$
2) For all state trajectories $z(.)$ of Σ'_ν such that $z(0) = z_0$, there exists a state trajectory $x(.)$ of Σ such that $x(0) = x_0$ and satisfying for all $t \geq 0$, $(z(t), x(t)) \in \mathcal{W}_\delta$.

If $(\bar{z}_0, \bar{x}_0) \in \mathcal{W}_\delta$ then we say that Σ approximately simulates Σ'_ν with precision δ (denoted $\Sigma'_\nu \preceq_\delta \Sigma$).

Then, if $\Sigma'_\nu \preceq_\delta \Sigma$, it follows that for all observed trajectories $w(.)$ of Σ' associated to an input $v(.)$ uniformely bounded by ν, there is an observed trajectory $y(.)$ of Σ that satisfies for all $t \geq 0$, $\|y(t) - w(t)\| \leq \delta$. Let us remark that for $\delta = 0$, we recover the usual notion of *exact* simulation relation.

B. Simulation function

The construction of approximate simulation relations can be done effectively using a simulation function.

Definition 3.2: Let $\mathcal{V} : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}^+$ be a differentiable function and $u_\mathcal{V} : \mathbb{R}^q \times \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}^p$ be a continuous function. $\mathcal{V}$ is a simulation function of Σ' by Σ and $u_\mathcal{V}$ is an associated interface if there exists $\gamma \geq 0$ such that for all $(z, x) \in \mathbb{R}^m \times \mathbb{R}^n$,

$$\mathcal{V}(z, x) \geq \|Cx - Hz\| \tag{3}$$

and for all $v \in \mathbb{R}^q$ satisfying $\gamma \|v\| < \mathcal{V}(z, x)$,

$$\frac{\partial \mathcal{V}(z,x)}{\partial z}(Fz + Gv) + \frac{\partial \mathcal{V}(z,x)}{\partial x}(Ax + Bu_\mathcal{V}(v, z, x)) < 0 \tag{4}$$

γ is called the precision gain of the simulation function.

An interesting property of simulation functions is that their level sets define approximate simulation relations of Σ'_ν by Σ.

Theorem 3.3: Let $\nu \geq 0$, and $\delta \geq \gamma\nu$, then

$$\mathcal{W}_\delta = \{(x, z) \in \mathbb{R}^n \times \mathbb{R}^m |\ \mathcal{V}(z, x) \leq \delta\}$$

is an approximate simulation relation of precision δ of Σ'_ν by Σ.

Proof: Let $(z_0, x_0) \in \mathcal{W}_\delta$, then we have

$$\|Cx_0 - Hz_0\| \leq \mathcal{V}(z_0, x_0) \leq \delta.$$

Thus, the first condition of Definition 3.1 holds. Let $z(.)$ be a state trajectory of Σ'_ν such that $z(0) = z_0$, let $v(.)$ be the associated input ($\forall t \geq 0$, $\|v(t)\| \leq \nu$). Let $x(.)$ be the solution of the differential equation:

$$\dot{x}(t) = Ax(t) + Bu_\mathcal{V}(v(t), z(t), x(t)),\ x(0) = x_0. \tag{5}$$

Clearly, $x(.)$ is a state trajectory of Σ (associated to the input $u(.) = u_\mathcal{V}(v(.), z(.), x(.))$). Initially, $\mathcal{V}(z(0), x(0)) \leq \delta$. Let us assume that there exists $\tau > 0$, such that $\mathcal{V}(z(\tau), x(\tau)) > \delta$. Then, there exists $0 \leq \tau' < \tau$ such that $\mathcal{V}(z(\tau'), x(\tau')) = \delta$, and for all $t \in (\tau', \tau]$, $\mathcal{V}(z(t), x(t)) > \delta$. Then, for all $t \in (\tau', \tau]$, $\gamma\|v(t)\| \leq \gamma\nu \leq \delta < \mathcal{V}(z(\tau), x(\tau))$. It follows from equation (4) that

$$\forall t \in (\tau', \tau],\ \frac{d\mathcal{V}(z(t), x(t))}{dt} < 0.$$

Therefore,

$$\mathcal{V}(z(\tau), x(\tau)) - \mathcal{V}(z(\tau'), x(\tau')) = \int_{\tau'}^{\tau} \frac{d\mathcal{V}(z(t), x(t))}{dt} dt < 0.$$

This contradicts $\mathcal{V}(z(\tau), x(\tau)) > \mathcal{V}(z(\tau'), x(\tau'))$. Therefore, we proved by contradiction that for all $t \geq 0$, $\mathcal{V}(z(t), x(t)) \leq \delta$. Thus the second condition of Definition 3.1 holds. ■

The existence of a simulation function of Σ' by Σ implies that, for all $\nu \geq 0$, Σ approximately simulates Σ'_ν.

