Approximate Bisimulations for Nonlinear Dynamical Systems

Antoine Girard and George J. Pappas

Abstract— **The notion of exact bisimulation equivalence for nondeterministic discrete systems has recently resulted in notions of exact bisimulation equivalence for continuous and hybrid systems. In this paper, we establish the more robust notion of approximate bisimulation equivalence for nondeterministic nonlinear systems. This is achieved by requiring that a distance between system observations starts and remains, close, in the presence of nondeterministic system evolution. We show that approximate bisimulation relations can be characterized using a class of functions called bisimulation functions. For nondeterministic nonlinear systems, we show that conditions for the existence of bisimulation functions can be expressed in terms of Lyapunov-like inequalities, which for deterministic systems can be computed using recent sum-of-squares techniques. Our framework is illustrated on a safety verification example.**

I. Introduction

Compositional modeling and hierarchical verification of discrete systems have resulted in established notions of system refinement and equivalence, such as language inclusion, simulation and bisimulation relations [2]. Much more recently, simulation and bisimulation relations have resulted in new equivalence notions for nondeterministic continuous and hybrid systems [9], [15], [20], [22].

The established notions for both discrete and continuous systems are all *exact*, requiring external behavior of two systems to be identical. When interacting with the physical world, exact equivalence notions are restrictive and not robust. Notions of system approximation seem much more appropriate in this context, and this has recently resulted in approximate bisimulation relations for stochastic [4], quantitative [3], and metric transition systems [7], [8].

In [8], we developed a framework for (discrete and continuous) system approximation for general metric transition systems. The approximation process is based on a metric on the set of observations. Rather than requiring that the distance between system observations is (and remains) zero, we require that the distance between observations is (and remains) arbitrarily close. We showed that approximate bisimulations can be characterized by a set of functions called bisimulation functions. In [7], a method to compute bisimulations functions is proposed for the class metric transition systems generated by linear systems with constrained inputs.

In this paper, we extend our work to the class of nonlinear dynamical systems[1]. We develop Lyapunov-like conditions for bisimulation functions, which are used to over-approximate the observational distance between two nonlinear systems, in the presence of nondeterminism (disturbances). For the class of deterministic systems, we show how to compute such functions using sum of squares decomposition techniques [14], [16], [17], [18].

Our results are a natural generalization of the exact notions of bisimulation of nonlinear dynamical systems [9], [20], [22]. Furthermore, our approach enables the approximate but safe reduction of verification problems for nonlinear systems [1], [13], to verification problems for linear systems where efficient and scalable methods are emerging recently [6], [10], [23]. Note that compared to model reduction techniques for nonlinear systems, which are generally based on sets of numerical simulation or experimental data [11], our approach is only based on the model and does not require simulations.

II. Approximation of Transition Systems

In this section, we summarize the notion of approximate bisimulation of labeled transition systems as developed in [8]. Labeled transition systems can be seen as graphs, possibly with an infinite number of states or transitions.

Definition 2.1: A labeled transition system with observations is a tuple $T = (\mathcal{Q}, \Sigma, \rightarrow, \mathcal{Q}^0, \Pi, \langle\langle . \rangle\rangle)$ that consists of:

- a (possibly infinite) set $\mathcal{Q}$ of states,
- a (possibly infinite) set Σ of labels,
- a transition relation $\rightarrow \subseteq \mathcal{Q} \times \Sigma \times \mathcal{Q}$,
- a (possibly infinite) set $\mathcal{Q}^0 \subseteq \mathcal{Q}$ of initial states,
- a (possibly infinite) set Π of observations, and
- an observation map $\langle\langle . \rangle\rangle : \mathcal{Q} \rightarrow \Pi$.

The transition $(q, \sigma, q') \in \rightarrow$ is denoted $q \xrightarrow{\sigma} q'$. For all labels $\sigma \in \Sigma$, the σ-successor is defined as the set valued map given by

$$\forall q \in Q,\ \mathrm{Post}^\sigma(q) = \left\{ q' \in Q |\ q \xrightarrow{\sigma} q' \right\}.$$

We assume that the systems we consider are non-blocking. Then, a state trajectory of T is an infinite sequence of transitions,

$$q^0 \xrightarrow{\sigma^0} q^1 \xrightarrow{\sigma^1} q^2 \xrightarrow{\sigma^2} \dots,\ \text{where } q^0 \in \mathcal{Q}^0.$$

The associated external trajectory $\pi^0 \xrightarrow{\sigma^0} \pi^1 \xrightarrow{\sigma^1} \pi^2 \xrightarrow{\sigma^2} \dots$ (where $\pi^i = \langle\langle q^i \rangle\rangle$ for all $i \in \mathbb{N}$) describes the evolution of the observations under the dynamics of the labeled transition system. The set of external trajectories of the labeled transition system T is called the language of T.

This research is partially supported by the Région Rhône-Alpes (Projet CalCel) and the NSF Presidential Early CAREER (PECASE) Grant 0132716.

Antoine Girard and George J. Pappas are with the Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA 19104 {agirard,pappasg}@seas.upenn.edu

[1] Though we consider only continuous-time dynamics, most of the results also hold for discrete-time systems.

