European Journal of Control (2011)5-6:568–578
© 2011 EUCA
DOI:10.3166/EJC.17.568–578

European
Journal of
Control

Approximate Bisimulation: A Bridge Between Computer Science and Control Theory

Antoine Girard[1], George J. Pappas[2,*]

[1] UFRIMAG, Université Joseph Fourier, Laboratoire Jean Kuntzmann, 51 rue des Mathématiques, 38041 Grenoble Cedex 9, France
[2] Joseph Moore Professor of Electrical and Systems Engineering, School of Engineering and Applied Science University of Pennsylvania, USA

Fifty years ago, control and computing were part of a broader system science. After a long period of separate development within each discipline, embedded and hybrid systems have challenged us to re-unite the, now sophisticated theories of continuous control and discrete computing on a broader system theoretic basis. In this paper, we present a framework of system approximation that applies to both discrete and continuous systems. We define a hierarchy of approximation metrics between two systems that quantify the quality of the approximation, and capture the established notions in computer science as zero sections. The central notions in this framework are that of approximate simulation and bisimulation relations and their functional characterizations called simulation and bisimulation functions and defined by Lyapunov-type inequalities. In particular, these functions can provide computable upper-bounds on the approximation metrics by solving a static game. Our approximation framework will be illustrated by showing some of its applications in various problems such as reachability analysis of continuous systems and hybrid systems, approximation of continuous and hybrid systems by discrete systems, hierarchical control design, and simulation-based approaches to verification of continuous and hybrid systems.

Keywords: Hybrid systems, control systems, computer science, verification, Lyapunov theory

1. Introduction

Fifty years ago, control and computing were part of a broader system science. In [32], there were attempts to bring the developing theories of state-space control systems and automata theory under a unified system theoretic foundation. Since there has been tremendous progress within each discipline, there was rarely cross-disciplinary communication or intellectual points of connection. The agenda of discrete event control [44] was a great program that developed a control theory for automata-based models, but this program has not resulted in the the two communities getting closer to each other.

It was the agenda of embedded and hybrid systems that was successful in bringing together researchers from theoretical computer science and control theory. Hybrid systems, that integrated discrete computing elements with continuous control and sensing models, are extremely hard to analyze or synthesize. In the early days of hybrid systems, each side tried to bring forward their own models, tools, and methods to address the problems with reasonable success. On the control side, this approach resulted in an established stability theory for switched systems [33], and, on the computer science side, this resulted in a verification theory for timed and hybrid automata [1].

A major challenge in the area of hybrid systems is to think about continuous control and discrete computing on a unified systems theoretic foundation. In particular, theories of system approximation, which are crucial

*Correspondence to: G.J. Pappas, E-mail: pappasg@seas.upenn.edu

Received 15 August 2011; Accepted 15 September 2011
Recommended by Eduardo F. Camacho

for the application of analysis and synthesis techniques for complex systems, have been developed independently on both sides. In control, approximation is traditionally tackled through metrics between transfer functions [4]. In computer science, abstraction relationships for discrete systems such as language inclusion, simulation and bisimulation relations [35, 7] require equality of observed behaviors. In the past decade, these notions have been extended in continuous control systems [37, 51, 27] providing a basis for a unified theory of system approximation.

The notions of language inclusion, simulation, and bisimulation for both discrete and continuous systems are all *exact*, requiring external behavior of two systems to be identical. As exact relationships between systems do not permit any error, there are clear limitations in the amount of system compression or approximation that can be achieved. Approximate relationships which do allow for the possibility of error, will certainly allow for more dramatic system compression. Even though this has been the tradition for deterministic continuous systems [4], it has been recently argued convincingly [5, 39], that even for more quantitative classes of finite transition systems, such as probabilistic automata [50], labeled Markov processes [12], and quantitative transition systems [11], notions of system approximation are not only better candidates for complexity reduction but also provide more robust relationships between systems. The challenge in developing approximate system relationships is the quantification of the quality of the approximation.

The goal of this paper is to review our developed theory of system approximation that applies to both finite (discrete) and infinite (continuous) transition systems by providing approximate generalizations of language inclusion, simulation, and bisimulation. Our approximation framework has been introduced in [22], which applies equally to discrete and continuous systems. It is based on a hierarchy of approximation metrics, which generalizes the usual *exact* relationship hierarchy of language inclusion, simulation and bisimulation. These metrics essentially quantify how well a system is approximated by another based on the distance between their observed behaviors. The traditional relationships are captured as the zero sections of these approximation metrics. The central notions in this framework are that of approximate simulation and bisimulation relations and their functional characterizations called simulation and bisimulation functions and defined by Lyapunov-type inequalities. In particular, these functions show to be very useful to compute guaranteed upper-bounds on the approximation metrics by solving a static game.

This paper aims at presenting the main features of this approximation framework and at showing some of its applications in various problems such as reachability analysis of continuous systems and hybrid systems, approximation of continuous and hybrid systems by discrete systems, hierarchical control design and simulation-based approaches to verification of continuous and hybrid systems.

2. Approximation Metrics

In this section, we summarize the main results of [22] presenting an approximation framework based on a hierarchy of metrics and which applies to discrete, continuous and hybrid systems. We consider transition systems which enable us to model in a unified framework discrete, continuous and hybrid systems with either deterministic or nondeterministic dynamics (see e.g. [1, 46]).

Definition 1: *A transition system* $T = (X, U, \mathcal{S}, X^0, Y, \mathcal{O})$ *consists of:*

- *a set of states* X*;*
- *a set of inputs* U*;*
- *a transition map* $\mathcal{S} : X \times U \to 2^X$*;*
- *a set of initial states* $X^0 \subseteq X$*;*
- *a set of outputs* Y*;*
- *and an output map* $\mathcal{O} : X \to Y$*.*

T is *metric* if the set of outputs Y is equipped with a metric d. It is sometimes necessary to require some additional technical assumptions on the transition systems (see [22] for details); transition systems satisfying these assumptions are called *regular*. If the set of states X and inputs U are finite or countable, T is said *discrete*. The transition map captures the dynamics of the system. For simplicity, we assume that the systems we consider are *non-blocking*: for all $x \in X$, there exists at least one input $u \in U$ such that $\mathcal{S}(x, u) \neq \emptyset$. If for all states $x \in X$ and all inputs $u \in U$, $\mathcal{S}(x, u)$ contains at most one element, then T is called *deterministic*.

A *state trajectory* of T is an infinite sequence of states and inputs, $\{(x^i, u^i) | \ i \in \mathbb{N}\}$ where $x^0 \in X^0$, $x^{i+1} \in \mathcal{S}(x^i, u^i)$ for all $i \in \mathbb{N}$. The associated *output trajectory* is $\{(y^i, u^i) | \ i \in \mathbb{N}\}$ where $y^i = \mathcal{O}(x^i)$ for all $i \in \mathbb{N}$. The set of output trajectories of T, denoted $\mathcal{L}(T)$, is called the *observed behavior* or the *language* of transition system T.

