Discrete Event Dyn Syst (2017) 27:443–461
DOI 10.1007/s10626-017-0244-y

Order-reduction abstractions for safety verification of high-dimensional linear systems

Hoang-Dung Tran[1] · Luan Viet Nguyen[2] · Weiming Xiang[1] · Taylor T. Johnson[1]

Received: 31 August 2016 / Accepted: 10 March 2017 / Published online: 12 April 2017

Abstract Order-reduction is a standard automated approximation technique for computer-aided design, analysis, and simulation of many classes of systems, from circuits to buildings. To be used as a sound abstraction for formal verification, a measure of the similarity of behavior must be formalized and computed, which we develop in a computational way for a class of asymptotic stable linear systems as the main contributions of this paper. We have implemented the order-reduction as a sound abstraction process through a source-to-source model transformation in the HyST tool and use SpaceEx to compute sets of reachable states to verify properties of the full-order system through analysis of the reduced-order system. Our experimental results suggest systems with thousand of state variables can be reduced to systems with tens of state variables such that the order-reduction overapproximation error is small enough to prove or disprove safety properties of interest using current reachability analysis tools. Our results illustrate this approach is effective in tackling the state-space explosion problem for verification of high-dimensional linear systems.

Keywords Abstraction · Model reduction · Order reduction · Verification · Reachability analysis

1 Introduction

The state-space explosion problem is a fundamental challenge in model checking and automated formal verification that has received significant attention from the verification community. Among many solutions, abstractions based on the concepts of exact and

✉ Taylor T. Johnson
taylor.johnson@gmail.com

[1] Vanderbilt University, Nashville, TN, USA

[2] University of Texas at Arlington, Arlington, TX, USA

approximate simulation and bisimulation relations have proved to be effective approaches for obtaining smaller state spaces by abstracting away information that is not needed in the verification process. Such abstractions have been applied broadly to simplify the controller synthesis and safety verification process for complex systems. Exact bisimulation relation-based abstractions for safety verification and controller synthesis have been investigated widely in the last decade (Pappas 2003; van der Schaft 2004; Tanner and Pappas 2003; Tabuada and Pappas 2004). In this context, the outputs of the abstract system capture exactly the outputs of the original system. As pointed out in Girard et al. (2008) and Girard and Pappas (2007), the term "exact" is not adequate when dealing with continuous and hybrid systems observed over real numbers since there may be numerical errors in observation, noise, and other imperfections. To obtain an abstraction that guarantees more robust relationship between systems, approximate bisimulation has been proposed and studied extensively in recent years (Girard et al. 2008; Girard and Pappas 2007; Julius 2006; Girard et al. 2006; Islam et al. 2015). The main advantage of this approach is that they allow a bounded precision δ which describes how "far off" the executions of the abstraction may be from those of the original system. Then, verifying whether the executions of the original system reach an unsafe region U can be reduced to verify whether the executions of the abstraction (with much lower dimension) reach the δ-neighborhood of the unsafe region U. Thus, finding an efficient way of computing a tight bound on the precision becomes an essential task for this approach. The proposed method shows a great benefit when it can deal both stable and unstable systems. Balanced truncation model reduction, a well-known technique in control (Antoulas et al. 2001), has recently been applied to obtain abstractions for formal verification of continuous and hybrid systems (Han and Krogh 2004; Han 2005). The bounded precision between a system and its abstraction, which is also essential in this framework, is determined using simulation.

Inspired by the results reported in Girard and Pappas (2007) and Han and Krogh (2004), in this paper, we discuss and improve the computation frameworks in existing techniques to make them more applicable to practical high-dimensional systems. The main contribution of our work is improving the way of computing and using the precision δ. Our method is shown to be efficient, robust and scalable compared with existing approaches over a set of practical benchmarks (several to a thousand dimensions). We implement the method as a model transformation pass within the HyST model transformation tool (Bak et al. 2015) which enables us to easily apply the techniques and compare results.

The remainder of the paper is organized as follows. Section 2 gives essential definitions used throughout the paper. Section 3 presents methods to find output abstractions of the linear time invariant (LTI) systems using the balanced truncation model reduction method. Section 4 discusses how to verify safety properties for a full-order LTI system using its output abstraction. Section 5 describes our implementation of the method in a prototype tool, and presents a number of examples to illustrate and evaluate the benefits of our method.

2 Preliminaries

Consider two *asymptotically stable* LTI systems:

$$M_n: \begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y(t) = Cx(t). \end{cases} \qquad M_k: \begin{cases} \dot{x}_r(t) = A_r x_r(t) + B_r u(t), \\ y_r(t) = C_r x_r(t). \end{cases}$$

where $x(t) \in \mathbb{R}^n$, $x_r(t) \in \mathbb{R}^k$ are the *system states*; $y(t) \in \mathbb{R}^p$, $y_r(t) \in \mathbb{R}^p$ are the *system outputs*; $u(t)$ is the *control input*; $A \in \mathbb{R}^{n\times n}$, $A_r \in \mathbb{R}^{k\times k}$, $B \in \mathbb{R}^{n\times m}$, $B_r \in \mathbb{R}^{k\times m}$ and

$C \in \mathbb{R}^{p\times n}$, $C_r \in \mathbb{R}^{p\times k}$ are system matrices. The initial set of states of M_n and M_k are respectively denoted by $X_0(M_n) \subseteq \mathbb{R}^n$, $X_0(M_k) \subseteq \mathbb{R}^k$ and the set of control inputs is denoted by U. We write initial conditions as $x(0) \in X_0(M_n)$, $x_r(0) \in X_0(M_k)$ and the control input as $u(s) \in U, \forall s \in [0, t]$. In this paper, we assume that M_n is observable and controllable.

Definition 1 Output Abstraction. M_k is called a *k-dimensional output abstraction of* M_n with *precision* $\delta = [\delta_1, \delta_2, \ldots, \delta_p]^T$, denoted as M_k^δ, where each δ_i is a finite positive real if, for $\forall x(0) \in X_0(M_n)$ and $u \in U, \exists x_r(0) \in X_0(M_k)$ such that, $\forall t \geq 0$, $\left\| y^i(t) - y_r^i(t) \right\| \leq \delta_i$, $1 \leq i \leq p$ where $y^i(t)$ is the i^{th} component of the output y at time t, and $\| \cdot \|$ denotes the Euclidean norm.

Informally, the output abstraction *behaviors*, which are defined as the trajectories of the output over real time intervals, will approximate within δ the *behaviors* of the full-order system M_n for all time.

Definition 2 Output Reach Set (Han and Krogh 2004). The output reach sets *at a time instant t* and *over a time interval* $[0, t_f]$, $t_f \geq 0$ of M_n respectively are:

$$R_t(M_n) \triangleq \{y(t, u, x(0)) | y(t, u, x(0)) = Ce^{At}x(0) + \int_0^t Ce^{A(t-\tau)}Bu(\tau)d\tau\},$$
$$R_{[0,t_f]}(M_n) \triangleq \bigcup\nolimits_{t\in[0,t_f]} R_t(M_n), \text{ where } x(0) \in X_0(M_n),\ u(\tau) \in U, \forall \tau \in [0, t].$$

Definition 3 Safety Specification A *safety specification* $S(M_n)$ of an LTI system M_n formalizes the safety requirements for the output y of M_n, and is a predicate over the output y of M_n. Formally, $S(M_n) \subseteq \mathbb{R}^p$. The *dual unsafe specification* $U(M_n)$ of the system is also a predicate over the system output, i.e. $U(M_n) \subseteq \mathbb{R}^p$.