A. Approximate Bisimulations

Exact bisimulation between two labeled transition systems requires that their observations are (and remain) identical [2]. Approximate bisimulation is less rigid since it only requires that the observations of both systems are (and remain) arbitrarily close. Let $T_1 = (\mathcal{Q}_1, \Sigma_1, \to_1, \mathcal{Q}_1^0, \Pi_1, \langle\langle . \rangle\rangle_1)$ and $T_2 = (\mathcal{Q}_2, \Sigma_2, \to_2, \mathcal{Q}_2^0, \Pi_2, \langle\langle . \rangle\rangle_2)$ be two labeled transition systems with the same set of labels ($\Sigma_1 = \Sigma_2 = \Sigma$) and the same set of observations ($\Pi_1 = \Pi_2 = \Pi$). Let us assume that the sets of states Q_1, Q_2 and the set of observations Π are metric spaces. We assume that the initial sets Q_1^0 and Q_2^0 as well as the sets $\text{Post}_1^\sigma(q_1)$ and $\text{Post}_2^\sigma(q_2)$ (for all $\sigma \in \Sigma$, $q_1 \in Q_1$, $q_2 \in Q_2$) are compact sets. Let us note d_Π the metric on Π.

Definition 2.2: A relation $\mathcal{B}_\delta \subseteq \mathcal{Q}_1 \times \mathcal{Q}_2$ is a δ-approximate bisimulation between T_1 and T_2 if for all $(q_1, q_2) \in \mathcal{B}_\delta$:

1) $d_\Pi(\langle\langle q_1 \rangle\rangle_1, \langle\langle q_2 \rangle\rangle_2) \leq \delta$,
2) $\forall q_1 \xrightarrow{\sigma}_1 q_1'$, $\exists q_2 \xrightarrow{\sigma}_2 q_2'$ such that $(q_1', q_2') \in \mathcal{B}_\delta$,
3) $\forall q_2 \xrightarrow{\sigma}_2 q_2'$, $\exists q_1 \xrightarrow{\sigma}_1 q_1'$ such that $(q_1', q_2') \in \mathcal{B}_\delta$.

Note that for $\delta = 0$, we have the usual notion of exact bisimulation [2].

Definition 2.3: T_1 and T_2 are said to be approximately bisimilar with the precision δ (noted $T_1 \sim_\delta T_2$), if there exists $\mathcal{B}_\delta$, a δ-approximate bisimulation between T_1 and T_2 such that for all $q_1 \in \mathcal{Q}_1^0$, there exists $q_2 \in \mathcal{Q}_2^0$ such that $(q_1, q_2) \in \mathcal{B}_\delta$, and conversely.

The approximate bisimilarity of two systems guarantees that the distance between their language is bounded.

Theorem 2.4: [8] If T_1 and T_2 are approximately bisimilar with the precision δ then for all external trajectory of T_1 (respectively T_2), $\pi_1^0 \xrightarrow{\sigma^0} \pi_1^1 \xrightarrow{\sigma^1} \pi_1^2 \xrightarrow{\sigma^2} \dots$, there exists an external trajectory of T_2 (respectively T_1) with the same sequence of labels $\pi_2^0 \xrightarrow{\sigma^0} \pi_2^1 \xrightarrow{\sigma^1} \pi_2^2 \xrightarrow{\sigma^2} \dots$ such that for all $i \in \mathbb{N}$, $d_\Pi(\pi_1^i, \pi_2^i) \leq \delta$.

B. Bisimulation Functions

The construction of approximate bisimulations as well as the evaluation of their precision can be performed using a class of functions called bisimulation functions. Essentially, bisimulation functions are positive functions defined on $\mathcal{Q}_1 \times \mathcal{Q}_2$, bounding the distance between the observations associated to a couple (q_1, q_2) and non increasing under the (nondeterministic) dynamics of the systems.

Definition 2.5: A continuous function $V_\mathcal{B} : \mathcal{Q}_1 \times \mathcal{Q}_2 \to \mathbb{R}^+$ is a bisimulation function between T_1 and T_2 if for all $(q_1, q_2) \in \mathcal{Q}_1 \times \mathcal{Q}_2$:

1) $V_\mathcal{B}(q_1, q_2) \geq d_\Pi(\langle\langle q_1 \rangle\rangle_1, \langle\langle q_2 \rangle\rangle_2)$,
2) $V_\mathcal{B}(q_1, q_2) \geq \max_{q_1 \xrightarrow{\sigma}_1 q_1'} \min_{q_2 \xrightarrow{\sigma}_2 q_2'} V_\mathcal{B}(q_1', q_2')$,
3) $V_\mathcal{B}(q_1, q_2) \geq \max_{q_2 \xrightarrow{\sigma}_2 q_2'} \min_{q_1 \xrightarrow{\sigma}_1 q_1'} V_\mathcal{B}(q_1', q_2')$.

The level sets of a bisimulation functions define approximate bisimulation relations.