2.1. Language metrics

In order to compare the observed behaviors of two metric transition systems, we define metrics measuring the distance between their languages and generalizing the notion of language inclusion. For $j \in \{1, 2\}$, let $T_j = (X_j, U, \mathcal{S}_j, X_j^0, Y, \mathcal{O}_j)$ be metric transition systems with the same sets of inputs and outputs, and

$o_j = \{(y_j^i, u_j^i) | i \in \mathbb{N}\} \in \mathcal{L}(T_j)$. The distance between output trajectories o_1 and o_2 is defined as

$$d_\infty(o_1, o_2) = \begin{cases} \sup\limits_{i \in \mathbb{N}} d(y_1^i, y_2^i) & \text{if } \forall i \in \mathbb{N},\ u_1^i = u_2^i; \\ +\infty & \text{otherwise.} \end{cases}$$

Essentially, the distance between o_1 and o_2 is finite if they have same sequence of inputs; in that case the distance between the output trajectories is the maximal distance between the sequences of outputs. Then, we define metrics measuring the distance between the observed behaviors of T_1 and T_2:

Definition 2: *The directed and undirected language metrics are defined respectively as*

$$d_L^{\rightarrow}(T_1, T_2) = \sup_{o_1 \in \mathcal{L}(T_1)} \inf_{o_2 \in \mathcal{L}(T_2)} d_\infty(o_1, o_2)$$

$$d_L(T_1, T_2) = \max \left\{ d_L^{\rightarrow}(T_1, T_2), d_L^{\rightarrow}(T_2, T_1) \right\}.$$

The meaning of the directed language metric is as follows: for any output trajectory of the system T_1, we can find an output trajectory of the system T_2, with the same sequence of inputs, such that the distance between the observations of the two systems remains bounded by $d_L^{\rightarrow}(T_1, T_2)$. In addition if $\mathcal{L}(T_1) \subseteq \mathcal{L}(T_2)$, then $d_L^{\rightarrow}(T_1, T_2) = 0$. One can show that the language metrics are actually directed and undirected pseudo-metrics on the set of metric transition systems.

The language metrics can be particularly useful for reachability analysis. Let us assume that we want to determine if there exists an output trajectory of T_1 that reaches a set $Y_F \subseteq Y$. If we can show that all output trajectories of T_2 remain at a distance of Y_F at least equal to $d_L^{\rightarrow}(T_1, T_2)$, then we can conclude that Y_F cannot be reached by an output trajectory of T_1.

The computation of $d_L^{\rightarrow}(T_1, T_2)$ and $d_L(T_1, T_2)$ is generally extremely difficult. We therefore define a hierarchy of stronger metrics, that are easier to compute and based on approximate versions of the notions of simulation and bisimulation relations.

2.2. Approximate (bi)simulation

The notion of *exact* simulation relation has been introduced in computer science as mean of abstraction of discrete transition systems. Essentially, a simulation relation of T_1 by T_2 is a relation on the states of the systems that describes how to select transitions of T_2 in order to match the transitions of T_1 and to produce the same output trajectory than T_1. The notion of approximate simulation relation is obtained by relaxing the equality of output trajectories: instead of requiring them to be identical, we require that they remain close.

Definition 3: *Let $\varepsilon \geq 0$, a relation $\mathcal{R}_\varepsilon \subseteq X_1 \times X_2$ is called an approximate simulation relation of T_1 by T_2, of precision ε, if for all $(x_1, x_2) \in \mathcal{R}_\varepsilon$:*

1. *$d(\mathcal{O}_1(x_1), \mathcal{O}_2(x_2)) \leq \varepsilon$,*
2. *$\forall u \in U$, $\forall x_1' \in \mathcal{S}_1(x_1, u)$, $\exists x_2' \in \mathcal{S}_2(x_2, u)$ such that $(x_1', x_2') \in \mathcal{R}_\varepsilon$.*

T_2 approximately simulates T_1 with precision ε (denoted $T_1 \preceq_\varepsilon T_2$), if there exists $\mathcal{R}_\varepsilon$, an approximate simulation relation of T_1 by T_2, of precision ε, such that for all $x_1 \in X_1^0$, there exists $x_2 \in X_2^0$ such that $(x_1, x_2) \in \mathcal{R}_\varepsilon$.

For $\varepsilon = 0$, we recover the established definition of exact simulation relation (denoted $T_1 \preceq T_2$). Approximate bisimulation is the symmetric version of approximate simulation and extends the usual notion of *exact* bisimulation relation:

Definition 4: *Let $\varepsilon \geq 0$, a relation $\mathcal{R}_\varepsilon \subseteq X_1 \times X_2$ is called an approximate bisimulation relation between T_1 and T_2, of precision ε, if for all $(x_1, x_2) \in \mathcal{R}_\varepsilon$:*

1. *$d(\mathcal{O}_1(x_1), \mathcal{O}_2(x_2)) \leq \varepsilon$,*
2. *$\forall u \in U$, $\forall x_1' \in \mathcal{S}_1(x_1, u)$, $\exists x_2' \in \mathcal{S}_2(x_2, u)$ such that $(x_1', x_2') \in \mathcal{R}_\varepsilon$.*
3. *$\forall u \in U$, $\forall x_2' \in \mathcal{S}_2(x_2, u)$, $\exists x_1' \in \mathcal{S}_1(x_1, u)$ such that $(x_1', x_2') \in \mathcal{R}_\varepsilon$.*

T_1 and T_2 are approximately bisimilar with precision ε (denoted $T_1 \sim_\varepsilon T_2$), if there exists $\mathcal{R}_\varepsilon$, an approximate bisimulation relation between T_1 and T_2, of precision ε, such that for all $x_1 \in X_1^0$, there exists $x_2 \in X_2^0$ such that $(x_1, x_2) \in \mathcal{R}_\varepsilon$, and conversely.

Again, for $\varepsilon = 0$, we recover the established notion of exact bisimulation relation (denoted $T_1 \sim T_2$). Based on the notions of approximate simulation and bisimulations, we can define metrics that intuitively measures how far two transition systems are from exact simulation or bisimulation.

Definition 5: *The simulation and bisimulation metrics are defined respectively by*

$$d_S^{\rightarrow}(T_1, T_2) = \inf \{\varepsilon | \ T_1 \preceq_\varepsilon T_2\},$$

$$d_B(T_1, T_2) = \inf \{\varepsilon | \ T_1 \sim_\varepsilon T_2\}.$$

One can show that the simulation and bisimulation metrics are respectively directed and undirected pseudo-metrics over the set of metric transition systems. Interestingly, the zero sections of these metrics capture the traditional system relationships.