Definition 4 Safety Verification. The *time-bounded safety verification problem* is to verify whether the system M_n satisfies a safety specification $S(M_n)$ over an *interval of time* $[0, t_f]$ (t_f *is finite and positive*), which is described formally in terms of the output reach set as:

$$R_{[0,t_f]}(M_n) \cap \neg S(M_n) = \emptyset \Leftrightarrow M_n \vDash S(M_n),$$
$$R_{[0,t_f]}(M_n) \cap \neg S(M_n) \neq \emptyset \Leftrightarrow M_n \nvDash S(M_n).$$

If M_n satisfies $S(M_n)$, then it is *safe* and we write $M_n \vDash S(M_n)$. If M_n does not satisfy $S(M_n)$, then it is *unsafe* and we write $M_n \nvDash S(M_n)$. The notation "$\neg$" denotes the logical negation which corresponds to set complement in the formulas above.

Definition 5 Safety Specification Transformation The *safety specification transformation* is the process of finding the corresponding safety (or dually, unsafe) specification for the output abstraction M_k^δ denoted by $S(M_k^\delta) \in \mathbb{R}^p$ (and dually $U(M_k^\delta) \in \mathbb{R}^p$) from the safety specification $S(M_n)$ of the full-order system M_n to guarantee the safety relation defined by:

$$\begin{aligned} &R_{[0,t_f]}(M_k^\delta) \cap \neg S(M_k^\delta) = \emptyset \Rightarrow M_n \vDash S(M_n), \\ &R_{[0,t_f]}(M_k^\delta) \cap U(M_k^\delta) \neq \emptyset \Rightarrow M_n \nvDash S(M_n). \end{aligned} \tag{1}$$

The main objective of this paper is to perform the safety verification for high-dimensional LTI systems using output abstraction and its transformed safety specification to reduce the computational complexity.

3 Output abstractions from balanced truncation reduction

The balanced truncation model reduction (BTMR) (Moore 1981) is a well-known method in control which is used to find a reduced model M_k for an asymptotically stable large scale linear system M_n so that its output trajectory captures the output trajectory of the original system within a estimated bounded error under *an identity control input*. The first step in BTMR is to implement a balanced transformation $\tilde{x}(t) = Hx(t),\ H \in \mathbb{R}^{n\times n}$ to transform M_n to an equivalent balanced system $\tilde{M}_n$. The k-order reduced model M_k which is also asymptotically stable is then obtained by selecting the first k states in the state vector of $\tilde{M}_n$ and truncating the other $n-k$ states (i.e. $x_r = S\tilde{x}(t) = SHx(t)$, $S = \begin{pmatrix} I_{k\times k} & 0_{k\times(n-k)} \end{pmatrix}$). The $(n+k)$-dimensional (asymptotically stable) augmented system $\bar{M}_{n+k}$ is defined by:

$$\begin{aligned} \dot{\bar{x}} &= \bar{A}\bar{x} + \bar{B}u = \begin{pmatrix} \tilde{A} & 0 \\ 0 & A_r \end{pmatrix}\bar{x} + \begin{pmatrix} \tilde{B} \\ B_r \end{pmatrix}u, \\ \bar{y} &= \bar{C}\bar{x} = \begin{pmatrix} \tilde{C} & -C_r \end{pmatrix}\bar{x}, \end{aligned}$$

where $\bar{x} = \begin{pmatrix} \tilde{x} \\ x_r \end{pmatrix}$; $\tilde{A}$, $\tilde{B}$ and $\tilde{C}$ are the matrices of balanced system $\tilde{M}_n$.

To derive an output abstraction M_k^δ from the system M_n, we first use BTMR to obtain the reduced model M_k, and then determine the precision δ which relates not only to the control input $u(t)$ but also to the initial condition $x(0)$. Determining the precision δ is equivalent to determining the bounds of the augmented system's individual outputs. Let $e_1(t) = \bar{C}e^{\bar{A}t}\bar{x}(0)$ be the *zero input response* and $e_2(t) = \int_0^t \bar{C}e^{\bar{A}(t-\tau)}\bar{B}u(\tau)d\tau$ be the *zero state response*. It is easy to see that $\left\|y^i(t) - y_r^i(t)\right\| = \left\|\bar{y}^i(t)\right\| \le \left\|e_1^i(t)\right\| + \left\|e_2^i(t)\right\| = \delta_i$, where $h^i(t)$ denotes the i^{th} element of vector $h(t)$.

The following theorems obtain the theoretical bounds for $e_1(t)$ and $e_2(t)$. The sum of these two bounds gives us the precision δ.

Theorem 1 *Let* $\bar{x}(0) = \begin{pmatrix} Hx(0) & SHx(0) \end{pmatrix}^T$, *then the zero input response* $e_1(t)$ *of the asymptotically stable augmented system* $\bar{M}_{n+k}$ *satisfies the following inequality for all* $t \in \mathbb{R}_{\ge 0}$*:*

$$\left\|e_1^i(t)\right\| \le \left\|\bar{C}_i\right\| \cdot \sup_{x(0)\in X_0} \|\bar{x}(0)\|,\ 1 \le i \le p,$$

where $\mathbb{R}_{\ge 0}$ *is the set of non-negative real numbers,* $\bar{C}_i$ *is the row* i *of the matrix* $\bar{C}$.

The proof of Theorem 1 is given in Appendix A.1.

Theorem 2 *Let* $\bar{x}(0) = \begin{pmatrix} Hx(0) & SHx(0) \end{pmatrix}^T$ *and* $P_0 > 0$ *is the solution of the following optimization problem:*

$$\begin{aligned} &P_0 = min(trace(P))\ subject\ to \\ &P > 0,\ \bar{A}^T P + P\bar{A} < 0,\ \bar{C}_i^T \bar{C}_i \le P \end{aligned}$$

where $\bar{C}_i$ *is the row* i *of the matrix* $\bar{C}$. *Then, the zero input response* $e_1(t)$ *of the asymptotically stable augmented system* $\bar{M}_{n+k}$ *satisfies the following inequality for all* $t \in \mathbb{R}_{\ge 0}$*:*

$$\left\|e_1^i(t)\right\| \le \sup_{x(0)\in X_0} \sqrt{\bar{x}(0)^T P_0 \bar{x}(0)},\ 1 \le i \le p,$$

The proof of Theorem 2 is given in Appendix A.2.

Theorem 3 (Han and Krogh 2004; Obinata and Anderson 2012). *The error e_2 between the full-order asymptotically stable system M_n and its k-dimensional reduced system M_k satisfies the following inequality for all $t \in \mathbb{R}_{\geq 0}$:*

$$\left\| e_2^i(t) \right\| \leq (4 \sum_{j=k+1}^{n} (2j-1)\sigma_j) \cdot \|u(t)\|_\infty, \ 1 \leq i \leq p,$$

where σ_j is the j^{th} Hankel singular value of the system M_n and $\|u(t)\|_\infty = \sup_{t \in \mathbb{R}_{\geq 0}} \|u(t)\|$.

We note that the precision δ can be obtained using different methods all of which have their advantages and drawbacks. In Han and Krogh (2004), the authors propose a simulation-based approach to determine δ. To determine the bound of e_1, the authors simulate the full-order system and the reduced system from each vertex in a polyhedral representation of the initial set of states. The advantage in using this method is that it gives a very tight bound of e_1 while the drawback is that the number of simulations required can grow exponentially. For example, if the initial set is a hypercube in 100-dimensions, we have to simulate the full-order system and its reduced system with $2^n = 2^{100}$ vertices, which is infeasible even if each simulation takes little time. The bound of e_2 is determined by integrating the norm of the impulse response of the augmented system via simulation. This method is especially useful in practice since it gives a tight bound for e_2 with only m simulations, where m is the number of inputs. Alternatively, without separately computing the bounds of e_1 and e_2, the precision δ can be calculated by solving a set of LMI optimization problems on sets of initial states and inputs (Girard and Pappas 2007). This approach has advantages when dealing with small and medium-dimensional systems (less than 50 dimensions) and works for both stable and unstable systems. When the system dimension is large, the error bound obtained is overly conservative and may not be useful.