Corollary 3.4: Let $\mathcal{V}$ be a simulation function of Σ' by Σ. Let $\nu \geq 0$, let δ be given by

$$\delta = \max\left(\mathcal{V}(\bar{z}_0, \bar{x}_0), \gamma\nu\right).$$

Then, $\Sigma'_\nu \preceq_\delta \Sigma$.

Proof: Since $\delta \geq \gamma\nu$, $\mathcal{W}_\delta$ is an approximate simulation relation of precision δ of Σ'_ν by Σ. Since $\mathcal{V}(\bar{z}_0, \bar{x}_0) \leq \delta$, it follows that $(\bar{z}_0, \bar{x}_0) \in \mathcal{W}_\delta$. Therefore, $\Sigma'_\nu \preceq_\delta \Sigma$. ■

From the proof of Theorem 3.3, it follows that the control inputs $v(.)$ of the abstract system Σ'_ν can be refined into control inputs $u(.)$ of the concrete system Σ by interconnection of the systems through the interface $u_\mathcal{V}$ as shown on Figure 1. Then, the observed trajectory $y(.)$ of Σ approximately tracks the observed trajectory $w(.)$ of Σ' with precision δ (*i.e.* for all $t \geq 0$, $\|y(t) - w(t)\| \leq \delta$).

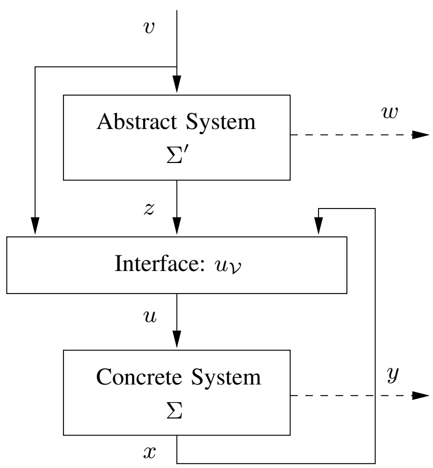

Fig. 1. Hierarchical control architecture.

Remark 3.5: In [2], the hierarchical control architecture shown on Figure 1 has been used to solve a synthesis problem where the specification consisted of a temporal logic property to be satisfied by the observed trajectory $y(.)$ of Σ. First, a control input $v(.)$ would be synthesized for the abstract system Σ'_ν such that the observed trajectory $w(.)$ of Σ'_ν satisfies the specification robustly: any function that remains within distance δ from $w(.)$ also satisfies the specification. Then, using the hierarchical control architecture, we can guarantee that the observed trajectory $y(.)$ of Σ satisfies the specification as well.

Remark 3.6: It is to be noted that approximate simulation of Σ'_ν by Σ (for all $\nu \geq 0$) is actually equivalent to the composite system

$$\left\{\begin{array}{rcl} \dot{z}(t) &=& Fz(t) + Gv(t) \\ \dot{x}(t) &=& Ax(t) + Bu_\mathcal{V}(v(t), z(t), x(t)) \\ e(t) &=& Cx(t) - Hz(t) \end{array}\right.$$

being input to output stable (where $v(.)$ is the input and $e(.)$ is the output). Moreover, the simulation function $\mathcal{V}$ is an input to output stability Lyapunov function for that system [12].

IV. SIMULATION FUNCTIONS FOR LINEAR SYSTEMS

In this section, we give an effective characterization of simulation functions and of their associated interfaces for linear control systems. In the following, we shall assume that the concrete system Σ is stabilizable. Then, there exists a $p \times n$ matrix K such that the matrix $A + BK$ is Hurwitz. The proof of the following proposition is ommited here but can be found in [4].

Proposition 4.1: There exists a positive definite symmetric matrix M and a strictly positive scalar number λ such that the following matrix inequalities hold:

$$M \geq C^T C, \tag{6}$$

$$(A + BK)^T M + M(A + BK) \leq -2\lambda M. \tag{7}$$

We now give a characterization of simulation functions.