Theorem 1: *If T_1 and T_2 are regular then*

$$T_1 \preceq T_2 \iff d_S^{\rightarrow}(T_1, T_2) = 0,$$

$$T_1 \sim T_2 \iff d_B(T_1, T_2) = 0.$$

The main result of the section is the following that relates the language, the simulation and bisimulation metrics:

Theorem 2 (Hierarchy of approximation metrics): *For all metric transitions systems T_1 and T_2 with the same sets of inputs and outputs, the following inequalities hold:*

$$\begin{array}{ccc} d_B(T_1,T_2) & \geq & d_L(T_1,T_2) \\ \rotatebox[origin=c]{-90}{$\geq$} & & \rotatebox[origin=c]{-90}{$\geq$} \\ d_S^{\rightarrow}(T_1,T_2) & \geq & d_L^{\rightarrow}(T_1,T_2) \end{array}$$

In addition, if T_1 and T_2 are deterministic, then

$$d_B(T_1,T_2) = d_L(T_1,T_2) \text{ and } d_S^{\rightarrow}(T_1,T_2) = d_L^{\rightarrow}(T_1,T_2).$$

2.3. (Bi)simulation functions

In this section, we focus on the computation of the simulation and bisimulation metrics. In the following, we assume that the metric transition systems T_1 and T_2 we consider are regular. In particular, this implies that the set of states X_1 and X_2 are equipped with metrics.

We present an approach enabling to compute certified upper-bounds of these metrics, based on the fundamental notion of simulation and bisimulation functions defined by Lyapunov like inequalities. A simulation function of T_1 by T_2 is a positive function defined on $X_1 \times X_2$, bounding the distance between the observations associated to the couple (x_1,x_2) and non-increasing under the dynamics of the systems.

Definition 6: *A function $\mathcal{V} : X_1 \times X_2 \to \mathbb{R}^+ \cup \{+\infty\}$ is called a simulation function of T_1 by T_2 if its sub-level sets are closed, and for all $(x_1,x_2) \in X_1 \times X_2$:*

$$\mathcal{V}(x_1,x_2) \geq \max\left\{\begin{array}{l} d(\mathcal{O}_1(x_1),\mathcal{O}_2(x_2)), \\ \displaystyle\sup_{\substack{u\in U\\ x_1'\in\mathcal{S}_1(x_1,u)}} \inf_{x_2'\in\mathcal{S}_2(x_2,u)} \mathcal{V}(x_1',x_2') \end{array}\right\}. \tag{1}$$

The sub-level sets of a simulation function of T_1 by T_2 provide a convenient way to define approximate simulation relations of T_1 by T_2.

Proposition 1: *Let $\mathcal{V}$ be a simulation function of T_1 by T_2. Then, for all $\varepsilon \geq 0$,*

$$\mathcal{R}_\varepsilon = \{(x_1,x_2) \in X_1 \times X_2 |\ \mathcal{V}(x_1,x_2) \leq \varepsilon\}$$

is an approximate simulation relation of T_1 by T_2, of precision ε.

Let us remark that, particularly, the zero set of a simulation function are an exact simulation relation. As a consequence of the previous result, an over-approximation of the simulation metric can be computed using a simulation function.

Theorem 3: *Let $\mathcal{V}$ be a simulation function of T_1 by T_2. Then,*

$$d_S^{\rightarrow}(T_1,T_2) \leq \sup_{x_1\in X_1^0} \inf_{x_2\in X_2^0} \mathcal{V}(x_1,x_2).$$

Actually, it is possible to show that for a particular simulation function, that satisfies a Bellman equation, the upper bound given in the previous theorem is tight.

Theorem 4: *There exists a simulation function of T_1 by T_2, $\mathcal{V}_S^{\min}$ such that for all simulation function of T_1 by T_2, $\mathcal{V}$, for all $(x_1,x_2) \in X_1 \times X_2$, $\mathcal{V}_S^{\min}(x_1,x_2) \leq \mathcal{V}(x_1,x_2)$. This smallest simulation function also satisfies for all $(x_1,x_2) \in X_1 \times X_2$,*

$$\mathcal{V}_S^{\min}(x_1,x_2) = \max\left\{\begin{array}{l} d(\mathcal{O}_1(x_1),\mathcal{O}_2(x_2)), \\ \displaystyle\sup_{\substack{u\in U\\ x_1'\in\mathcal{S}_1(x_1,u)}} \inf_{x_2'\in\mathcal{S}_2(x_2,u)} \mathcal{V}_S^{\min}(x_1',x_2') \end{array}\right\}. \tag{2}$$

Moreover, the simulation metric can be computed by

$$d_S^{\rightarrow}(T_1,T_2) = \sup_{x_1\in X_1^0} \inf_{x_2\in X_2^0} \mathcal{V}_S^{\min}(x_1,x_2).$$

Unfortunately, it is often difficult to solve the Bellman equation (2) and thus to compute exactly the simulation metrics. Therefore, in practice, we shall often use the characterization given by Lyapunov like inequalities (1) and compute upper-bounds of the simulation metrics.

Similarly, the bisimulation metric can be computed or approximated using bisimulation functions, which are essentially symmetric versions of the simulation functions:

Definition 7: *A function $\mathcal{V} : X_1 \times X_2 \to \mathbb{R}^+ \cup \{+\infty\}$ is called a bisimulation function between T_1 and T_2 if its sub-level sets are closed, and for all $(x_1,x_2) \in X_1 \times X_2$:*

$$\mathcal{V}(x_1,x_2) \geq \max\left\{\begin{array}{l} d(\mathcal{O}_1(x_1),\mathcal{O}_2(x_2)), \\ \displaystyle\sup_{\substack{u\in U\\ x_1'\in\mathcal{S}_1(x_1,u)}} \inf_{x_2'\in\mathcal{S}_2(x_2,u)} \mathcal{V}(x_1',x_2'), \\ \displaystyle\sup_{\substack{u\in U\\ x_2'\in\mathcal{S}_2(x_2,u)}} \inf_{x_1'\in\mathcal{S}_1(x_1,u)} \mathcal{V}(x_1',x_2') \end{array}\right\}. \tag{3}$$

Results similar to those of simulation functions can be shown for the case of bisimulation functions. In particular, the sub-level sets of a bisimulation function is an approximate bisimulation relation.

Proposition 2: *Let $\mathcal{V}$ be a bisimulation function between T_1 and T_2. Then, for all $\varepsilon \geq 0$,*

$$\mathcal{R}_\varepsilon = \{(x_1,x_2) \in X_1 \times X_2 | \ \mathcal{V}(x_1,x_2) \leq \varepsilon\}$$

is an approximate bisimulation relation between T_1 and T_2, of precision ε.

Let us remark that the zero set of a bisimulation function is an exact bisimulation relation. An over-approximation of the bisimulation metric can be computed using a bisimulation function.

Theorem 5: *Let $\mathcal{V}$ be a bisimulation function between T_1 and T_2. Then,*

$$d_B(T_1,T_2) \leq \max \left\{ \sup_{x_1 \in X_1^0} \inf_{x_2 \in X_2^0} \mathcal{V}(x_1,x_2), \sup_{x_2 \in X_2^0} \inf_{x_1 \in X_1^0} \mathcal{V}(x_1,x_2) \right\}.$$

Also, the computation of the exact value of the bisimulation metric is possible using a particular bisimulation function.