Exploiting the advantages and overcoming the drawbacks of existing approaches when dealing with practical systems are the main contributions of our approach. First, we compute the bounds for e_1 and e_2 separately, making the computation process more robust and thus produce less conservative results in comparison with (Girard and Pappas 2007). Second, Theorems 1 and 2 are proposed to overcome the drawback of simulation-based method (Han and Krogh 2004) as described above. It should be emphasized that in most circumstances, our approach can be used to compute the bound of e_1 and the simulation-based approach can be used to determine the bound of e_2. This combination is scalable for high-dimensional systems while still obtaining a good precision δ. Finally, the output abstraction defined based on individual outputs $(y^i(t), y_r^i(t))$ and precisions δ_i allows us to easily verify the safety of multi-output systems. Next, we briefly analyze the time complexity in computing the precision δ to show theoretically the scalability of different methods.

We analyze the complexity of the simulation-based approach (Han and Krogh 2004) in terms of number of simulations. For an n-dimension system with m inputs and p outputs, the number of vertices in a polyhedral initial set (in the worst case) is 2^n. Therefore, the number of simulations needed to determine the bound of e_1 is 2^n. For e_2, the number of simulations needed is m. Consequently, the number of simulations needed is $O((2^n + m))$. In Girard and Pappas (2007), to determine the precision δ, this method solves two LMIs and quadratic optimization problems on the sets of initial state and inputs in which the time complexity for solving two LMI constraints using interior point algorithms can be estimated by $O([(n^2+n)/2]^{2.75} \times 2^{1.5})$ (Vandenberghe and Boyd 1994; Nesterov et al. 1994) and the time complexity for solving the optimization problem in Girard and Pappas (2007) using interior point algorithms is $O([(n^2+n)/2]^3)$. Overall, the time complexity of computing the precision is $O([(n^2+n)/2]^3) + O([(n^2+n)/2]^{2.75} \times 2^{1.5})$, which can be

bounded by $O(n^6)$. In our approach, it is easy to see that Theorem 1 computation mainly relates to solving the optimization problem to find $\sup_{x(0)\in X_0} \|\bar{x}(0)\|$. The time complexity for solving this problem using interior point algorithms is $O((n+1)^{3.5})$. For Theorem 2, we first need to solve the eigenvalue problem (EVP) subject to two matrix inequalities that has time complexity $O([((n+k)^2+n+k)/2]^{2.75}\times 2^{1.5})$ if using interior point algorithms. Then, we need to solve the quadratic optimization problem that has time complexity $O((n+k)^3)$ (using the interior point algorithm). Overall, the time complexity of Theorem 2 can be bounded by $O((n+k)^{5.5})$. The computational cost of Theorem 3 in our approach is smaller compared to Theorems 1 and 2. Table 1 shows the simplified time complexity analysis of different approaches. The computation time of these approaches will be measured and discussed in detail in Section 5.

4 Safety verification with output abstractions

It should be emphasized that first, our approach is different from Han and Krogh (2004) in the way of using the precision δ where the precision is used to compute the reach set of the full-order system before using this reach set to verify the safety of the system. In our approach, the precision is used to obtain the safety specification of the output abstraction that satisfies the safety relation (1). Then, we verify the safety of the output abstraction to conclude safety of the original system. It is important to notice from Eq. 1 that it may be the case that we cannot conclude anything about the safety of the original system using the output abstraction. Second, the output abstraction is different from the approximate abstraction (Girard and Pappas 2007) which depends on a single precision δ because it is defined based on individual outputs $(y^i(t), y_r^i(t))$ and individual precisions δ_i. As a result, we can get a more accurate safety specification transformation based on the individual precisions. This transformation is convenient for verifying safety in practical multi-output systems where the magnitudes of element outputs are usually significantly different from each other. Our safety transformation rules are addressed in the following.

***S*(M_n) as convex polytopes** Assume that the safety specification of the full-order system is in the following form:

$$S(M_n) = \{y \in \mathbb{R}^p |\ \Gamma y + \Psi \le 0,\ \Gamma = [\alpha_{ij}] \in \mathbb{R}^{q\times p},\ \Psi = [\beta_i] \in \mathbb{R}^q\}, \tag{2}$$

Lemma 1 *Given $S(M_n)$ described by* Eq. 2*, then $S(M_k^\delta)$ and $U(M_k^\delta)$ defined as follows guarantee the safety relation* (1)*.*

$$\begin{aligned} S(M_k^\delta) &= \{y_r \in \mathbb{R}^p |\ \Gamma y_r + \overline{\Psi} \le 0,\ \overline{\Psi} = \Psi + \Delta\},\\ U(M_k^\delta) &= \{y_r \in \mathbb{R}^p |\ \Gamma y_r + \underline{\Psi} > 0,\ \underline{\Psi} = \Psi - \Delta\}. \end{aligned} \tag{3}$$

where $\Delta = [\Delta_i] \in \mathbb{R}^q,\ \Delta_i = \sum_{j=1}^p |\alpha_{ij}|\delta_j$.

Table 1 Time complexity of different methods to compute the precision δ

(Girard and Pappas 2007)	$O(n^6)$
(Han and Krogh 2004)	$O(2^n+m)$
Theorem 1	$O(n^{3.5})$
Theorem 2	$O((n+k)^{5.5})$

The proof is given in Appendix A.1.

***S*(M_n) as ellipsoids** Assume that the safety specification of the full-order system is described by an ellipsoid with radius R and centered at $a \in \mathbb{R}^p$ as:

$$S(M_n) = \{y \in \mathbb{R}^p | \ (y-a)^T Q(y-a) \leq R^2, \ a \in \mathbb{R}^p, \ R > 0, \ 0 < Q \in \mathbb{R}^{p \times p}\}, \quad (4)$$

Since Q is a symmetric matrix, there exists an orthogonal matrix $E = [l_1, l_2, .., l_p] = [\gamma_{ij}] \in \mathbb{R}^{p \times p}$ such that $E^T Q E = \Lambda = diag(\lambda_1, \ \lambda_2, ..., \lambda_p)$, where λ_i (> 0) is eigenvalue of Q and l_i is the eigenvector of Q corresponding to λ_i. The transformed safety and unsafe specifications of the output abstraction can be obtained using the following lemma.

Lemma 2 *Given $S(M_n)$ described by* Eq. 4*, then $S(M_k^\delta)$ and $U(M_k^\delta)$ defined as follows guarantee the safety relation* (1)*:*

$$\begin{aligned} S(M_k^\delta) &= \{y_r \in \mathbb{R}^p | \ (y_r - a)^T Q(y_r - a) \leq (R - \Delta_R)^2\}, \\ U(M_k^\delta) &= \{y_r \in \mathbb{R}^p | \ (y_r - a)^T Q(y_r - a) > (R + \Delta_R)^2\}, \\ \Delta_R &= \sqrt{\sum_{i=1}^{p} [\lambda_i (\sum_{j=1}^{p} |\gamma_{ij}| \delta_j)^2]}. \end{aligned}$$

The proof of this result is given in Appendix A.2.

We can see that the transformed safety specification of the output abstraction is also an ellipsoid (with smaller radius $R - \Delta_R$) located inside the original ellipse defining the safety specification of the full-order system. Meanwhile the corresponding transformed unsafe specification is defined by the region outside the larger ellipse with the radius $R + \Delta_R$.