Theorem 2.6: [8] Let $V_\mathcal{B}$ be a bisimulation function. Then, for all $\delta \geq 0$, the set

$$\mathcal{B}_\delta = \{(q_1, q_2) \in \mathcal{Q}_1 \times \mathcal{Q}_2, V_\mathcal{B}(q_1, q_2) \leq \delta\}$$

is a δ-approximate bisimulation between T_1 and T_2.

Let us remark that particularly, the zero set of a bisimulation function is an exact bisimulation between T_1 and T_2. The following corollary is straightforward from Theorem 2.6 and Definition 2.3.

Corollary 2.7: [8] Let $V_\mathcal{B}$ be a bisimulation function. Let δ be the value of the following game:

$$\delta = \max\left(\max_{q_1 \in \mathcal{Q}_1^0} \min_{q_2 \in \mathcal{Q}_2^0} V_\mathcal{B}(q_1, q_2), \max_{q_2 \in \mathcal{Q}_2^0} \min_{q_1 \in \mathcal{Q}_1^0} V_\mathcal{B}(q_1, q_2)\right) \tag{1}$$

Then, T_1 and T_2 are approximately bisimilar with the precision δ.

Thus, the challenge consists in developing methods to compute bisimulation functions for several classes of transition systems. In the following, this is done for nonlinear dynamical systems.

III. Approximate Bisimulations for Nonlinear Dynamical Systems

We consider nonlinear dynamical systems of the form:

$$\Delta_i = \begin{cases} \dot{x}_i(t) &= f_i(x_i(t), u_i(t)), \\ y_i(t) &= g_i(x_i(t)) \end{cases}, \ i = 1, 2$$

with $x_i(t) \in \mathbb{R}^{n_i}$, $x_i(0) \in I_i$ where I_i is a compact subset of $\mathbb{R}^{n_i}$, $u_i(t) \in U_i$ where U_i is a compact subset of $\mathbb{R}^{m_i}$ and $y_i(t) \in \mathbb{R}^{p_i}$. We assume that Δ_1 and Δ_2 have the same observation space (*i.e.* $\mathbb{R}^{p_1} = \mathbb{R}^{p_2} = \mathbb{R}^p$) which is equipped with the usual Euclidean distance. In the spirit of [15], we rewrite the nonlinear dynamical system Δ_i as the following labeled transition system $T_i = (\mathcal{Q}_i, \Sigma_i, \to_i, \mathcal{Q}_i^0, \Pi_i, \langle\langle . \rangle\rangle_i)$, where:

- the set of states is $\mathcal{Q}_i = \mathbb{R}^{n_i}$,
- the set of labels is $\Sigma_i = \mathbb{R}^+$,
- the transition relation $\to_i$ is given by $x_i \xrightarrow{t}_i x_i'$ if and only if there exist a locally measurable function $u_i(.)$ and an absolutely continuous function $z_i(.)$ such that $z_i(0) = x_i$, $z_i(t) = x_i'$ and
$$\forall s \in [0, t], u_i(s) \in U_i \text{ and } \dot{z}_i(s) = f_i(z_i(s), u_i(s)),$$
- the set of initial states is $\mathcal{Q}_i^0 = I_i$,
- the set of observations is $\Pi_i = \mathbb{R}^p$,
- the observation map is given by $\langle\langle x \rangle\rangle_i = g_i(x)$.

Note that the transition system is nondeterministic, since for a given t, there are many possible evolutions from a state. We define the following notations: $x = [x_1 \ x_2]^T$ and

$$f(x, u_1, u_2) = \begin{bmatrix} f_1(x_1, u_1) \\ f_2(x_2, u_2) \end{bmatrix}, \ g(x) = g_1(x_1) - g_2(x_2).$$

We now consider the problem of computing a bisimulation function between the nonlinear dynamical systems Δ_1 and Δ_2. The following proposition gives a more tractable characterization than Definition 2.5.

Proposition 3.1: Let $p : \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \to \mathbb{R}^+$ be differentiable and let ∇p denote its gradient. If for all $x \in \mathbb{R}^{n_1 + n_2}$,

$$p(x) \geq g(x)^T g(x), \tag{2}$$

$$\max_{u_1 \in U_1} \min_{u_2 \in U_2} \nabla p(x)^T f(x, u_1, u_2) \leq 0, \tag{3}$$

$$\max_{u_2 \in U_2} \min_{u_1 \in U_1} \nabla p(x)^T f(x, u_1, u_2) \leq 0, \tag{4}$$

then $V_{\mathcal{B}}(x) = \sqrt{p(x)}$ is a bisimulation function.

Due to the lack of space, the proof of this result is not stated here.

Remark 3.2: The concept of bisimulation function is reminiscent of robust control Lyapunov functions [5], [12], though they slightly differ in spirit. Indeed, considering the input u_1 as a disturbance and the input u_2 as a control variable, the interpretation of equation (3) is that for all disturbances their exist a control such that the bisimulation function decreases during the evolution of the system. In this context, u_2 may have full knowledge (and be a function) of u_1. In comparison, a robust control Lyapunov function would require that their exist a control u_2 such that for all possible (and unknown) disturbances u_1 the function decreases during the evolution of the system. Therefore robust control Lyapunov functions require stronger conditions than bisimulation functions.