Theorem 6: *There exists a bisimulation function between T_1 and T_2, $\mathcal{V}_B^{\min}$ such that for all bisimulation function between T_1 and T_2, $\mathcal{V}$, for all $(x_1,x_2) \in X_1 \times X_2$, $\mathcal{V}_B^{\min}(x_1,x_2) \leq \mathcal{V}(x_1,x_2)$. This smallest bisimulation function also satisfies for all $(x_1,x_2) \in X_1 \times X_2$,*

$$\mathcal{V}_B^{\min}(x_1,x_2) = \max \left\{ d(\mathcal{O}_1(x_1),\mathcal{O}_2(x_2)), \sup_{\substack{u \in U \\ x_1' \in \mathcal{S}_1(x_1,u)}} \inf_{x_2' \in \mathcal{S}_2(x_2,u)} \mathcal{V}_B^{\min}(x_1',x_2'), \sup_{\substack{u \in U \\ x_2' \in \mathcal{S}_2(x_2,u)}} \inf_{x_1' \in \mathcal{S}_1(x_1,u)} \mathcal{V}_B^{\min}(x_1',x_2') \right\}. \quad (4)$$

Moreover, the bisimulation metric can be computed by

$$d_B(T_1,T_2) = \max \left\{ \sup_{x_1 \in X_1^0} \inf_{x_2 \in X_2^0} \mathcal{V}_B^{\min}(x_1,x_2), \sup_{x_2 \in X_2^0} \inf_{x_1 \in X_1^0} \mathcal{V}_B^{\min}(x_1,x_2) \right\}.$$

Prior to our work presented above, approximation metrics instead of relationships had been considered in the context of probabilistic transition systems [39, 12] where it is natural to consider approximations of the transition probabilities. Our work is more related to that of [11], where (non-stochastic) finite transition systems are considered; the notion of *branching distance* defined in that work is actually equivalent to the smallest bisimulation function, satisfying equation (4), between a system and itself. Similar ideas were also explored in [49].

There has been several extensions of the approximation framework presented in this section, essentially for relaxing the equality of inputs required in the definition of approximate (bi)simulation. One approach consists in considering metrics on the set of outputs and on the set of inputs [29, 42]; another approach extends the framework by defining the notion of *alternating approximate (bi)simulation* [41].

3. Applications of Approximate (Bi)simulation

In this section, we show several applications of the approximation framework presented in the previous section.

3.1. Bisimulation metrics for linear systems

Before considering more involved applications of our approximation framework, let us consider a first simple example presenting a procedure for the computation of bisimulation functions for deterministic linear systems. For $j \in \{1,2\}$, let

$$\Sigma_j : \begin{cases} \dot{\mathbf{x}}_j(t) &= A_j \mathbf{x}_j(t), \ \mathbf{x}_j(t) \in \mathbb{R}^{n_j}, \ \mathbf{x}_j(0) \in I_j^0, \\ \mathbf{y}_j(t) &= C_j \mathbf{x}_j(t), \ \mathbf{y}_j(t) \in \mathbb{R}^p. \end{cases}$$

We define associated transition systems $T(\Sigma_j) = (X_j, U, \mathcal{S}_j, X_j^0, Y, \mathcal{O}_j)$ where the sets of states are $X_j = \mathbb{R}^{n_j}$, the common set of inputs is $U = \mathbb{R}^+$ and represents time, the transition maps are deterministic and given by

$$x_j' = \mathcal{S}_j(x_j,t) \iff x_j' = e^{tA_j}x_j,$$

the common set of outputs is $Y = \mathbb{R}^p$, the sets of initial states are $X_j^0 = I_j^0$ and the observation maps are given by $\mathcal{O}_j(x_j) = C_j x_j$. The common set of outputs is equipped with the usual Euclidean distance, making $T(\Sigma_j)$ a metric transition system.

We aim at computing the bisimulation metrics between $T(\Sigma_1)$ and $T(\Sigma_2)$. For these simple systems, equation (4) can be shown to be equivalent to a partial differential equation, which is hard to solve, particularly for high dimensional systems. Therefore, we shall compute only an upper bound of the bisimulation metrics using a bisimulation function. Restricting our attention to functions of the form

$$\mathcal{V}(x_1,x_2) = \sqrt{(x_1,x_2)^\top M(x_1,x_2)},$$

equation (3) reduces to the following set of linear matrix inequalities:

$$M \geq C^\top C \text{ and } A^\top M + MA \leq 0$$

where

$$C = [C_1 \quad -C_2] \text{ and } A = \begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix}.$$

The first inequality states that the bisimulation function bounds the distance between outputs while the second one ensures that the function decreases during the evolution of the systems. This set of linear matrix inequalities can be efficiently solved using semi-definite programming, even for high dimensional systems. Then, following theorem 5, the obtained bisimulation function can be used to compute an over-approximation of the bisimulation metric. Let us remark that the linear matrix inequalities are always solvable if both systems are stable. Hence, it is always possible to compute an upper-bound of the bisimulation metric between two stable linear systems.

This approach has been extended in [21] to handle (possibly unstable) linear systems with constrained inputs. For polynomial systems, a similar technique can be used based on sum of squares programs [19]. The computed bisimulation functions between these continuous dynamics can also be used for defining approximate simulation relations for hybrid systems [18]. Similar approaches can also be used for stochastic hybrid systems for which a notion of stochastic bisimulation function is needed [31]. These methods can be used for reducing the complexity of continuous and hybrid systems. Examples of application of our framework to reachability analysis can be found in the mentioned papers.

3.2. Approximately bisimilar discrete abstractions of continuous and switched systems

The use of discrete abstractions for continuous dynamics has become standard in hybrid systems design (see e.g. [1, 46]). The main advantage of this approach is that it offers the possibility to leverage controller synthesis techniques developed in the areas of supervisory control of discrete-event systems or algorithmic game theory. Historically, the first attempts to compute discrete abstractions for continuous or hybrid systems were based on traditional systems behavioral relationships such as simulation or bisimulation. Since these systems are generally observed over metric spaces, the notions of approximate simulation and bisimulation seem much more natural.

We briefly describe the approach presented in [24] for computing discrete abstractions for a class of switched systems:

$$\Sigma : \ \dot{\mathbf{x}}(t) = f_{\mathbf{p}(t)}(\mathbf{x}(t)), \ \mathbf{x}(t) \in \mathbb{R}^n, \ \mathbf{p}(t) \in P$$

where P is a finite set of modes. Given a parameter $\tau > 0$, we define a transition system $T_\tau(\Sigma)$ that describes trajectories of duration τ of Σ. This can be seen as a time sampling process. This is natural when the switching in Σ is determined by a time-triggered controller with period τ.

Formally, $T_\tau(\Sigma) = (X_1, U, \mathcal{S}_1, X_1^0, Y, \mathcal{O}_1)$ where the set of states is $X_1 = \mathbb{R}^n$; the set of inputs is the set of modes $U = P$; the deterministic transition map is given by

$$x_1' = \mathcal{S}_1(x_1,p) \iff x_1' = \mathbf{x}(\tau),$$
$$\text{where } \dot{\mathbf{x}}(t) = f_p(\mathbf{x}(t)), \mathbf{x}(0) = x_1;$$

the set of outputs is $Y = \mathbb{R}^n$; the observation map $\mathcal{O}_1$ is the identity map over $\mathbb{R}^n$; and the set of initial states is $X_1^0 = \mathbb{R}^n$. $T_\tau(\Sigma)$ is metric when the set of observations $Y = \mathbb{R}^n$ is equipped with the Euclidean norm.