Lemma 3 *Given an asymptotically stable LTI system M_n, whenever its output abstraction M_k^δ is safe or unsafe, it is sound to claim that the system M_n is safe or unsafe respectively.*

Proof According to Section 3, for a stable LTI system M_n, there exists an output abstraction M_k^δ. We can see that, for any control input $u \in U$ and initial state $x(0) \in X_0(M_n)$, there is a set of p output trajectories $y_i, 1 \leq i \leq p$ of M_n. From the definition of the output abstraction, there exists a corresponding initial state $x_r(0) \in X_0(M_k)$ such that the abstraction produces a set of p output trajectories $y_r^i, 1 \leq i \leq p$ in which the distance between two pair trajectories $\| y^i - y_r^i \|$ is always bounded by a sound δ_i *all the time*. Moreover from Lemmas 1 and 2, because the transformed specifications ($S(M_k^\delta)$ and $U(M_k^\delta)$) satisfy the safety relation (1), thus when the set of p output trajectories $y_r^i, 1 \leq i \leq p$ of M_k^δ satisfies the transformed safety specification $S(M_k^\delta)$, its corresponding set of p output trajectories $y_i, 1 \leq i \leq p$ of M_n also satisfies the original safety specification $S(M_n)$ *all the time*. Consequently, when the output abstraction is safe, it is sound to claim that the full-order system is safe. A similar proof can be given for the unsafe case. This completes the proof. □

Remark 1 It is useful to mention that the procedure of safety verification for high-dimensional linear systems using order-reduction abstraction can be done automatically. The core issue is checking whether the reach set of the abstraction returned by verification tools such as SpaceEx is contained inside the transformed safe set. This can be done in two steps by taking advantages of the existing powerful SMT solvers such as Z3 (De Moura and Bjørner 2008). First, the transformed safe set can be described by a predicate over the output

variables. Then, we check whether the predicate is satisfied by all vertices of the returned reach set from SpaceEx using a SMT solver.

We have studied using output abstraction for safety verification. Next, we evaluate and compare our approach with others via a set of benchmarks.

5 Case studies and evaluation

To evaluate the order-reduction abstraction method presented in this paper, we have implemented a software prototype that automatically creates output abstractions from full-order systems and applied it to a set of benchmarks (Chahlaoui and Van Dooren 2005; Frehse et al. 2011) presented in Table 2. The method is integrated in HyST by calling Matlab related functions.[1]

Precision and computation time evaluation The experiments were performed using Matlab 2014a and SpaceEx (Frehse et al. 2011) on a personal computer with the following configuration: Intel (R) Core(TM) i7-2677M CPU at 1.80GHz, 4GB RAM, and 64-bit Window 7. We set the upper limit for Matlab simulation and SpaceEx running time at two hours. In all experimental result tables, the term of "N/A" stands for *not applicable* and "OOT" stands for *timeout* when the result could not be computed within two hours.

Table 3 presents the precision δ and computation times of the different methods. In the table, the "mixed bound" column contains the bound of e_1 computed by Theorem 2 and the bound of e_2 determined by simulation and the corresponding precision δ. The "theoretical bound" column contains the bounds of e_1 and e_2 which are computed using Theorems 1 and 3 respectively. The table shows that, when the initial set of states X_0 is close to the origin, e.g. BM benchmark, the bounds of e_1 computed by Theorem 1 are fairly good and acceptable. However, conservative results are derived when the initial set of states X_0 is far from the origin, e.g. helicopter benchmark. Theorem 3 indicates that the theoretical bound of e_2 depends on Hankel singular values. This can be seen from the PDE benchmark where the theoretical bounds of e_2 for all cases of the output abstraction's dimension k are very small due to the fact that the Hankel singular values σ_k (which are not presented here) of the corresponding balanced system are very small (almost equal to zero) as $k \geq 5$. In contrast, in the helicopter benchmark, the theoretical bound of e_2 becomes larger when the lower dimension output abstraction is obtained. It is small when $k = 20$ because $\sum_{21}^{28} \sigma_j$ is small. The theoretical bound of e_2 becomes conservative as $k = 10$ since $\sum_{11}^{28} \sigma_j$ is large. From the results in the table, we see that for *low and medium-dimensional systems* ($n < 100$), the simulation-based approach (Han and Krogh 2004) gives very tight bounds for the errors, e.g, the motor control system and helicopter benchmarks. This approach is powerful when dealing with systems having a small number of vertices in the initial set. When the number of vertices increases (e.g, PDE benchmark), it is unable to apply the simulation-based approach due to the number of simulations growing exponentially. The approximate bisimulation approach (Girard and Pappas 2007) integrated in the Matisse toolbox gives a good precision for the PDE benchmark but very conservative results for the MCS and Helicopter benchmarks due to the appearance of ill-conditioned matrices in solving LMI and optimization problems.

[1] The prototype implementation and SpaceEx model files for the examples evaluated, both before and after order reduction, are available at: http://verivital.com/hyst/pass-order-reduction/.

Table 2 Benchmarks for the order-reduction abstraction method in which n is dimension of the system; m and p are the number of inputs and outputs respectively

Benchmark	Property	Initial set of states $X_0 = \{x_0 \in \mathbb{R}^n \mid lb(i) \le x_0(i) \le ub(i),\ 1 \le i \le n\}$	Input constraint $u = [u_1, \cdots, u_m]^T$	Safety specification $y = [y_1, \cdots, y_p]^T$
Motor control system (MCS)	$n = 8$	$lb(i) = ub(i) = 0,\ i = 2, 3, 4, 6, 7, 8,$	$u_1 \in [0.16,\ 0.3],$	unsafe region:
	$m = 2$	$lb(2) = 0.002,\ ub(2) = 0.0025,$	$u_2 \in [0.2,\ 0.4].$	$0.35 \le y_1 \le 0.4,$
	$p = 2$	$lb(3) = 0.001,\ ub(3) = 0.0015.$		$0.45 \le y_2 \le 0.6$
Helicopter (HELI) [7]	$n = 28$	$lb(i) = ub(i) = 0.1,\ i = 1, 4, 5, 6, 7,$	$u_i \in [-1,\ 1],$	unsafe region:
	$m = 6$	$lb(2) = lb(3) = 0.098, ub(2) = 0.11, ub(3) = 0.102,$	$1 \le i \le 6.$	$-1 \le y_1 \le 1,$
	$p = 2$	$lb(i) = ub(i) = 0,\ 8 \le i \le 28.$		$10 \le y_2$
Building model (BM) [4]	$n = 48$	$lb(i) = 0.0002,\ ub(i) = 0.00025,\ 1 \le i \le 10,$	$u_1 \in [0.8,\ 1].$	unsafe region:
	$m = 1$	$lb(25) = -0.0001,\ ub(25) = 0.0001,$		$0.008 \le y_1$
	$p = 1$	$lb(i) = ub(i) = 0,\ 11 \le i \le 48,\ i \ne 25.$		
Partial differential equation (PDE) [4]	$n = 84$	$lb(i) = 0,\ ub(i) = 0,\ 1 \le i \le 64$	$u_1 \in [0.5,\ 1].$	safe region:
	$m = 1$	$lb(i) = 0.001,\ ub(i) = 0.0015,\ 64 \le i \le 80,$		$y_1 \le 12$
	$p = 1$	$lb(i) = -0.002,\ ub(i) = -0.0015,\ 81 \le i \le 84.$		
International space station (ISS) [4]		$lb(i) = -0.0001,\ ub(i) = 0.0001,\ 1 \le i \le 270.$		Safe region:
				$-461y_1 + 887y_2 + 0.67 \le 0,$
	$n = 270$		$u_1 \in [0,\ 0.1],$	$-440y_1 - 898y_2 - 0.68 \le 0,$
	$m = 3$		$u_2 \in [0.8,\ 1],$	$-76.7y_1 + 997y_2 - 0.54 \le 0,$
	$p = 3$		$u_3 \in [0.9,\ 1].$	$898y_1 - 440y_2 - 0.89 \le 0,$
				$945y_1 + 326y_2 - 0.95 \le 0,$
				$-0.0005 \le y_3 \le 0.0005.$
FOM [4]	$n = 1006$	$lb(i) = -0.0001,\ ub(i) = 0.0001,\ 1 \le i \le 400$	$u_1 \in [-1,\ 1].$	safe region:
	$m = 1$	$lb(i) = 0.0002,\ ub(i) = 0.00025,\ 401 \le i \le 800,$		$y_1 \le 45$
	$p = 1$	$lb(i) = 0,\ ub(i) = 0,\ 801 \le i \le 1006.$		