Note that many functions may satisfy equations (2), (3), and (4), which give sufficient conditions for obtaining a bisimulation function. Furthermore, by Corollary 2.7, we know that we can evaluate the precision of the approximate bisimulation by solving the static game (1). Thus, the smaller the bisimulation function, the smaller the value of the game, the tighter the precision of the approximation.

IV. Sum of Squares Formulation

In order to design computational methods to compute bisimulation functions, we restrict our study to the class of deterministic (autonomous) systems (*i.e.* $f(x, u_1, u_2) = f(x)$). Under this assumptions, equations (3) and (4) become equivalent, and reduce to a Lyapunov-like condition. For linear dynamical systems, it is well known that Lyapunov functions can be found in the class of quadratic forms and that they can be computed solving some linear matrix inequalities. In [7], we showed that, for linear systems, quadratic bisimulation functions are universal for linear systems. For nonlinear systems, the set of quadratic forms is often too restrictive to find global Lyapunov functions or bisimulation functions.

Recently [14], [16], [17], [18], it has been shown that functions that can be expressed as sum of squares are good candidates for Lyapunov functions of nonlinear systems. Moreover, these can be computed using semidefinite programming [19], [21]. In this section, we show that a similar approach can be used for the computation of bisimulation functions for nonlinear systems.

A. Sum of Squares Programs

A multivariate polynomial $p(x)$ is a sum of squares, if there exist polynomials $q_1(x), \dots, q_m(x)$, such that

$$p(x) = \sum_{i=1}^{i=m} q_i^2(x).$$

Hence, a sum of squares is a positive function. Moreover, it has been shown [16] that the condition $p(x)$ *is a sum of squares* is computationally much more tractable than the condition $p(x) \geq 0$.

For a large range of computational problems involving polynomial inequality constraints (such as Lyapunov function search or constrained optimization), replacing nonnegativity conditions by sum of squares constraints leads to a conservative approximation of the problem (in the sense that all the constraints of the initial problem are satisfied by the solution of the approximate one) which can be solved using semidefinite programming.

The sum of squares optimization toolbox SOSTOOLS [17], [18] allows to solve several computational problems involving sum of squares constraints. Though SOSTOOLS can handle more complex sum of squares programs (*e.g.* with equality constraints), we will only consider programs in the following class:

$$\begin{gathered}\text{Minimize}_{c \in \mathbb{R}^N}\ \omega^T c, \qquad (5)\\ \text{subject to } b_j(x) + c^T a_j(x) \textit{ is a sum of squares}\\ \text{for all } j \in \{1, \dots, J\}\end{gathered}$$

where ω is a vector of $\mathbb{R}^N$ and for all $j \in \{1, \dots, J\}$, $a_j(x)$ is a N-dimensional vector of polynomials and $b_j(x)$ is a polynomial.

SOSTOOLS translates program (5) into an equivalent semidefinite program for which efficient solvers exist (see for instance [19], [21]).

B. Computing Bisimulation Functions using Sum of Squares

Let us assume that the vector fields f_1 and f_2 as well as the observation maps g_1 and g_2 are vectors of polynomials. We search a bisimulation function between the systems Δ_1 and Δ_2 under the form :

$$V_{\mathcal{B}}(x) = \sqrt{p(x)}$$

where $p(x)$ is a sum of squares. Rewriting the inequalities (2) and (3) as sum of squares constraints, we have:

$$\begin{aligned} p(x) - g(x)^T g(x) &\quad \textit{is a sum of squares}, \qquad (6)\\ -\nabla p(x)^T f(x) &\quad \textit{is a sum of squares}. \qquad (7)\end{aligned}$$

Let $a(x)$ be a N-dimensional vector of polynomials and ω be an element of $\mathbb{R}^N$. We denote by $J_a(x)$ the Jacobian matrix of $a(x)$. Let us consider the following sum of squares program

$$\text{Minimize}_{c \in \mathbb{R}^N}\ \omega^T c \qquad (8)$$

subject to

$$\begin{aligned} -g(x)^T g(x) + c^T a(x) &\quad \textit{is a sum of squares},\\ -c^T J_a(x) f(x) &\quad \textit{is a sum of squares}.\end{aligned}$$

It belongs to the class of program (5). Hence, it can be solved using SOSTOOLS. Moreover, it is clear that the function $p(x) = c^T a(x)$ where c is the solution of (8) satisfies inequalities (6) and (7). Therefore,

$$V_{\mathcal{B}}(x) = \sqrt{c^T a(x)}$$

is a bisimulation function between the dynamical systems Δ_1 and Δ_2.

Let us remark that ω and $a(x)$ are arbitrarily chosen. A bad choice can result in program (8) being infeasible (*i.e.* there does not exist any c such that the sum of squares constraints are satisfied). Even if it is feasible, the resulting bisimulation function may be inaccurate and therefore of poor utility. It appears that the choice of ω and $a(x)$ is critical for the success of the method. This is illustrated in the next section with an example.