The computation of a discrete abstraction of $T_\tau(\Sigma)$ can be done by the following simple approach. We start by approximating the set of states $X_1 = \mathbb{R}^n$ by a lattice:

$$[\mathbb{R}^n]_\eta = \left\{ x \in \mathbb{R}^n \ \middle| \ x_i = k_i \frac{2\eta}{\sqrt{n}}, \ k_i \in \mathbb{Z}, \ i = 1, ..., n \right\},$$

where x_i is the i-th coordinate of x and $\eta > 0$ is a state space discretization parameter.

We can then define the abstraction of $T_\tau(\Sigma)$ as the transition system $T_{\tau,\eta}(\Sigma) = (X_2, U, \mathcal{S}_2, X_2^0, Y, \mathcal{O}_2)$, where the set of states is $X_2 = [\mathbb{R}^n]_\eta$; the set of labels remains the same $U = P$; the transition relation is essentially obtained by rounding the transition relation of $T_\tau(\Sigma)$ on the lattice $[\mathbb{R}^n]_\eta$ (see Fig. 1):

$$x_2' = \mathcal{S}_2(x_2,p) \iff x_2' = \arg \min_{x' \in [\mathbb{R}^n]_\eta} \|x' - \mathcal{S}_1(x_2,p)\|;$$

the set of outputs remains the same $Y = \mathbb{R}^n$; the observation map $\mathcal{O}_2$ is given by $\mathcal{O}_2(q) = q$, the set of initial states is $X_2^0 = [\mathbb{R}^n]_\eta$. Note that the transition system $T_{\tau,\eta}(\Sigma)$ is discrete since its sets of states and actions are respectively countable and finite.

The approximate bisimilarity of $T_\tau(\Sigma)$ and $T_{\tau,\eta}(\Sigma)$ is related to the notion of *incremental stability* [3]. Intuitively, incremental global uniform asymptotic stability (δ-GUAS) of a switched system means that all the trajectories associated with the same switching signal converge to the same reference trajectory independently of their initial

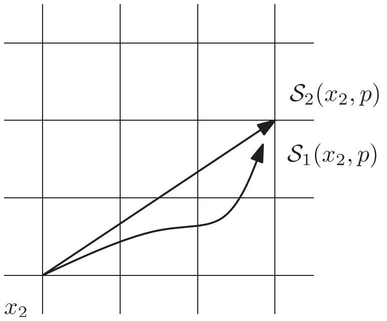

Fig. 1. Principle for the computation of a discrete abstraction of a switched system.

condition. Incremental stability of a switched system can be characterized using Lyapunov functions.

Definition 8: *A smooth function $\mathcal{V} : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^+$ is a common δ-GUAS Lyapunov function for Σ if there exist $\mathcal{K}_\infty$ functions[1] $\underline{\alpha}$, $\overline{\alpha}$ and $\kappa > 0$ such that for all $x, y \in \mathbb{R}^n$, for all $p \in P$:*

$$\underline{\alpha}(\|x-y\|) \leq \mathcal{V}(x,y) \leq \overline{\alpha}(\|x-y\|);$$
$$\frac{\partial \mathcal{V}}{\partial x}(x,y) \cdot f_p(x) + \frac{\partial \mathcal{V}}{\partial y}(x,y) \cdot f_p(y) \leq -\kappa \mathcal{V}(x,y).$$

The existence of a common δ-GUAS Lyapunov function ensures that the switched system Σ is incrementally stable. We need to make the supplementary assumption on the δ-GUAS Lyapunov function that there exists a $\mathcal{K}_\infty$ function γ such that

$$\forall x,y,z \in \mathbb{R}^n,\ |\mathcal{V}(x,y) - \mathcal{V}(x,z)| \leq \gamma(\|y-z\|). \quad (5)$$

This assumption is not restrictive provided $\mathcal{V}$ is smooth and we are interested in the dynamics of Σ on a compact subset of $\mathbb{R}^n$, which is often the case in practice.

Theorem 7: *Consider a switched system Σ, time and state space sampling parameters $\tau, \eta > 0$ and a desired precision $\varepsilon > 0$. If there exists a common δ-GUAS Lyapunov function $\mathcal{V}$ for Σ such that equation (5) holds and*

$$\eta \leq \min\left\{\gamma^{-1}\left((1-e^{-\kappa\tau})\underline{\alpha}(\varepsilon)\right), \overline{\alpha}^{-1}\left(\underline{\alpha}(\varepsilon)\right)\right\} \quad (6)$$

then

$$\mathcal{R}_\varepsilon = \left\{(x_1,x_2) \in X_1 \times X_2 |\ \mathcal{V}(x_1,x_2) \leq \underline{\alpha}(\varepsilon)\right\}$$

is an ε-approximate bisimulation relation between $T_\tau(\Sigma)$ and $T_{\tau,\eta}(\Sigma)$. Moreover, $T_\tau(\Sigma) \sim_\varepsilon T_{\tau,\eta}(\Sigma)$.

Let us remark that the δ-GUAS Lyapunov function $\mathcal{V}$ essentially plays the role of bisimulation function here. Particularly, it should be noted that given a time sampling parameter $\tau > 0$ and a desired precision $\varepsilon > 0$, it is always possible to choose $\eta > 0$ such that equation (6) holds. This essentially means that approximately bisimilar discrete abstractions of arbitrary precision can be computed for $T_\tau(\Sigma)$.

For simplicity, we only presented results for switched systems that have a common δ-GUAS Lyapunov function. However, let us remark that it is possible to extend these results for systems with multiple Lyapunov functions by imposing a dwell time (see [24] for more details). The approach described above has been used in [16, 17] for synthesizing safety and reachability switching controllers for a model of a DC-DC converter (see Fig. 2). Similar approaches can be used to compute approximately bisimilar abstractions for incrementally stable nonlinear systems [38] and time-delay nonlinear systems [40]. An approach relaxing the incremental stability assumptions for non-linear systems can be found [52]; this approach is made currently available in the tool PESSOA [43]. The development of discrete abstractions defined on non-uniform grids has been treated in [48, 6]. Finally, an approach for computing discrete abstractions of linear systems without griding the state-space can be found in [15].

3.3. Hierarchical control design using simulation functions

Controlling complex (e.g. 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 intractable problems and therefore a hierarchical approach to control synthesis is highly desirable. In [23], a hierarchical control framework based on the notion of approximate simulation was introduced. We present the main ideas in the following.

Let us consider the control system given, for $j \in \{1,2\}$ by:

$$\Sigma_j : \left\{ \begin{array}{lcl} \dot{\mathbf{x}}_j(t) & = & f_j(\mathbf{x}_j(t), \mathbf{u}_j(t)),\ \mathbf{x}_j(t) \in \mathbb{R}^{n_j}\ \mathbf{u}_j(t) \in \mathbb{R}^{p_j} \\ \mathbf{y}_j(t) & = & g_j(\mathbf{x}_j(t)),\ \mathbf{y}_j(t) \in \mathbb{R}^k. \end{array} \right.$$

We refer to Σ_1 as the *concrete system*, that is the (complex) system that we actually want to control. Control is synthesized hierarchically, using the *abstract system* Σ_2, giving a simpler, though less precise, description of the dynamics of the system. Note that systems Σ_1 and Σ_2 have the same output space (i.e. $\mathbb{R}^k$), but may have different state and input spaces. In particular, the fact that we have different input spaces differs from the work presented in the previous sections. Since we have different input spaces, we cannot ask for equality of inputs as previously.