Table 3 Experimental results obtained from different methods in which: k is the dimension of the output abstraction, δ is the precision, e_1 is the zero input response error, e_2 is the zero state response error, t is the error computing time (in second) and N is the number of simulations

	k	Bisim		Sim			Mixed bound				Theoretical bound			
		δ	$t(s)$	δ	$t(s)$	N	e_1	e_2	δ	$t(s)$	e_1	e_2	δ	$t(s)$
MCS	5	2.1	0.93	$\begin{pmatrix}0.0021\\0.047\end{pmatrix}$	0.17	2^2+2	$\begin{pmatrix}0.00049\\0.00062\end{pmatrix}$	$\begin{pmatrix}0.002\\0.047\end{pmatrix}$	$\begin{pmatrix}0.0025\\0.047\end{pmatrix}$	0.92	$\begin{pmatrix}0.0098\\0.0093\end{pmatrix}$	$\begin{pmatrix}0.53\\0.53\end{pmatrix}$	$\begin{pmatrix}0.54\\0.54\end{pmatrix}$	0.15
	4	1.5	0.64	$\begin{pmatrix}0.036\\0.047\end{pmatrix}$	0.19	2^2+2	$\begin{pmatrix}0.00086\\0.00062\end{pmatrix}$	$\begin{pmatrix}0.035\\0.047\end{pmatrix}$	$\begin{pmatrix}0.036\\0.047\end{pmatrix}$	0.9	$\begin{pmatrix}0.009\\0.009\end{pmatrix}$	$\begin{pmatrix}0.91\\0.91\end{pmatrix}$	$\begin{pmatrix}0.92\\0.92\end{pmatrix}$	0.15
HELI	20	0.84	17	$\begin{pmatrix}7.1e-05\\4.3e-05\end{pmatrix}$	0.6	2^4+6	$\begin{pmatrix}0.0072\\0.018\end{pmatrix}$	$\begin{pmatrix}5.7e-05\\3.2e-05\end{pmatrix}$	$\begin{pmatrix}0.0073\\0.018\end{pmatrix}$	35	$\begin{pmatrix}0.28\\0.95\end{pmatrix}$	$\begin{pmatrix}0.0017\\0.0017\end{pmatrix}$	$\begin{pmatrix}0.28\\0.95\end{pmatrix}$	0.49
	16	28	12	$\begin{pmatrix}0.00075\\0.0013\end{pmatrix}$	0.56	2^4+6	$\begin{pmatrix}0.0072\\0.018\end{pmatrix}$	$\begin{pmatrix}0.0007\\0.00087\end{pmatrix}$	$\begin{pmatrix}0.0079\\0.019\end{pmatrix}$	23	$\begin{pmatrix}0.28\\0.95\end{pmatrix}$	$\begin{pmatrix}0.029\\0.029\end{pmatrix}$	$\begin{pmatrix}0.3\\0.97\end{pmatrix}$	0.45
	10	160	8.1	$\begin{pmatrix}0.024\\0.038\end{pmatrix}$	0.55	2^4+6	$\begin{pmatrix}0.0085\\0.021\end{pmatrix}$	$\begin{pmatrix}0.021\\0.031\end{pmatrix}$	$\begin{pmatrix}0.03\\0.053\end{pmatrix}$	13	$\begin{pmatrix}0.27\\0.93\end{pmatrix}$	$\begin{pmatrix}1\\1\end{pmatrix}$	$\begin{pmatrix}1.3\\1.9\end{pmatrix}$	0.45
BM	25	0.0096	180	0.0051	22	$2^{11}+1$	0.013	$6.2e-05$	0.013	130	0.083	0.0072	0.09	1
	15	0.069	120	0.005	18	$2^{11}+1$	0.012	0.00044	0.013	58	0.078	0.084	0.16	0.97
	6	0.1	44	0.0058	14	$2^{11}+1$	0.011	0.00025	0.012	24	0.073	0.21	0.28	0.98
PDE	30	0.75	230	N/A	OOT	$2^{20}+1$	0.033	$5.6e-14$	0.033	1500	1	$5e-12$	1	1.7
	20	0.038	160	N/A	OOT	$2^{20}+1$	0.033	$3.5e-14$	0.033	890	1	$5.4e-12$	1	1.7
	10	0.086	55	N/A	OOT	$2^{20}+1$	0.033	$9.8e-13$	0.033	520	0.92	$2.7e-11$	0.92	1.7
	6	0.1	42	N/A	OOT	$2^{20}+1$	0.033	$3.5e-07$	0.033	370	0.89	$5.5e-06$	0.89	1.7

Table 3 (continued)

	k	Bisim		Sim			Mixed bound				Theoretical bound			
		δ	$t(s)$	δ	$t(s)$	N	e_1	e_2	δ	$t(s)$	e_1	e_2	δ	$t(s)$
ISS	25	N/A	OOT	N/A	OOT	$2^{270}+3$	N/A	$\begin{pmatrix} 2.1e-05 \\ 0.001 \\ 4.6e-05 \end{pmatrix}$	N/A	OOT	$\begin{pmatrix} 0.00043 \\ 0.00026 \\ 0.00026 \end{pmatrix}$	$\begin{pmatrix} 0.47 \\ 0.47 \\ 0.47 \end{pmatrix}$	$\begin{pmatrix} 0.47 \\ 0.47 \\ 0.47 \end{pmatrix}$	11
	10	N/A	OOT	N/A	OOT	$2^{270}+3$	N/A	$\begin{pmatrix} 2.4e-05 \\ 5.6e-05 \\ 9e-05 \end{pmatrix}$	N/A	OOT	$\begin{pmatrix} 0.00042 \\ 0.00022 \\ 0.00021 \end{pmatrix}$	$\begin{pmatrix} 1.7 \\ 1.7 \\ 1.7 \end{pmatrix}$	$\begin{pmatrix} 1.7 \\ 1.7 \\ 1.7 \end{pmatrix}$	12
FOM	20	N/A	OOT	N/A	OOT	$2^{800}+1$	N/A	$2.7e-07$	N/A	OOT	1.3	$1.1e-05$	1.3	48
	15	N/A	OOT	N/A	OOT	$2^{800}+1$	N/A	0.00021	N/A	OOT	1.3	0.0065	1.3	48
	10	N/A	OOT	N/A	OOT	$2^{800}+1$	N/A	0.1	N/A	OOT	1.3	2.2	3.5	48

Bisim column contains the results of approximate bisimulation approach proposed by Girard and Pappas (2007). **Sim** column are the results of the simulation-based approach proposed by Han and Krogh (2004)

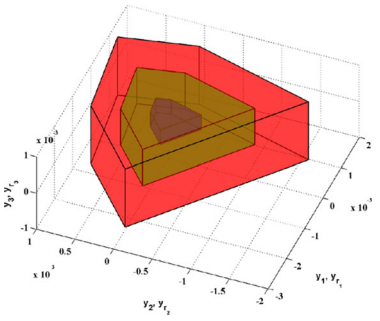
Combining Theorem 2 and simulation bound of e_2 (i.e. mixed bound) is more efficient as it produces very tight precisions for all benchmarks. In the case of high-dimensional systems ($n > 100$), the approximate bisimulation and simulation-based approaches give no result due to running out of time while our method can still be applied. We can see that Theorem 1 and Theorem 3 have small computation times while Theorem 2 and the approximate bisimulation approach require much more time to compute the precision.