V. Example

Let us consider the following three dimensional nonlinear dynamical system:

$$\Delta : \begin{cases} \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} -(1+\gamma x_2^2)x_1 \\ -\frac{1-\gamma x_1^2}{2}x_2 + 2x_3 \\ -(1-\gamma x_1)2x_2 - \frac{1}{2}x_3 \end{bmatrix} \\ \\ \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} \gamma x_1 + x_2 \\ x_3 \end{bmatrix} \end{cases} \quad (9)$$

where $\gamma = 0.1$. We assume that the initial values of (x_1, x_2, x_3) are in the subset of $\mathbb{R}^3$:

$$I = [-2, 2] \times \{0\} \times [4, 6].$$

In this section, we consider two approximations of Δ. The first one consists a three dimensional linear system while the second one is a two dimensional system. We show that these linear systems are approximately bisimilar to the nonlinear system Δ by computing bisimulation functions. Finally, we show that such approximations can be useful for the safety verification of nonlinear systems.

A. Three dimensional approximation

It is well known that (under some assumptions) a nonlinear dynamical system is locally topologically equivalent to its linearization at a singular point. Δ has an unique singular point which is the origin. Then, around the origin Δ can be approximated by the following linear system:

$$\tilde{\Delta} : \begin{cases} \begin{bmatrix} \dot{\tilde{x}}_1 \\ \dot{\tilde{x}}_2 \\ \dot{\tilde{x}}_3 \end{bmatrix} = \begin{bmatrix} -\tilde{x}_1 \\ -\frac{1}{2}\tilde{x}_2 + 2\tilde{x}_3 \\ -2\tilde{x}_2 - \frac{1}{2}\tilde{x}_3 \end{bmatrix} \\ \\ \begin{bmatrix} \tilde{y}_1 \\ \tilde{y}_2 \end{bmatrix} = \begin{bmatrix} \gamma\tilde{x}_1 + \tilde{x}_2 \\ \tilde{x}_3 \end{bmatrix} \end{cases}$$

Let $\tilde{\Delta}$ have the same set of initial values than Δ. We can show that the approximation is not only local but actually global by computing a bisimulation function between Δ and $\tilde{\Delta}$.

Remark 5.1: It is easy to check that the subset of $\mathbb{R}^6$

$$\mathcal{B} = \{x_1 = 0, \tilde{x}_1 = 0, x_2 = \tilde{x}_2, x_3 = \tilde{x}_3\} \quad (10)$$

is an exact bisimulation between Δ and $\tilde{\Delta}$. However, Δ and $\tilde{\Delta}$ are not exactly bisimilar because for $(x_1, x_2, x_3) \in I$ with $x_1 \neq 0$ (respectively $(\tilde{x}_1, \tilde{x}_2, \tilde{x}_3) \in I$ with $\tilde{x}_1 \neq 0$) there does not exist $(\tilde{x}_1, \tilde{x}_2, \tilde{x}_3) \in I$ (respectively $(x_1, x_2, x_3) \in I$) such that $(x_1, x_2, x_3, \tilde{x}_1, \tilde{x}_2, \tilde{x}_3) \in \mathcal{B}$.

As mentioned before, we have to define the shape of the bisimulation function (*i.e.* the vector of polynomials $a(x)$) as well as the cost function (*i.e.* the vector ω) to use the sum of squares formulation (8). We first try to find a bisimulation function with a very simple shape using the following vector of polynomials:

$$a(x) = [x_1^2, x_2^2, x_3^2, \tilde{x}_1^2, \tilde{x}_2^2, \tilde{x}_3^2]^T.$$

The cost function is chosen so that we globally try to minimize the value of the bisimulation function:

$$\omega = [1, 1, 1, 1, 1, 1]^T.$$

For these choices of $a(x)$ and ω, the sum of squares program (8) is feasible and hence a bisimulation function exists. The associated sum of squares is

$$\begin{aligned} c^T a(x) \;=\; & 1.061x_1^2 + 1.924x_2^2 + 1.976x_3^2 + \\ & 0.234\tilde{x}_1^2 + 2.336\tilde{x}_2^2 + 2.025\tilde{x}_3^2. \end{aligned}$$

Hence, we proved that Δ and $\tilde{\Delta}$ are approximately bisimilar. However, this bisimulation is not good for the following reasons. Let us consider game (1). In $p(x)$ the variables of Δ and the variables of $\tilde{\Delta}$ are clearly decoupled. Hence, it is clear that with our bisimulation function, the maximization over (x_1, x_2, x_3) (respectively $(\tilde{x}_1, \tilde{x}_2, \tilde{x}_3)$) and the minimization over $(\tilde{x}_1, \tilde{x}_2, \tilde{x}_3)$, (respectively (x_1, x_2, x_3)) can be done independently. This means that for the approximation of a trajectory of Δ, the choice of the initial value of $\tilde{\Delta}$ does not depend on the initial value of Δ. Thus it is clear that this function does not carry the necessary information to evaluate the quality of the approximation. This is mainly due to the fact that when we chose the shape of our function we did not use the information we had on the approximation process.