The proposed hierarchical control approach allows to refine inputs designed for the abstract system Σ_2 in order

[1] A continuous function $\gamma : \mathbb{R}^+ \to \mathbb{R}^+$ is said to belong to class $\mathcal{K}_\infty$ if it is strictly increasing, $\gamma(0) = 0$ and $\gamma(r) \to \infty$ when $r \to \infty$.

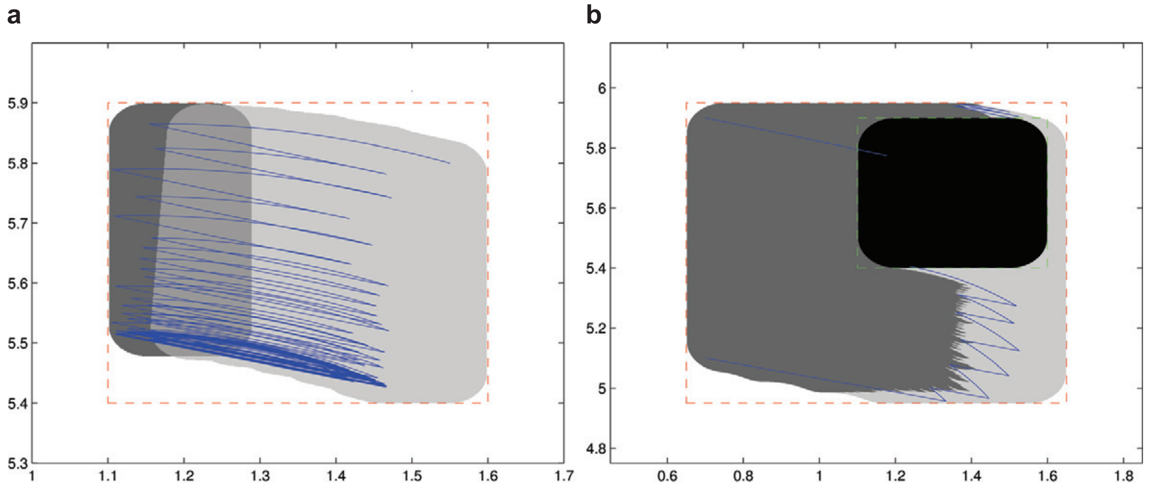

Fig. 2. Example of switching safety (left) and reachability (right) controllers synthesized using approximately bisimilar discrete abstractions of a model of a DC-DC converter [16, 17].

to control the concrete system Σ_1. We adapt the definition of simulation function given previously to the specific case of continuous control systems. Essentially, a simulation function is a function bounding the distance between the outputs of Σ_1 and Σ_2 and decreasing during the evolution of systems. The different inputs are taken care of using the notion of *interface*.

Definition 9: *Let $\mathcal{V} : \mathbb{R}^{n_2} \times \mathbb{R}^{n_1} \to \mathbb{R}^+$ be a smooth function and $u_{\mathcal{V}} : \mathbb{R}^{p_2} \times \mathbb{R}^{n_2} \times \mathbb{R}^{n_1} \to \mathbb{R}^{p_1}$ be a continuous function. $\mathcal{V}$ is a simulation function of Σ_2 by Σ_1 and $u_{\mathcal{V}}$ is an associated interface if there exists a $\mathcal{K}$ function[2] γ such that for all $(x_2, x_1) \in \mathbb{R}^{n_2} \times \mathbb{R}^{n_1}$,*

$$\mathcal{V}(x_2, x_1) \geq \|g_1(x_1) - g_2(x_2)\| \tag{7}$$

and for all $u_2 \in \mathbb{R}^{p_2}$, satisfying $\gamma(\|u_2\|) < \mathcal{V}(x_2, x_1)$,

$$\frac{\partial \mathcal{V}(x_2, x_1)}{\partial x_2} \cdot f_2(x_2, u_2) + \frac{\partial \mathcal{V}(x_2, x_1)}{\partial x_1} \cdot f_1(x_1, u_{\mathcal{V}}(u_2, x_2, x_1)) < 0 \tag{8}$$

It is interesting to note that the notion of interface can be dropped if one adopts the formalism of alternating approximate simulation [41]. As previously stated, a simulation function allows us to bound the distance between outputs of Σ_2 and Σ_1:

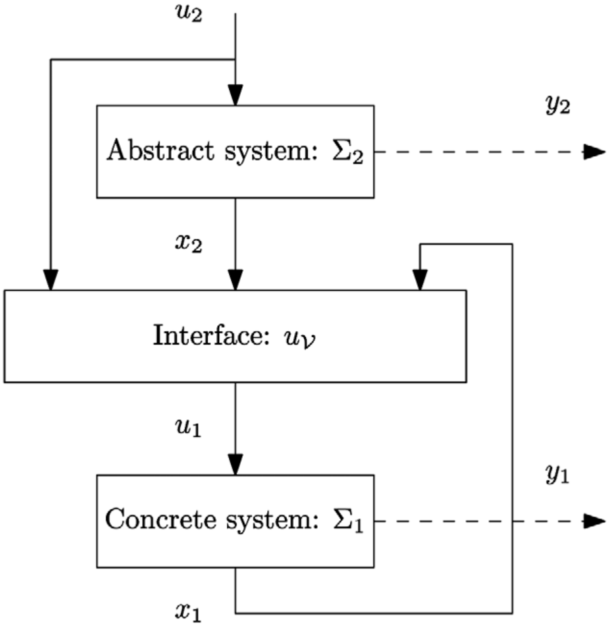
u_2
y_2
Abstract system: Σ_2
x_2
Interface: $u_{\mathcal{V}}$
u_1
y_1
Concrete system: Σ_1
x_1

Fig. 3. Hierarchical control system architecture.

Theorem 8: *Let $\mathcal{V}$ be a simulation function of Σ_2 by Σ_1 and $u_{\mathcal{V}}$ an associated interface. Let $\mathbf{u}_2 : \mathcal{I} :\to \mathbb{R}^{p_2}$ be an input of Σ_2, let $\mathbf{x_2}$ and $\mathbf{y_2}$ be the state and output trajectories of Σ_2. Let $\mathbf{x_1}$ be a state trajectory of Σ_1 satisfying the differential equation*

$$\dot{\mathbf{x}}_1(t) = f\left(\mathbf{x}_1(t), u_{\mathcal{V}}(\mathbf{u}_2(t), \mathbf{x}_2(t), \mathbf{x}_1(t))\right) \tag{9}$$

and let $\mathbf{y_1}$ be the associated output trajectory. Then, for all $t \in \mathcal{I}$,

$$\|\mathbf{y}_2(t) - \mathbf{y}_1(t)\| \leq \max\left\{\mathcal{V}(\mathbf{x}_2(0), \mathbf{x}_1(0)), \gamma(\|\mathbf{u}_2\|_\infty)\right\}.$$

The control architecture, allowing us to refine the inputs the abstract system through the interface $u_{\mathcal{V}}$ is shown on Fig. 3.