Table 4 shows the computation cost of the verification process for the full-order benchmarks and their different output abstractions. The bounded times for running all SpaceEx models are set at $t_f = 20s$. As shown in the table, although using output abstraction does not help much to reduce the memory used in verification, it can help to reduce significantly the computation time. Moreover, output abstraction can be applied to check the safety of high-dimensional systems (e.g. PDE, ISS and FOM) that cannot be verified directly using existing verification tools. Next, we consider the whole process of using output abstraction to verify the safety of the international space station system.

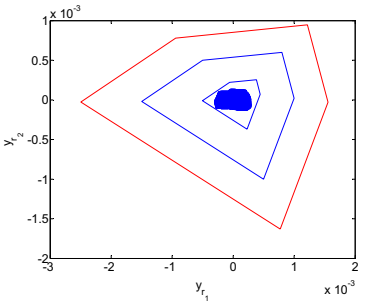
International space station (ISS) Verification for the full-order system with 270 state variables (denoted by M_{270}) may be difficult for existing verification tools. Our approach can help to verify safety of such high-dimensional system with a small computation cost. There are different output abstractions that can be used to verify whether the full-order system satisfies its safety requirements. In this paper, a 10-order output abstraction is chosen

Table 4 Computation cost for verification process of the full order original LTI system and its output abstractions using SpaceEx in which T_1 is the time for SpaceEx to compute the reach set of the output abstraction; T_2 is the time for obtaining the output abstraction; "Total Time" column states for the total time of verification process for the output abstraction, "Memory" column presents the memory used for computing reach set which is measure in kilobyte; time is measured in second

Benchmark	Full order system		Output abstraction				
	Time(s)	Memory(Kb)	k	$T_1(s)$	$T_2(s)$	Total time(s)	Memory(Kb)
Motor control system	27	3048	5	26	0.92	26.9	3044
			4	16.7	0.9	17.6	3044
Helicopter	287	3052	20	206	35	241	3052
			16	128	23	151	3048
			10	68	13	81	3048
Building model	893	3056	25	237.2	130	367.2	3048
			15	82.3	58	140.3	3044
			6	19.5	24	43.5	3040
Partial differential equation	OOT	N/A	30	725.6	1500	2225.6	3048
			20	310	890	1200	3048
			10	75.2	520	595.2	3040
			6	31.9	370	401.9	3040
International space station	OOT	N/A	25	254.3	11	265.3	3064
			10	72.8	12	84.8	3052
FOM model	OOT	N/A	20	95.4	48	143.4	3048
			15	56.2	48	104.2	3044
			10	34.8	48	82.8	3040



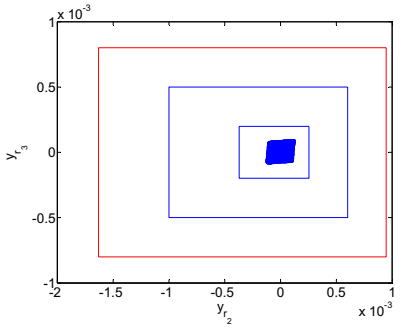
(a) Safety specification S_{270} of the full order ISS system which is the region inside the middle green polytopes and the transformed safety specifications of its 10-order output abstraction in which the safe region S^{δ}_{10} is inside the smallest blue polytopes and the unsafe region U^{δ}_{10} is outside the largest red polytope.



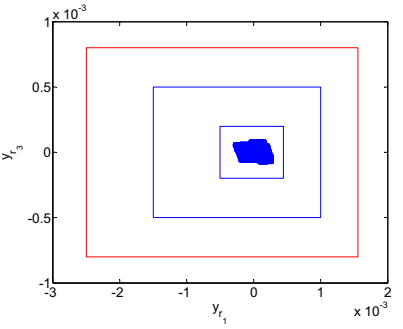
(b) Reachable set of (y_{r_1}, y_{r_2}) of the ISS's 10-order abstraction M^{δ}_{10} in the period of time $[0,\ 20s]$ and its safe region S^{δ}_{12} (inside the smallest blue polygon) and unsafe region U^{δ}_{12} (outside the red polygon).

Fig. 1 The transformed safety specifications of the 10-order output abstraction of ISS system and the its projected reachable set on the (y_{r_1}, y_{r_2}) plane

to check the safety of the full-order system. The precision $\delta = 10^{-3} \times [0.44,\ 0.28,\ 0.3]^T$ between the full-order system and its 10-order output abstraction (denoted by M^{δ}_{10}) is obtained from the theoretical bound of e_1 and the simulation bound of e_2. The safety specification of the full-order ISS system S_{270} is visualized by the region inside the middle blue polytope in Fig. 1a. The transformed safety specifications (safe and unsafe specifications) of the corresponding 10-order output abstraction respectively are the region inside the smallest blue polytope and the region outside the red polytope. Figures 1b, 2a and b present the



(a) Reachable set of (y_{r_2}, y_{r_3}) of the ISS's 10-order abstraction M^{δ}_{10} in the period of time $[0,\ 20s]$ and its safe region S^{δ}_{23} (inside the smallest blue polygon) and unsafe region U^{δ}_{23} (outside the red polygon).



(b) Reachable set of (y_{r_1}, y_{r_3}) of the ISS's 10-order abstraction M^{δ}_{10} in the period of time $[0,\ 20s]$ and its safe region S^{δ}_{13} (inside the smallest blue polygon) and unsafe region U^{δ}_{13} (outside the red polygon).

Fig. 2 The projected transformed safety specifications and reachable set of the 10th-order output abstraction of ISS system on the (y_{r_2}, y_{r_3}) and (y_{r_1}, y_{r_3}) planes

safety specification transformation and output reach set in the period of time $[0,\ 20s]$ computed by SpaceEx for the 10-order output abstraction on 2-dimension axes. In the figures, the regions inside the middle blue polygons are the 2-dimensional projections of the safety regions of the full-order system. The corresponding projected transformed safety and unsafe specifications S_{10}^{δ}, U_{10}^{δ} of the output abstraction are described by the regions inside the smallest blue polygons and the regions outside the red polygons respectively. The reach sets $R_{ij}^{\delta},\ i \neq j,\ 1 \leq i, j \leq 3$ for each pair output (y_{r_i}, y_{r_j}) of the abstraction M_{10}^{δ} are depicted by the solid blue regions. As shown in Fig. 1a, the reach set of the abstraction is completely inside its transformed safe set. Thus, it can be concluded that the full-order system M_{270} satisfies the safety requirement S_{270}. Therefore, the full-order system is safe. The conclusion about the safety of the full-order system can also be given using projection as follows. Due to the fact that one face of the polytopes defining the transformed safe set is aligned with one sub-coordinate axes (i.e. $y_{r_1} O y_{r_2}$), so for all (i, j), we have $R_{ij}^{\delta} \cap \neg S_{10}^{\delta} = \emptyset \Rightarrow M_{10}^{\delta} \vDash S_{10}^{\delta}$. Consequently, the full-order system is safe. It should be noted that the above relation is not true in general because we cannot conclude a set is a subset of another set by considering the relations of their projections on all sub-coordinate axes. The conclusion about safety of the ISS system is a special case as one face of the safe set is aligned with one sub-coordinate axes.

6 Conclusion and future work

We have proposed an approach to verify safety specifications for high-dimensional linear systems by verifying transformed safety specifications of a lower-dimensional output abstraction using existing hybrid system verification tools. By reducing the dimensionality, our method significantly reduces the time of reachability computations in the verification process.

There are two interesting directions for future work. First, the proposed method which only deals with stable linear systems can be extended for unstable linear systems. Second, our approach can also be extended to more general hybrid systems. The main idea is that the states in each location that are related to guards/invariants need to be declared as the outputs of that location. Then, the output abstraction for each location can be obtained. A new hybrid system is then constructed based on these output abstractions. he guards/invariants of the new hybrid system are obtained by transforming the former guards/invariants of the original hybrid system in the same manner of safety specifications transformation proposed in this paper. This approach may benefit from other notions of "similarity" between behaviors (executions) of systems such as discrepancy functions (Duggirala et al. 2013), or conformance degree (Abbas et al. 2014).