Theorem 4.2: Let us assume there exists an $n \times m$ matrix P and a $p \times m$ matrix Q such that the following Sylvester linear matrix equations hold:

$$PF = AP + BQ, \tag{8}$$

$$H = CP. \tag{9}$$

Then, the function defined by

$$\mathcal{V}(z, x) = \sqrt{(x - Pz)^T M (x - Pz)} \tag{10}$$

is a simulation function of Σ' by Σ and an associated interface is given by

$$u_\mathcal{V}(v, z, x) = Rv + Qz + K(x - Pz). \tag{11}$$

where R is an arbitrary $p \times q$ matrix. The precision gain γ of the simulation function is given by

$$\gamma = \frac{\left\|\sqrt{M}(BR - PG)\right\|}{\lambda}$$

and it is minimal for

$$R = (B^T M B)^{-1} B^T M P G. \tag{12}$$

Proof: From equations (6) and (9), we have that

$$\mathcal{V}(x, z)^2 \geq (x - \Pi z)^T C^T C (x - \Pi z) = \|Cx - Hz\|^2.$$

Then, equation (3) holds. Now, let us show that equation (4) holds as well. We have

$$\frac{\partial \mathcal{V}(z,x)}{\partial z}(Fz + Gv) + \frac{\partial \mathcal{V}(z,x)}{\partial x}(Ax + Bu_\mathcal{V}) = \frac{(x - Pz)^T M (Ax + B(Rv + Qz + K(x - Pz)) - P(Fz + Gv))}{\sqrt{(x - Pz)^T M (x - Pz)}}$$

From equation (8), it follows that

$$\frac{\partial \mathcal{V}(z,x)}{\partial z}(Fz + Gv) + \frac{\partial \mathcal{V}(z,x)}{\partial x}(Ax + Bu_\mathcal{V}) = \frac{(x - Pz)^T M ((A + BK)(x - Pz) + (BR - PG)v)}{\sqrt{(x - Pz)^T M (x - Pz)}}$$

Using equation (7), we can show that

$$\frac{\partial \mathcal{V}(z,x)}{\partial z}(Fz + Gv) + \frac{\partial \mathcal{V}(z,x)}{\partial x}(Ax + Bu_\mathcal{V}) \leq \frac{-\lambda (x - Pz)^T M (x - Pz) + (x - Pz)^T M (BR - PG)v}{\sqrt{(x - Pz)^T M (x - Pz)}} \leq$$

$$-\lambda \mathcal{V}(z, x) + \|\sqrt{M}(BR - PG)v\| \leq \lambda(-\mathcal{V}(z, x) + \gamma\|v\|)$$

Therefore, for all $v \in \mathbb{R}^q$ such that $\gamma\|v\| < \mathcal{V}(z,x)$, equation (4) holds. Then, $\mathcal{V}$ is a simulation function of Σ' by Σ, $u_{\mathcal{V}}$ is an associated interface and γ is the precision gain of the simulation function.

The minimization of γ over the matrix R is clearly a least squares approximation problem. Then, since $\text{rank}(B) = p$ and M is definite positive, it follows that the matrix B^TMB is invertible and it is straightforward that the minimal γ is reached for $R = (B^TMB)^{-1}B^TMPG$. ■

The minimization of the precision gain γ is important for the following reason. From Corollary 3.4, the precision δ with which Σ approximately simulates Σ'_ν is greater than $\gamma\nu$. Hence, it is clear that to get the finest precision possible, it is necessary to minimize the precision gain γ.

Remark 4.3: Conditions on the matrices A, B, C, F and H that guarantee the existence of a matrix P such that the Sylvester equations (8) and (9) hold can be found in [7].

Remark 4.4: If the abstract system Σ' does not have any input (*i.e.* $G = 0$), then our problem becomes similar to the classical regulator problem [17]. It is to be noted, for instance, that the Sylvester equations (8) and (9) are key ingredients in the resolution of the regulator problem [7]. Note that if Σ' does not have an input, our approach actually solves the regulator problem. Indeed, it is easy to prove that, in that case, the distance between the observed trajectories $y(.)$ and $w(.)$ converges to 0. Our approach therefore generalizes the regulator problem by giving the possibility of considering an abstract system Σ' with inputs. If the inputs are disturbances, then this can be viewed as robust regulation, but if the input are higher level control signals, this can be viewed as hierarchical regulation. One additional contribution of our approach to the regulator problem is that it makes it possible, in addition, to compute an explicit bound on the distance between the observed trajectories.