The applicability of our approach relies on our capability of computing a simulation function and an associated

[2] A function $\gamma : \mathbb{R}^+ \to \mathbb{R}^+$ is a $\mathcal{K}$ function if it is continuous, strictly increasing and satisfies $\gamma(0) = 0$.

interface. In [23], an effective characterization of simulation functions for linear control systems is given allowing us to design algorithmic procedures for their computation. This hierarchical approach has been used to solve control problems with complex temporal logic specifications in [13]. An approach with similar flavors has been proposed in [8, 9] for hierarchical stabilization and tracking control. The hierarchical control methods presented in [45, 47], based on the use of discrete abstractions, are also quite similar to the one presented here.

3.4. Verification using simulation

Verification consists in analyzing the behavior of a dynamical system Σ against some specification: given a property Φ defined on trajectories (e.g. "a trajectory reaches the set Y_F"), we would like to be able to prove that it is satisfied by all trajectories of the system. For discrete systems, the problem has attracted a lot of attention in computer science, resulting in the success story of Model Checking [7] with the associated set of techniques widely used in the industry. Verification of continuous and hybrid systems is in general much more challenging since these generally have an uncountable number of trajectories. There has also been a lot of work on the subject using set-based reachability computations, abstraction and or deductive techniques (see [2] for a recent survey).

In the following, we briefly describe an approach based on simulation of individual trajectories of a system together with the use of bisimulation function [14]. Let us consider the following dynamical system:

$$\Sigma : \left\{ \begin{array}{lll} \dot{\mathbf{x}}(t) & = & f(\mathbf{x}(t)),\ \mathbf{x}(t) \in \mathbb{R}^n,\ \mathbf{x}(0) \in I^0 \\ \mathbf{y}(t) & = & g(\mathbf{x}(t)),\ \mathbf{y}(t) \in \mathbb{R}^k. \end{array} \right.$$

Let be given a property Φ defined on the output trajectories $\mathbf{y}$, formulated for instance in some temporal logic; $\mathbf{y} \models \Phi$ means that trajectory $\mathbf{y}$ satisfies property Φ. Let us assume that we are given a measure $\rho(\mathbf{y}, \Phi)$ estimating how robustly property Φ is satisfied by Φ:

$$\big(\forall t \geq 0,\ \|\mathbf{y} - \mathbf{y}'\| \leq \rho(\mathbf{y}, \Phi)\big) \implies \mathbf{y}' \models \Phi.$$

We refer the reader to [14] for a precise definition of such a measure and algorithms for its computation. The last ingredient of the verification approach is a bisimulation function between Σ and itself:

Definition 10: *Let $\mathcal{V} : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^+$ be a smooth function, $\mathcal{V}$ is an (auto)bisimulation function of Σ if for all $(x_1, x_2) \in \mathbb{R}^n \times \mathbb{R}^n$,*

$$\mathcal{V}(x_1, x_2) \geq \|g(x_2) - g(x_1)\|$$

$$\frac{\partial \mathcal{V}(x_1, x_2)}{\partial x_1} \cdot f(x_1) + \frac{\partial \mathcal{V}(x_1, x_2)}{\partial x_2} \cdot f(x_2) \leq 0$$

Using robustness measures for property satisfaction and a bisimulation function makes it possible to verify that the property holds for an infinite number of trajectories by simulating only one trajectory:

Theorem 9: *Let $x^0 \in I^0$ be an initial condition of Σ and let $\mathbf{y}$ be the associated output trajectory. Then, for all $x'^0 \in I^0$, with associated output trajectories $\mathbf{y}'$*

$$\Big(\mathcal{V}(x^0, x'^0) \leq \rho(\mathbf{y}, \Phi)\Big) \implies \mathbf{y}' \models \Phi$$

The previous result allows us to verify that the property Φ holds for all the trajectories of Σ by computing only a finite number of them. Let $\{x_1^0, \ldots, x_n^0\} \subseteq I^0$ be a finite subset of initial conditions and $\{\mathbf{y}_1, \ldots, \mathbf{y}_n\}$ the associated output trajectories of Σ such that for all $x^0 \in I^0$, there exists x_i^0 such that $\mathcal{V}(x^0, x_i^0) \leq \rho(\mathbf{y}_i, \Phi)$ then all the trajectories of Σ satisfy Φ. An algorithm to construct iteratively the sample of initial states can be found in [14]. In the case, we cannot cover the whole set of initial states, the algorithm identifies a subset of initial states for which the property holds.

Similar approaches have been developed for the verification of non-deterministic metric transition systems [20, 25] or hybrid systems [30, 34]. The same kind of ideas can be used for controller synthesis by defining control laws for a finite number of trajectories [28].

4. Conclusion

In this paper we have reviewed an approximation theory that bridges established notions in computer science and control theory. We believe that such frameworks will enable the transfer of intellectual ideas from control to computer science. In particular notions of approximation and robustness, so critical to the success of continuous control, could have a transformative effect in converting boolean computing from a qualitative science to a quantitative science [26]. Likewise, notions of robust computing are also beginning to emerge in computer science [36, 10]. Our hope is that in the next fifty years, we will see a stronger convergence between the disciplines of control and computing.

References

1. Alur R, Henzinger TA, Lafferriere G, Pappas GJ. Discrete abstractions of hybrid systems. *Proceedings IEEE 2000*; 88(7): 971–984.
2. Alur R. Formal verification of hybrid systems. In 11th International Conference on Embedded Software 2011.
3. Angeli D. A Lyapunov approach to incremental stability properties. *IEEE Trans Autom Control* 2002; 47(3): 410–421.