Acknowledgments The authors gratefully acknowledge the detailed feedback provided by the reviewers, which have helped improve this manuscript. The authors thank Dr. Andrew Sogokon of Vanderbilt University for carefully reading and providing feedback on the final draft of the manuscript. The material presented in this paper is based upon work supported by the National Science Foundation (NSF) under grant numbers CNS 1464311 and SHF 1527398, the Air Force Research Laboratory (AFRL) through contract number FA8750-15-1-0105, and the Air Force Office of Scientific Research (AFOSR) under contract numbers FA9550-15-1-0258 and FA9550-16-1-0246. The U.S. government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of AFRL, AFOSR, or NSF.

Appendix A

A.1 Proof of Theorem 1

Consider the uncontrolled augmented system $\dot{\bar{x}} = \bar{A}\bar{x}$, we have, $d(\bar{x}^T\bar{x})/dt = \bar{x}^T(\bar{A} + \bar{A}^T)\bar{x}$. It is easy to see that the controllability grammian and the observability grammian of the augmented system are the same (denoted as $\bar{\Sigma}$): $\bar{A}\bar{\Sigma} + \bar{\Sigma}\bar{A}^T + \bar{B}\bar{B}^T = 0$ and $\bar{A}^T\bar{\Sigma} + \bar{\Sigma}\bar{A} + \bar{C}^T\bar{C} = 0$. Combining two equations yields: $(\bar{A} + \bar{A}^T)\bar{\Sigma} + \bar{\Sigma}(\bar{A} + \bar{A}^T) = -\bar{B}\bar{B}^T - \bar{C}^T\bar{C}$. This Lyapunov equation implies that the real parts of all eigenvalues of $\bar{A}+\bar{A}^T$ are necessarily non-positive. Because $\bar{A}+\bar{A}^T$ is symmetric, its eigenvalues are real. Thus, these eigenvalues are either negative or zero. Note that $\bar{A}$ is asymptotically stable, hence $\bar{x}^T(t)\bar{x}(t) \to 0$ when $t \to \infty$ with all the initial state $\bar{x}(0)$. Combine all, we can conclude that $\bar{x}^T(t)\bar{x}(t)$ is monotonically converge to zero for any initial state $\bar{x}(0)$. Using the monotonic convergent property, the bound of the error e_1 satisfies: $\left\|e_1^i(t)\right\| = \left\|\bar{C}_i\bar{x}\right\| \leq \left\|\bar{C}_i\right\| \|\bar{x}\| \leq \left\|\bar{C}_i\right\| \|\bar{x}(0)\| \leq \left\|\bar{C}_i\right\| \cdot \sup_{x(0)\in X_0} \|\bar{x}(0)\|$.

A.2 Proof of Theorem 2

Consider the uncontrolled augmented system (i.e. $u = 0$), let $V(\bar{x}(t)) = \bar{x}(t)^T P\bar{x}(t)$, we have $\dot{V}(\bar{x}(t)) = \bar{x}(t)^T(\bar{A}^T P + P\bar{A})\bar{x}(t)$. Assume P_0 is the solution of the optimization problem in Theorem 2. Because of $(\bar{A}^T P_0 + P_0\bar{A}) < 0$, then $V(\bar{x}(t)) < V(\bar{x}(0) = \bar{x}(0)^T P_0\bar{x}(0)$. Note that $\left\|e_1^i(t)\right\|^2 = \bar{x}^T\bar{C}_i^T\bar{C}_i\bar{x},\ 1 \leq i \leq p$. Since we also have $\bar{C}_i^T\bar{C}_i \leq P_0$, the bound of the error satisfies $\left\|e_1^i(t)\right\| \leq \sqrt{\bar{x}(0)^T P_0\bar{x}(0)}$.

A.3 Proof of Lemma 1

From the definition of output abstraction, we have:

$$\alpha_{ij}y_{r_j} - |\alpha_{ij}|\delta_j \leq \alpha_{ij}y_j \leq \alpha_{ij}y_{r_j} + |\alpha_{ij}|\delta_j \Rightarrow \Gamma y_r + \overline{\Psi}_2 \leq \Gamma y + \Psi \leq \Gamma y_r + \overline{\Psi}_1.$$

Thus, $S(M_k^\delta)$ and $U(M_k^\delta)$ defined by Eq. 3 satisfy the safety relation (1), which completes the proof.

A.3.1 Proof of Lemma 2

Let $\bar{y} = E(y - a)$, $\bar{y}_r = E(y_r - a)$. We have:

$$\begin{aligned}(y-a)^T Q(y-a) &= \bar{y}^T\Lambda\bar{y} = \sum\nolimits_{i=1}^{p}\lambda_i\bar{y}_i^2,\\ (y_r-a)^T Q(y_r-a) &= \bar{y}_r^T\Lambda\bar{y}_r = \sum\nolimits_{i=1}^{p}\lambda_i\bar{y}_{r_i}^2.\end{aligned} \tag{5}$$

From the definition of output abstraction (Definition 1), it is easy to see that:

$$-\bar{\delta}_i \leq \bar{y}_i - \bar{y}_{r_i} = E(i,:)(y - y_r) \leq \bar{\delta}_i,\ \bar{\delta}_i = \sum\nolimits_{j=1}^{p}|\gamma_{ij}|\delta_j. \tag{6}$$

Using Eq. 6 and the Cauchy-Schwarz inequality yields:

$$\begin{aligned}
&\sum_{i=1}^{p} \lambda_i (\bar{y}_i - \bar{y}_{r_i})^2 \leq \Delta_R^2 = \sum_{i=1}^{p} \lambda_i \bar{\delta}_i^2, \\
&\sum_{i=1}^{p} 2\lambda_i \bar{y}_{r_i} (\bar{y}_i - \bar{y}_{r_i}) \leq 2\Delta_R \sqrt{\sum_{i=1}^{p} \lambda_i \bar{y}_{r_i}^2}, \\
&\sum_{i=1}^{p} 2\lambda_i \bar{y}_i (\bar{y}_{r_i} - \bar{y}_i) \leq 2\Delta_R \sqrt{\sum_{i=1}^{p} \lambda_i \bar{y}_i^2}.
\end{aligned} \tag{7}$$

From Eq. 7, the following inequalities are true:

$$\sum_{i=1}^{p} \lambda_i \bar{y}_i^2 \leq \left(\sqrt{\sum_{i=1}^{p} \lambda_i \bar{y}_{r_i}^2} + \Delta_R \right)^2, \quad \sum_{i=1}^{p} \lambda_i \bar{y}_{r_i}^2 \leq \left(\sqrt{\sum_{i=1}^{p} \lambda_i \bar{y}_i^2} + \Delta_R \right)^2. \tag{8}$$

From Eqs. 5 and 8, we have:

$$\begin{aligned}
&\sqrt{(y-a)^T Q (y-a)} \leq \sqrt{(y_r - a)^T Q (y_r - a)} + \Delta_R, \\
&\sqrt{(y-a)^T Q (y-a)} \geq \sqrt{(y_r - a)^T Q (y_r - a)} - \Delta_R.
\end{aligned} \tag{9}$$

Using Eq. 9, we can conclude that $S(M_k^{\delta})$ and $S(M_k^{\delta})$ defined in Lemma 2 satisfy the safety relation (1), which completes the proof.