V. Construction of the Abstraction

In the previous sections, we assumed that the abstract system Σ' was known a priori. It is clear that this is generally not the case. In this section, we give a procedure to compute an abstraction Σ' such that Σ' *exactly* simulates Σ and for all $\nu \geq 0$, Σ approximately simulates Σ'_ν.

First, we determine conditions on an injective map P such that there exists matrices F, Q satisfying equation (8). Additional conditions on P are determined such that Σ' is P^+-related to Σ where P^+ denotes the Penrose Moore pseudo-inverse of P. Second, given a suitable abstraction map P, we will determine the matrices F, G and H of the abstract system associated to P.

A. Computation of the abstraction map

We start by remarking that equation (8) is equivalent to $AP = PF - BQ$. Then, the following result is straightforward.

Lemma 5.1: Let P be an injective map, there exists matrices F and Q satisfying the linear matrix equation (8) if and only if

$$\text{im}(AP) \subseteq \text{im}(P) + \text{im}(B). \tag{13}$$

Equation (9) gives us the matrix $H = CP$. We determine conditions on P so that Σ' is P^+-related to Σ. From the first condition of Definition 2.2, we must have $C = HP^+$.

Lemma 5.2: Let P be an injective map, let $H = CP$. Then, $C = HP^+$ if and only if

$$\text{im}(C^T) \subseteq \text{im}(P). \tag{14}$$

Proof: Let us assume $\text{im}(C^T) \subseteq \text{im}(P)$, then there exists a matrix D such that $C^T = PD$. It follows that

$$HP^+ = CPP^+ = D^TP^TPP^+.$$

Since P is injective, we have $P^+ = (P^TP)^{-1}P^T$:

$$HP^+ = D^TP^TP(P^TP)^{-1}P^T = D^TP^T = C.$$

We now assume that $C = HP^+$, then let $x \in \ker(P^T)$, then

$$Cx = HP^+x = H(P^TP)^{-1}P^Tx = 0.$$

Therefore $\ker(P^T) \subseteq \ker(C)$. Let us remark that $\text{im}(C^T) = \ker(C)^\perp$ and $\text{im}(P) = \ker(P^T)^\perp$. It follows that we have $\text{im}(C^T) \subseteq \text{im}(P)$. ■

Hence, equations (13) and (14) give necessary conditions for a suitable abstraction map P. The following algorithm allows the computation of such a map.

Algorithm 5.3 (Computation of the abstraction map): Let P_0 be an injective map such that $\text{im}(C^T) \subseteq \text{im}(P_0)$. While $\text{im}(AP_k) \not\subseteq \text{im}(P_k) + \text{im}(B)$:

1) Determine an injective map E_k such that

$$\text{im}(AP_k) = [(\text{im}(P_k) + \text{im}(B)) \cap \text{im}(AP_k)] \oplus \text{im}(E_k). \tag{15}$$

2) Set $P_{k+1} = [P_k \quad E_k]$, $k = k + 1$.

Define the abstraction map $P = P_k$.

For initialization of the algorithm, let us remark that we can use $P_0 = C^T$.

Proposition 5.4: Algorithm 5.3 terminates within a finite number of steps. Let P be the computed map, then P is injective and satisfies equations (13) and (14).

Proof: By assumption, P_0 is injective. Let us assume that P_k is injective. From equation (15), it is clear that $\text{im}(P_k) \cap \text{im}(E_k) = \emptyset$. Let $x \in \ker(P_{k+1})$, then x is of the form $[x_1, x_2]^T$ and $P_{k+1}x = P_kx_1 + E_kx_2 = 0$. Since $\text{im}(P_k) \cap \text{im}(E_k) = \emptyset$, we must have $P_kx_1 = 0$ and $E_kx_2 = 0$. Since P_k and E_k are both injective maps we have $x_1 = 0$, $x_2 = 0$ and therefore $x = 0$. Thus P_{k+1} is injective. We proved by induction that the map P_k is injective for all k. Then, the rank of P_k striclty increases with k and is upper-bounded by n. It follows that Algorithm 5.3 terminates within a finite number of steps K. The computed map $P = P_K$ is injective and satisfies equation (13) since it is the termination condition of the algorithm. Finally, let us remark that $P_K = [P_0 \quad E_0 \dots E_{k-1}]$. Thus, we have $\text{im}(C^T) \subseteq \text{im}(P_0) \subseteq \text{im}(P_K)$. Equation (14) holds as well. ■

B. Computation of the abstract system

In the previous paragraph, we gave an algorithm for the computation of a suitable abstraction map. In this paragraph, given such a map, we compute the associated abstract system Σ' such that Σ' *exactly* simulates Σ and for all $\nu \geq 0$, Σ approximately simulates Σ'_ν.