4. Antoulas AC, Sorensen DC, Gugercin S. A survey of model reduction methods for large-scale systems. *Contemp Math* 2000; 280: 193–219.
5. Caspi P, Benveniste A. Toward an approximation theory for computerised control. In Alberto Sangiovanni-Vincentelli and Joseph Sifakis, editors, Embedded Software, volume 2491 of Lecture Notes in Computer Science, pages 294–304. Springer Berlin / Heidelberg, 2002.
6. Camara J, Girard A, Goessler G. Synthesis of switching controllers using approximately bisimilar multiscale abstractions. *In Hybrid Systems*: *Comput Control* 2011.
7. Clarke EM, Grumberg O, Peled DA. Model Checking. MIT Press, 2000.
8. Cavarischia L, Lanari L. Hierarchical control implementation. In *Mediterranean Control Conference* 2007.
9. Cavarischia L, Lanari L. Hierarchical tracking implementation. In *IEEE Conference Decision Control* 2007.
10. Chaudhuri S, Lezama AS. Smooth interpretation. In Proceedings of the 2010 ACM SIGPLAN conference on Programming language design and implementation, PLDI '10, 279–291, New York, NY, USA, 2010. ACM.
11. de Alfaro L, Faella M, Stoelinga M. Linear and branching metrics for quantitative transition systems. In International Colloquium on Automata, Languages and Programming, volume 3142 of LNCS, pages 1150–1162. Springer, 2004.
12. Desharnais J, Gupta V, Jagadeesan R, Panangaden P. Metrics for labeled Markov processes. *Theor Comput Sci* 2004; 318(3): 323–354.
13. Fainekos GE, Girard A, Kress-Gazit H, Pappas GJ. Temporal logic planning for dynamic models. *Automatica* 2009; 45(2): 343–352.
14. Fainekos GE, Girard A, Pappas GJ. Temporal logic verification using simulation. In *Formal Modeling and Analysis of Timed Systems* 2006; 4202 of LNCS: 171–186. Springer.
15. Girard A. Approximately bisimilar finite abstractions of stable linear systems. In *Hybrid Systems: Comput and Control* 2007; 4416 of LNCS: 231–244. Springer.
16. Girard A. Synthesis using approximately bisimilar abstractions: State-feedback controllers for safety specifications. In *Hybrid Systems: Comput and Control* 2010; 111–120.
17. Girard A. Synthesis using approximately bisimilar abstractions: Time-optimal control problems. In *IEEE Conf Decision Control* 2010.
18. Girard A, Julius AA, Pappas GJ. Approximate simulation relations for hybrid systems. *Discret Event Dyn Syst* 2008; 18(2): 163–179.
19. Girard A, Pappas GL. Approximate bisimulations for nonlinear dynamical systems. *In IEEE Conference Decision Control European Control Conf* 2005; 684–689.
20. Girard A, Pappas GJ. Verification using simulation. In *Hybrid Systems: Comput Control* 2006; 3927 of LNCS: 272–286. Springer.
21. Girard A, Pappas GJ. Approximate bisimulation relations for constrained linear systems. *Automatica* 2007; 43(8): 1307–1317.
22. Girard A, Pappas GJ. Approximation metrics for discrete and continuous systems. *IEEE Trans Autom Control* 2007; 52(5): 782–798.
23. Girard A, Pappas GJ. Hierarchical control system design using approximate simulation. *Automatica*; 45(2): 566–571.
24. Girard A, Pola G, Tabuada P. Approximately bisimilar symbolic models for incrementally stable switched systems. *IEEE Trans Autom Control* 2010; 55(1): 116–126.
25. Girard A, Zheng G. Verification of safety and liveness properties of metric transition systems. ACM Transactions on Embedded Computing Systems, 2011. To appear. [Hen07] Thomas A. Henzinger. Quantitative generalizations of languages. In Proceedings of the 11th international conference on Developments in language theory, DLT'07; 20–22, Berlin, Heidelberg, 2007. Springer-Verlag.
26. Thomas A. Henzinger. Quantitative generalizations of languages. In Proceedings of the 11th international conference on Developments in language theory, DLT'07; 20–22, Berlin, Heidelberg, 2007. Springer-Verlag.
27. Haghverdi E, Tabuada P, Pappas GJ. Bisimulation relations for dynamical, control, and hybrid systems. *Theor Comput Sci* 2005; 342(2-3): 229–261.
28. Julius AA, Afshari S. Using computer games for hybrid systems controller synthesis. In *IEEE Conf Decision Control* 2010.
29. Julius AA, D'Innocenzo A, di Benedetto MD, Pappas GJ. Approximate equivalence and synchronization of metric transition systems. *Syst Control Lett* 2009; 58: 94–101.
30. Julius AA, Fainekos G, Anand M, Lee I, Pappas GJ. Robust test generation and coverage for hybrid systems. In *Hybrid Systems: Comput Control* 2007; 4416 of LNCS: 329–342. Springer.
31. Julius AA, Pappas GJ. Approximate abstraction of stochastic hybrid systems. *IEEE Trans Autom Control* 2009; 54(6): 1193–1203.
32. Kalman RE, Falb PL, Arbib MA. Topics in Mathematical System Theory. McGraw-Hill, 1969.
33. Liberzon D. Switching in systems and control. Birkhauser, 2003.
34. Lerda F, Kapinski J, Clarke EM, Krogh BH. Verification of supervisory control software using state proximity and merging. In Hybrid Systems: Computation and Control 2008; 4981 of LNCS: 344–357. Springer.
35. Milner R. Communication and Concurrency. Prentice-Hall, 1989.
36. Majumdar R, Render E, Tabuada P. A theory of robust software synthesis. CoRR, abs/1108.3540, 2011.
37. Pappas GJ. Bisimilar linear systems. *Automatica* 2003; 39(12): 2035–2047.
38. Pola G, Girard A, Tabuada P. Approximately bisimilar symbolic models for nonlinear control systems. *Automatica* 2008; 44(10): 2508–2516.
39. Di Pierro A, Hankin C, Wiklicky H. Quantitative relations and approximate process equivalences. In *Conference Concurrency Theory* 2003; 2761 of LNCS: 508–522. Springer, 2003.
40. Pola G, Pepe P, di Benedetto MD, Tabuada P. Symbolic models for nonlinear time-delay systems using approximate bisimulations. *Syst Control Lett* 2010; 59: 365–373.
41. Pola G, Tabuada P. Symbolic models for nonlinear control systems: Alternating approximate bisimulations. *SIAM J Control Optim* 2009; 48(2): 719–733.
42. Quesel JD, Frnzle M, Damm W. Crossing the bridge between similar games. In *Formal Modeling Analysis Timed Systems* 2011; 6919 of LNCS: 160–176. Springer.
43. Roy P, Tabuada P, Majumdar R. Pessoa 2.0: A controller synthesis tool for cyber-physical systems. In *Hybrid Systems: Comput Control* 2011.
44. Ramadge PJG, Wonham WM. The control of discrete event systems. *Proceedings IEEE* 1989; 77(1): 81–98.
45. Tabuada P. An approximate simulation approach to symbolic control. *IEEE Trans Autom Control* 2008; 53(6): 1406–1418.
46. Tabuada P. Verification and Control of Hybrid Systems-A Symbolic Approach. Springer 2009.

47. Tazaki Y, Imura J. Finite abstractions of discrete-time linear systems and its application to optimal control. In IFAC World Congress 2008.
48. Tazaki Y, J. Imura J. Discrete-state abstractions of nonlinear systems using multi-resolution quantizer. In *Hybrid Systems: Comput Control* 2009; 5469 of LNCS: 351–365. Springer.
49. van Breugel F. A behavioural pseudometric for metric labelled transition systems. In *Conference Concurrency Theory* 2005; 3653 of LNCS: 141–155. Springer.
50. van Breugel F, Mislove M, Ouaknine J, Worrell J. An intrinsic characterization of approximate probabilistic bisimilarity. In Proceedings of the 6th International conference on Foundations of Software Science and Computation Structures and joint European conference on Theory and practice of software, FOSSACS'03/ETAPS'03, 2003; pages 200–215, Berlin, Heidelberg. Springer-Verlag.
51. van der Schaft A. Equivalence of dynamical systems by bisimulation. *IEEE Trans Autom Control* 2004; 49(12): 2160–2172.
52. Zamani M, Pola G, Mazo M, Tabuada P. Symbolic models for nonlinear control systems without stability assumptions. 2010; arXiv:1002.0822, submitted for publication.