References

Abbas H, Mittelmann H, Fainekos G (2014) Formal property verification in a conformance testing framework. In: Twelfth ACM/IEEE international conference on formal methods and models for codesign (memocode), 2014. IEEE, pp 155–164

Antoulas AC, Sorensen DC, Gugercin S (2001) A survey of model reduction methods for large-scale systems. Contemp Math 280:193–219

Bak S, Bogomolov S, Johnson TT (2015) HyST: a source transformation and translation tool for hybrid automaton models. In: 18th international conference on hybrid systems: computation and control. ACM, Seattle, Washington

Chahlaoui Y, Van Dooren P (2005) Benchmark examples for model reduction of linear time-invariant dynamical systems. Dimension Reduction of Large-Scale Systems, pp 379–392. Springer

De Moura L, Bjørner N (2008) Z3: an efficient smt solver. In: International conference on tools and algorithms for the construction and analysis of systems. Springer, pp 337–340

Duggirala PS, Mitra S, Viswanathan M (2013) Verification of annotated models from executions. In: Proceedings of the eleventh acm international conference on embedded software. Emsoft '13. IEEE Press, Piscataway, NJ, USA. ISBN 978-1-4799-1443-2

Frehse G, Le Guernic C, Donzé A, Cotton S, Ray R, Lebeltel O, Ripado R, Girard A, Dang T, Maler O (2011) Spaceex: scalable verification of hybrid systems. In: Computer aided verification. Springer, pp 379–395

Girard A, Pappas GJ (2007) Approximate bisimulation relations for constrained linear systems. Automatica 43(8):1307–1317

Girard A, Julius AA, Pappas GJ (2008) Approximate simulation relations for hybrid systems. Discrete Event Dynamic Systems 18(2):163–179

Girard A, Pappas GJ et al. (2006) Approximate bisimulation for a class of stochastic hybrid systems. In: American control conference, 2006, vol 6. IEEE

Han Z (2005) Formal verification of hybrid systems using model order reduction and decomposition. PhD diss PhD thesis, Dept. of ECE, Carnegie Mellon University

Han Z, Krogh B (2004) Reachability analysis of hybrid control systems using reduced-order models. In: American control conference, 2004. Proceedings of the 2004, vol 2. IEEE, pp 1183–1189

Islam MdA, Murthy A, Bartocci E, Cherry EM, Fenton FH, Glimm J, Smolka SA, Grosu R (2015) Model-order reduction of ion channel dynamics using approximate bisimulation. Theor Comput Sci 599:34–46. doi:10.1016/j.tcs.2014.03.018. Advances in Computational Methods in Systems Biology

Julius AA (2006) Approximate abstraction of stochastic hybrid automata. In: Hybrid systems: computation and control. Springer, pp 318–332

Moore B (1981) Principal component analysis in linear systems: controllability, observability, and model reduction. IEEE Trans Autom Control 26(1):17–32

Nesterov Y, Nemirovskii A, Ye Y (1994) Interior-point polynomial algorithms in convex programming, vol 13. SIAM

Obinata G, Anderson BDO (2012) Model reduction for control system design. Springer

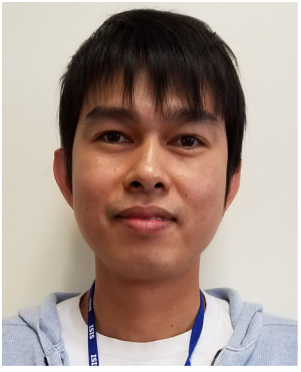
Pappas GJ (2003) Bisimilar linear systems. Automatica 39(12):2035–2047

Tabuada P, Pappas GJ (2004) Bisimilar control affine systems. Syst Control Lett 52(1):49–58

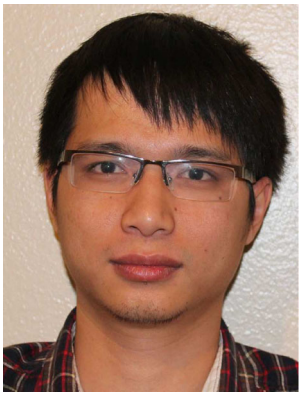
Tanner HG, Pappas GJ (2003) Abstractions of constrained linear systems. In: American control conference, 2003. Proceedings of the 2003, vol 4. IEEE, pp 3381–3386

van der Schaft A (2004) Equivalence of dynamical systems by bisimulation. IEEE Trans Autom Control 49(12):2160–2172

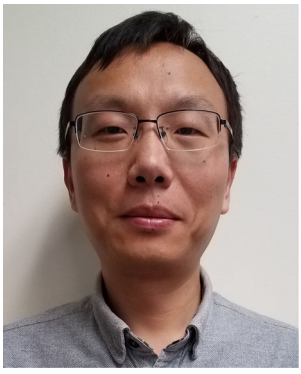
Vandenberghe L, Boyd S (1994) Positive definite programming. Mathematical Programming: State of the Art



Hoang-Dung Tran is a PhD student in Computer Science in Vanderbilt University. He earned his BSEE from the Gifted Undergraduate Program in Automatic Control at Ho Chi Minh University of Technology, Vietnam in 2007 and Master's Degree in Control Science at the Huazhong University of Science and Technology, China in 2013. Mr. Tran has been a research assistant in the Verification and Validation for Intelligent and Trustworthy Autonomy Laboratory (VeriVITAL) supervised by Prof. Taylor T. Johnson since 2015. His research interests are formal verification for cyber-physical systems, autonomous multi-agents systems, and control theory.



Luan Viet Nguyen is a Ph.D. student in Computer Science and Engineering at the University of Texas at Arlington. He graduated with honors from MSc program in May 2013 at the Catholic University of America, where he had previously earned his BSEE in December 2012. He is currently working at the Verification and Validation for Intelligent and Trustworthy Autonomy Laboratory, under the supervision of Prof. Taylor T. Johnson. His main research interests are cyber-physical system (CPS), distributed system, formal method, and control theory, with a focus on conducting reachability analysis, specification mining, and building software tools to enhance the reliability of CPS.



Weiming Xiang received his Ph.D. degree in Southwest Jiaotong University, Chengdu, China in 2014, and his Ph.D. dissertation was awarded the Outstanding Doctoral Dissertation of Southwest Jiaotong University in 2014. During May 2015 to October 2015, he held a position of Research Associate in the Department of Mechanical Engineering at The University of Hong Kong, Hong Kong. During November 2015 to August 2016, he held a position of Postdoctoral Research Associate in the Department of Computer Science and Engineering at the University of Texas at Arlington, Texas, USA. After that, he joined Department of Electrical Engineering and Computer Science at Vanderbilt University in Nashville, USA, working as a Postdoctoral Researcher until now. Dr. Xiang's current research interests are in developing theories of switched system and control, robust control and filtering, nonlinear system and control, fuzzy system and applying them to complex real-world systems. He is an IEEE member and currently on the Editorial Board of Neurocomputing.

Taylor T. Johnson is an Assistant Professor of Electrical Engineering and Computer Science (EECS) at Vanderbilt University (since August 2016), where he directs the Verification and Validation for Intelligent and Trustworthy Autonomy Laboratory (VeriVITAL) and is a Senior Research Scientist in the Institute for Software Integrated Systems. Dr. Johnson was previously an Assistant Professor of Computer Science and Engineering (CSE) at the University of Texas at Arlington (September 2013 to August 2016). Dr. Johnson is a 2016 recipient of the AFOSR Young Investigator Research Program (YIP) award. Dr. Johnson earned a PhD in Electrical and Computer Engineering (ECE) at the University of Illinois at Urbana-Champaign in 2013, an MSc in ECE at Illinois in 2010, and a BSEE in ECE from Rice University in 2008. Dr. Johnson's research focus is developing formal verification techniques and software tools for cyber-physical systems (CPS) with goals of improving safety, reliability, and security. Dr. Johnson has published over four-dozen papers on these verification and validation methods and their applications across domain areas such as power and energy systems, aerospace, transportation systems, and robotics, two of which were recognized with best paper awards, from the IEEE and IFIP, respectively. Dr. Johnson gratefully acknowledges the support of his group's research by AFRL, AFOSR, ARO, NSF (CISE CCF/SHF, CNS/CPS; ENG ECCS/EPCN), NVIDIA, ONR, Toyota, and USDOT.