Proposition 5.5: Let P be an injective map satisfying equations (13) and (14). Let us define the matrices F, G, H and Q as follows,

$$\begin{bmatrix} F \\ Q \end{bmatrix} = [P \quad -B]^+ AP \tag{16}$$

$$G = [P^+ B \quad P^+ A\pi_1 \dots P^+ A\pi_l] \tag{17}$$

$$H = CP \tag{18}$$

where $[P \ -B]^+$ and P^+ denotes the Penrose Moore pseudo inverses of $[P \quad -B]$ and P, and $\pi_1, \dots, \pi_l$ span $\ker(P^T)$. Then, Σ' is P^+-related to Σ and the Sylvester linear matrix equations (8) and (9) hold.

Proof: From equation (18), equation (9) holds. Since equation (13) holds, there exists matrices F' and Q' such that $AP = PF' - BQ'$ which we can rewrite under the form:

$$AP = [P \quad -B] \begin{bmatrix} F' \\ Q' \end{bmatrix}.$$

Then, let F and Q be given by equation (16), we have

$$\begin{aligned} PF - BQ &= [P \quad -B][P \quad -B]^+ AP \\ &= [P \quad -B][P \quad -B]^+ [P \quad -B] \begin{bmatrix} F' \\ Q' \end{bmatrix} \end{aligned}$$

Using the properties of the Penrose Moore pseudo inverse it follows that

$$PF - BQ = [P \quad -B] \begin{bmatrix} F' \\ Q' \end{bmatrix} = AP.$$

Hence, equation (8) holds as well. It remains to prove that Σ' is P^+-related to Σ. Since equation (14) holds, we have $H = CP^+$ and the first condition of Definition 2.2 holds. The second condition of Definition 2.2 is equivalent to say that

$$\text{im}(P^+A - FP^+) + \text{im}(P^+B) \subseteq \text{im}(G). \tag{19}$$

We know that equation (8) holds, moreover since P is injective, we have $P^+P = I$. This leads to

$$F = P^+AP + P^+BQ.$$

Equation (19) is then equivalent to

$$\text{im}(P^+A - P^+APP^+ + P^+BQP^+) + \text{im}(P^+B) \subseteq \text{im}(G)$$

which is also equivalent to

$$\text{im}(P^+A - P^+APP^+) + \text{im}(P^+B) \subseteq \text{im}(G). \tag{20}$$

As shown in [11] (Proposition 5.1), the matrix G defined by equation (17) satisfies equation (20). Hence, Σ' is P^+-related to Σ. ∎

This construction of the abstract system also allows the computation of the matrix Q which is needed for the interface $u_{\mathcal{V}}$. Let us remark that G can obviously be replaced by an injective map G' satisfying $\text{im}(G) = \text{im}(G')$. The initial state of Σ' is given by $\bar{z}_0 = P^+\bar{x}_0$.

Remark 5.6: If $\text{im}(P)$ is an invariant subspace of A, then we have $Q = 0$. In that case the matrix F is given by $F = P^+AP$ which coincides with the construction of abstractions presented in [11].

VI. Example

In this section, we use our approach to control Σ in the following class of systems:

$$\Sigma : y^{(r)}(t) = u(t),\ u(t) \in \mathbb{R}^k,\ y(t) \in \mathbb{R}^k$$

where $r \geq 1$ and $y^{(r)}(.)$ denotes the r^{th} order derivative of the observed output $y(.)$. This system can be written under the form (1) with the following matrices:

$$A = \begin{bmatrix} 0_k & I_k & 0_k & \dots & 0_k \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0_k \\ \vdots & & & \ddots & I_k \\ 0_k & \dots & \dots & \dots & 0_k \end{bmatrix},$$

$$B^T = [0_k \ \dots \ 0_k \ I_k], \quad C = [I_k \ 0_k \ \dots \ 0_k],$$

where I_k denotes the $k \times k$ identity matrix, 0_k denotes the $k \times k$ zero matrix. In [3], for $r = 2$, we computed an abstraction for the system Σ. Using, the procedure presented in the previous section, we show that it is possible for any $r \geq 1$.