Let us remark that when approximating Δ by $\tilde{\Delta}$, we implicitly define $\tilde{x}_1$ (respectively $\tilde{x}_2$, $\tilde{x}_3$) as an approximation of x_1, (respectively x_2, x_3). Therefore, this should be visible in the shape of the bisimulation function.

Furthermore, according to Theorem 2.6, the zero set of a bisimulation function is an exact bisimulation. Hence, in order to characterize the approximation as well as possible, the vector of polynomials $a(x)$ should be chosen so that the zero set of the bisimulation function is the relation $\mathcal{B}$ defined by equation (10). This means that for all $x \in \mathcal{B}$, we should have $a(x) = 0$. Hence let us consider the following vector of polynomials:

$$a(x) = \begin{bmatrix} (x_1 - \tilde{x}_1)^2 \\ (x_2 - \tilde{x}_2)^2 \\ (x_3 - \tilde{x}_3)^2 \\ x_1^2 \\ (x_1 - \tilde{x}_1)(x_2 - \tilde{x}_2) \\ (x_2 - \tilde{x}_2)(x_3 - \tilde{x}_3) \\ (x_1 - \tilde{x}_1)(x_3 - \tilde{x}_3) \\ x_1(x_1 - \tilde{x}_1) \\ x_1(x_2 - \tilde{x}_2) \\ x_1(x_3 - \tilde{x}_3) \\ x_1^4 \end{bmatrix}$$

The fourth order polynomial x_1^4 was added as a component of $a(x)$ because the sum of square program (8) was infeasible using the vector composed of the first ten components.

It remains to define the vector ω used in the cost function. Let us consider game (1) where the bisimulation function is of the form $V_{\mathcal{B}}(x) = \sqrt{c^T a(x)}$. For all $(x_1, x_2, x_3) \in I$, there exists $(\tilde{x}_1, \tilde{x}_2, \tilde{x}_3) \in I$ (take for instance $(x_1, x_2, x_3) = (\tilde{x}_1, \tilde{x}_2, \tilde{x}_3)$) such that

$$V_B(x_1, x_2, x_3, \tilde{x}_1, \tilde{x}_2, \tilde{x}_3) \leq \sqrt{c_4 x_1^2 + c_{11} x_1^4}$$

and conversely. Hence, a way to minimize the value the game (1) is to minimize its upper bound

$$\max_{(x_1, x_2, x_3) \in I} \sqrt{c_4 x_1^2 + c_{11} x_1^4} = \sqrt{4c_4 + 16c_{11}}$$

The cost function was chosen so that we try to minimize $c_4 + 4c_{11}$. The sum of squares associated to the bisimulation function is then

$$\begin{aligned} c^T a(x) \;\; = \;\; & 0.020(x_1 - \tilde{x}_1)^2 + 1.285(x_2 - \tilde{x}_2)^2 + \\ & 1.081(x_3 - \tilde{x}_3)^2 + 0.075x_1^2 + \\ & 0.181(x_1 - \tilde{x}_1)(x_2 - \tilde{x}_2) - \\ & 0.067(x_2 - \tilde{x}_2)(x_3 - \tilde{x}_3) + \\ & 0.015(x_1 - \tilde{x}_1)(x_3 - \tilde{x}_3) - \\ & 0.036x_1(x_1 - \tilde{x}_1) - 0.181x_1(x_2 - \tilde{x}_2) - \\ & 0.015x_1(x_3 - \tilde{x}_3) + 0.010x_1^4. \end{aligned}$$

Hence, we showed that the systems Δ and $\tilde{\Delta}$ are approximately bisimilar and that the precision of the approximate bisimulation is bounded by $\sqrt{4c_4 + 16c_{11}} = 0.679$.

B. Two dimensional approximation

When looking at the previous bisimulation function, it appears that the approximation of x_1 by $\tilde{x}_1$ is a much less important factor than the approximation of x_2 by $\tilde{x}_2$ or x_3 by $\tilde{x}_3$. Therefore, let us consider the following two dimensional linear approximation of the three dimensional nonlinear system Δ.

$$\tilde{\Delta} : \left\{ \begin{array}{lcl} \left[\begin{array}{c} \dot{\tilde{x}}_2 \\ \dot{\tilde{x}}_3 \end{array} \right] & = & \left[\begin{array}{c} -\frac{1}{2}\tilde{x}_2 + 2\tilde{x}_3 \\ -2\tilde{x}_2 - \frac{1}{2}\tilde{x}_3 \end{array} \right] \\ \\ \left[\begin{array}{c} \tilde{y}_1 \\ \tilde{y}_2 \end{array} \right] & = & \left[\begin{array}{c} \tilde{x}_2 \\ \tilde{x}_3 \end{array} \right] \end{array} \right. \qquad (11)$$

with the set of initial values

$$\tilde{I} = \{0\} \times [4, 6].$$

Following the same approach than before we can compute a bisimulation function between Δ and $\tilde{\Delta}$. Actually, after some experimentations, we found that a bisimulation function of the form $V_{\mathcal{B}}(x) = \sqrt{c^T a(x)}$ could be computed for the following very simple vector of polynomials:

$$a(x) = \left[(x_2 - \tilde{x}_2)^2, (x_3 - \tilde{x}_3)^2, x_1^2, x_1^4 \right]^T.$$