We apply Algorithm 5.3, starting with $P_0 = C^T$. Let us remark that $AP_0 = 0_n$. Hence, a suitable abstraction map is $P = C^T$. For the computation of the abstract system, we use the construction procedure presented in Proposition 5.5. Note that $\text{im}(P)$ is an invariant subspace of A. Hence, we have $Q = 0$ and $F = 0_k$, $G = I_k$, and $H = I_k$. Then, the computed abstraction is

$$\Sigma : \dot{w}(t) = v(t),\ v(t) \in \mathbb{R}^k,\ w(t) \in \mathbb{R}^k.$$

In order to use this abstraction for hierarchical control design, we need to compute a simulation function of Σ' by Σ and an associated interface. We use the results presented in Section IV. We start by computing a stabilizing feedback for the system Σ. We consider a stabilizing control matrix K under the form:

$$K = [-\alpha_0 I_k \ \dots \ -\alpha_{r-1} I_k],$$

where $\alpha_0, \dots, \alpha_{r-1} \in \mathbb{R}$. The choice of $\alpha_0, \dots, \alpha_{r-1}$ is done using pole assignment: it is necessary and sufficient that the roots of the polynomial $s^r + \alpha_{r-1}s^{r-1} + \dots + \alpha_1 s + \alpha_0$ have all a strictly negative real part. Then, we compute a positive definite symmetric matrix M satisfying the linear matrix inequalities (6) and (7). This can be done using semi-definite programming. We used the MATLAB toolbox

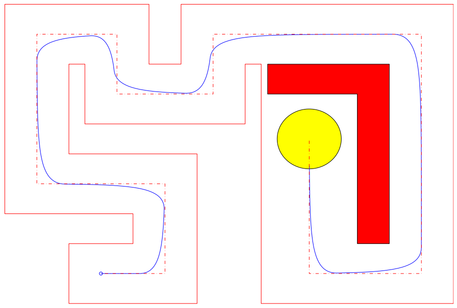

Fig. 2. Trajectory $y(.)$ of the concrete system Σ (plain, blue) and the trajectory $w(.)$ of the abstract system Σ' (dashed, red). $y(.)$ satisfies the specification of the problem.

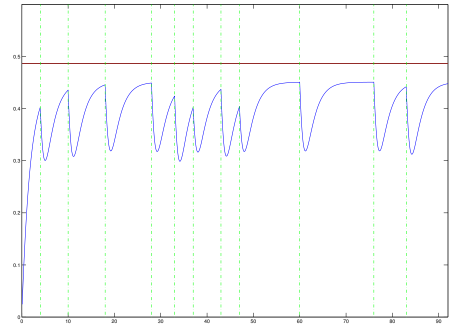

Fig. 3. Value of $\|y(t)-w(t)\|$ for the trajectories of Σ and Σ' presented in Figure 2. We can see that it is bounded by $\gamma\nu = 0.49$ (horizontal line). The vertical lines correspond to the times at which the direction of the trajectory of the abstract system Σ' changes.

YALMIP [8]. From M, we can then compute the matrix R in the interface and the precision gain γ of the simulation function.

For $r = 3$, we designed a simulation function $\mathcal{V}$ of precision gain $\gamma = 1.95$. The associated interface is given by:

$$u_{\mathcal{V}}(v, w, y, \dot{y}, \ddot{y}) = 31.4v - 17.5\ddot{y} - 63\dot{y} - 57.1(y - w). \quad (21)$$

We can now use our approach for a motion planning problem. Assume that Σ describes the (high-order) dynamics of a mobile planar robot. The observed ouput $y(.) \in \mathbb{R}^2$ gives the position of the robot. Initially, $\dot{y}(0) = 0$ and $\ddot{y}(0) = 0$. Particularly, this means that initially the value of the simulation function $\mathcal{V}$ is 0.

We consider the problem of driving the robot in a complex environment shown in Figure 2. It consists of a corridor of width 1. At the end of the corridor, there is a room with an obstacle. The goal of the motion planning problem consists in reaching a target which is a circle of diameter 1, behind the obstacle. Since the abstract system Σ' is fully actuated, it is easy to synthesize a path for this system. This path is represented by the dashed line in Figure 2. It is clear that any trajectory remaining within distance 0.5 from this path satisfies the problem specification. We thus choose a bound ν for the inputs of the abstraction Σ' such that $\gamma\nu \leq 0.5$. Then, from Corollary 3.4, we know that the observed trajectory $y(.)$ remains within distance 0.5 from $w(.)$. Therefore, $y(.)$ satisfies the specification of the motion planning problem.

For $r = 3$, we can chose $\nu = 0.25$ ($\gamma\nu = 0.49 \leq 0.5$). The observed trajectory $y(.)$ obtained by connecting the abstract system and the concrete system through the interface given by equation (21) is represented by the full line in Figure 2. It is clear that it satisfies the specification of the motion planning problem. In Figure 3, we represented the evolution of $\|y(t) - w(t)\|$ for the trajectories of Σ and Σ' presented in Figure 2. We can check that it remains bounded by $\gamma\nu = 0.49$ which is expected from Corollary 3.4. Moreover, we can see that this bound is quite tight.

VII. CONLUSION

In this paper, we gave a characterization of simulation functions for the hierarchical control of linear systems. This characterization allows the computation of the interfaces for the refinement of the control inputs of the abstract system into control inputs of the concrete system. We proposed a procedure for the computation of abstractions of linear control systems that can be used in our hierachical control framework and showed the effectiveness of the developed techniques on an example.

REFERENCES

[1] E. M. Clarke, O. Grumberg, and D. A. Peled, *Model Checking*. MIT Press, 2000.

[2] G. Fainekos, A. Girard, and G. Pappas, "Hierarchical synthesis of hybrid controllers from temporal logic specification." in *Hybrid Systems: Computation and Control*, ser. LNCS, vol. 4416. Springer, 2007, pp. 203–216.

[3] A. Girard and G. Pappas, "Hierarchical control using approximate simulation relations." in *Proc. of the 45th IEEE Conference on Decision and Control*, 2006.

[4] ——, "Approximate bisimulation relations for constrained linear systems," *Automatica*, vol. 43, no. 8, pp. 1307–1317, 2007.

[5] ——, "Approximation metrics for discrete and continuous systems," *IEEE Trans. on Automatic Control*, vol. 52, no. 5, pp. 782–798, 2007.

[6] E. Haghverdi, P. Tabuada, and G. J. Pappas, "Bisimulation relations for dynamical, control, and hybrid systems," *Theo. Comp. Sc.*, vol. 342, pp. 229–261, 2005.

[7] M. Hautus, "Linear matrix equations with applications to the regulator problem," in *Outils et Modèles Mathématiques pour l'Automatique*, I. Landau, Ed. Editions du CNRS, 1983, pp. 399–412.

[8] J. Löfberg, "YALMIP : A toolbox for modeling and optimization in MATLAB," in *Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004. [Online]. Available: http://control.ee.ethz.ch/ joloef/yalmip.php

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

[10] G. J. Pappas, "Bisimilar linear systems," *Automatica*, vol. 39, no. 12, pp. 2035–2047, 2003.

[11] G. Pappas, G. Lafferriere, and S. Sastry, "Hierarchically consistent control systems," *IEEE Trans. on Auto. Cont.*, vol. 45, no. 6, pp. 1144–1160, 2000.

[12] E. Sontag and Y. Wang, "Lyapunov characterizations of input to output stability," *SIAM J. Control Optim.*, vol. 39, no. 1, pp. 226–249, 2000.

[13] P. Tabuada, "Approximate simulation relations and finite abstractions of quantized control systems," in *Hybrid Systems: Computation and Control*, ser. LNCS, vol. 4416. Springer, 2007, pp. 529–542.

[14] P. Tabuada and G. J. Pappas, "Bisimilar control affine systems," *Sys. and Cont. Letters*, vol. 52, no. 1, pp. 49–58, 2004.

[15] ——, "Hierarchical trajectory generation for a class of nonlinear systems," *Automatica*, vol. 41, no. 4, pp. 701–708, 2005.

[16] A. van der Schaft, "Equivalence of dynamical systems by bisimulation," *IEEE Trans. on Automatic Control*, vol. 49, pp. 2160–2172, 2004.

[17] W. Wonham, *Linear Multivariable Control: a Geometric Approach*. Springer, 1979.