It is easy to see that the value of game (1) can be bounded by $\sqrt{4c_3 + 16c_4}$. Therefore, to define the cost function we used $\omega = (0, 0, 1, 4)^T$. The sum of squares associated to the bisimulation function is then

$$\begin{aligned} c^T a(x) \;\; = \;\; & 1.205(x_2 - \tilde{x}_2)^2 + 1.202(x_3 - \tilde{x}_3)^2 + \\ & 0.059x_1^2 + 0.007x_1^4. \end{aligned}$$

We proved that the systems Δ and $\tilde{\Delta}$ are approximately bisimilar and that the precision of the approximate bisimulation is bounded by $\sqrt{4c_3 + 16c_4} = 0.590$.

C. Safety verification

We now show how these results can be used for the approximation of a nonlinear system in the context of safety verification.

Let Δ be a nonlinear system. Then $\mathrm{Reach}(\Delta)$ denotes the reachable set of Δ and is defined as the subset of $\mathbb{R}^p$ of points reachable by the external trajectories of Δ. We consider the problem of checking wether the intersection of $\mathrm{Reach}(\Delta)$ with a set Π_F of unsafe sets is empty or not. Though recent progress has been made in the reachability analysis of linear systems allowing to design efficient and scalable methods [6], [10], [23], the reachability analysis of nonlinear systems [1], [13] remains expensive and is an important issue of the safety verification of continuous and hybrid systems.

Our approach consists in approximating Δ by a linear system $\tilde{\Delta}$ and in performing the reachability analysis on $\tilde{\Delta}$.

Let us remark that $\tilde{\Delta}$ is not necessarily the linearization of Δ. Furthermore, Δ and $\tilde{\Delta}$ do not need to be of the same dimension. The only requirement is that Δ and $\tilde{\Delta}$ are approximately bisimilar with some precision δ. This can be proved by constructing a bisimulation function between Δ and $\tilde{\Delta}$ using a sum of squares program. Then, the precision δ can be evaluated by solving the game (1).

If the distance of $\mathrm{Reach}(\tilde{\Delta})$ to Π_F is greater than δ then from Theorem 2.4 it is straightforward that Δ is safe.

We applied this method to the three dimensional nonlinear system Δ given by equation (9). On Figure 1, we represented the trajectories of Δ for several initial values in I. The circle represents the set of unsafe sets Π_F, we can reasonably conjecture that the system is safe though we did not perform the reachability analysis for Δ.

Δ was approximated by the two dimensional linear system $\tilde{\Delta}$ defined in equation (11). Δ and $\tilde{\Delta}$ are approximately bisimilar with the precision $\delta = 0.590$. The reachable set $\mathrm{Reach}(\tilde{\Delta})$ was computed using zonotope techniques [6]. On Figure 2, we can see that the distance between $\mathrm{Reach}(\tilde{\Delta})$ and Π_F is greater than δ. Therefore, this allows us to conclude that three dimensional nonlinear system Δ is effectively safe.

This example also illustrates the important point that robustness simplifies verification. Indeed, if the distance between $\mathrm{Reach}(\Delta)$ and Π_F would have been smaller (and hence the system Δ less robust with regard to the safety property) then the approximation of Δ by the two dimensional linear system $\tilde{\Delta}$ might not have been sufficient to check the safety of Δ. Generally, the more robustly safe a system is, the larger the distance from the unsafe safe, resulting in larger model compression and easier safety verification.

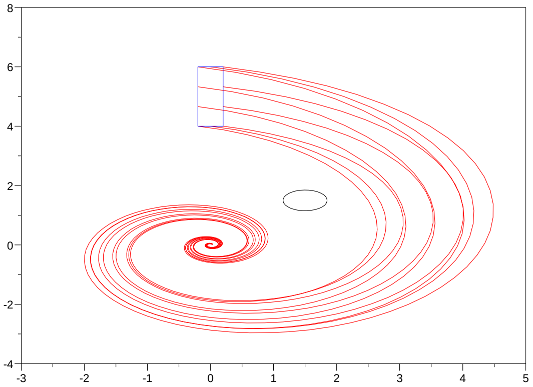

Fig. 1. Several trajectories of the three dimensional nonlinear system Δ, the circle represents the set of unsafe sets Π_F. The system Δ seems to be safe.

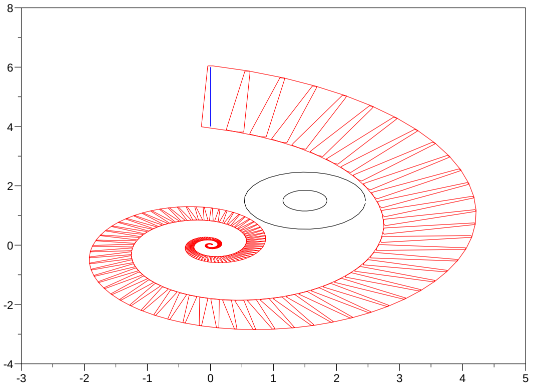

Fig. 2. Reachable set of the two dimensional linear system $\tilde{\Delta}$. The inner circle represents the set of unsafe sets Π_F and the outer circle consists of the points whose distance to Π_F is smaller than the precision of the approximate bisimulation.

VI. Conclusion

In this paper, we applied the framework of approximate bisimulations to the approximation of nonlinear systems. We proposed a methodology to compute bisimulation functions for autonomous systems using sum of squares programs and we showed how these functions could be used in the context of safety verification. Future research includes the development of a methodology for the automatic choice of the shape of a bisimulation function (*i.e.* $a(x)$) and of the cost function used in the sum of squares program (*i.e.* ω). We also intend to develop methods for computing bisimulation functions for nondeterministic nonlinear systems (with disturbances) and hybrid systems.

References

[1] E. Asarin, T. Dang, and A. Girard, “Reachability of non-linear systems using conservative approximations,” in *Hybrid Systems: Computation and Control*, ser. LNCS, vol. 2623. Springer, 2003, pp. 22–35.

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

[3] L. de Alfaro, M. Faella, and M. Stoelinga, “Linear and branching metrics for quantitative transition systems,” in *ICALP’04*, ser. LNCS, vol. 3142. Springer, 2004, pp. 1150–1162.

[4] J. Desharnais, V. Gupta, R. Jagadeesan, and P. Panangaden, “Metrics for labelled markov processes,” *Theoretical Computer Science*, vol. 318, no. 3, pp. 323–354, June 2004.

[5] R. A. Freeman and P. V. Kokotovic, “Inverse optimality in robust stabilization,” *SIAM J. Control and Optimization*, vol. 34, no. 4, pp. 1365–1391, July 1996.

[6] A. Girard, “Reachability of uncertain linear systems using zonotopes,” in *Hybrid Systems: Computation and Control*, ser. LNCS, vol. 3414. Springer, 2005, pp. 291–305.

[7] A. Girard and G. J. Pappas, “Approximate bisimulations for constrained linear systems,” in *Proc. 44th IEEE Conference on Decision and Control and European Control Conference*, December 2005.

[8] ——, “Approximation metrics for discrete and continuous systems,” May 2005, technical Report MS-CIS-05-10, Dept. of CIS, University of Pennsylvania.

[9] E. Haghverdi, P. Tabuada, and G. J. Pappas, “Bisimulation relations for dynamical, control, and hybrid systems,” *Theoretical Computer Science*, vol. 342, pp. 229–261, 2005.

[10] Z. Han and B. H. Krogh, “Reachability of hybrid control systems using reduced-order models,” in *Proc. American Control Conference*, Boston, MA, July 2004.

[11] S. Lall, J. E. Marsden, and S. Glavaski, “A subspace approach to balanced truncation for model reduction of nonlinear control systems,” *International Journal on Robust and Nonlinear Control*, vol. 12, no. 5, pp. 519–535, 2002.

[12] D. Liberzon, E. D. Sontag, and Y. Wang, “Universal construction of feedback laws achieving ISS and integral-ISS disturbance attenuation,” *Systems and Control Letters*, vol. 46, pp. 111–127, 2002.

[13] I. Mitchell and C. Tomlin, “Level set methods for computation in hybrid systems,” in *Hybrid Systems: Computation and Control*, ser. LNCS, vol. 1790. Springer, 2000.

[14] A. Papachristodoulou and S. Prajna, “On the construction of Lyapunov functions using the sum of squares decomposition,” in *Proc. 41st IEEE Conference on Decision and Control*, December 2002.

[15] G. J. Pappas, “Bisimilar linear systems,” *Automatica*, vol. 39, no. 12, pp. 2035–2047, December 2003.

[16] P. A. Parrilo, “Structured semidefinite programs and semialgebraic methods in robustness and optimization,” Ph.D. dissertation, California Institute of Technology, Pasadena, CA, 2000.

[17] S. Prajna, A. Papachristodoulou, P. Seiler, and P. A. Parrilo. (2004) SOSTOOLS, sum of squares optimization toolbox for MATLAB. User’s guide. [Online]. Available: http://www.cds.caltech/edu/sostools

[18] ——, “SOSTOOLS and its control applications,” in *Positive Polynomials in Control*. Springer, 2005, to appear.

[19] J. F. Sturm, “Using SEDUMI 1.02, a MATLAB toolbox for optimization over symmetric cones,” *Optimization Methods and Softwares*, vol. 11-12, pp. 625–653, 1999.

[20] P. Tabuada and G. J. Pappas, “Bisimilar control affine systems,” *Systems and Control Letters*, vol. 52, pp. 49–58, 2004.

[21] K. C. Toh, R. H. Tutuncu, and M. J. Todd. SDPT3 - a MATLAB software package for semidefinite quadratic linear programming. [Online]. Available: http://www.math.nus.edu.sg/˜mattohkc/sdpt3.html

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

[23] H. Yazarel and G. J. Pappas, “Geometric programming relaxations for linear system reachability,” in *Proc. American Control Conference*, Boston, MA, June 2004.