Automatica 78 (2017) 287–296

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Construction of parametric barrier functions for dynamical systems using interval analysis☆

Adel Djaballah [a], Alexandre Chapoutot [a], Michel Kieffer [b], Olivier Bouissou [c]

[a] U2IS, ENSTA ParisTech, Université Paris-Saclay, 828 bd des Maréchaux, 91762 Palaiseau Cedex, France

[b] L2S, CNRS, Supélec, Univ. Paris-Sud 91192 Gif-sur-Yvette Cedex and Institut Universitaire de France, 75005 Paris, France

[c] CEA Saclay Nano-INNOV Institut CARNOT, 91191 Gif-sur-Yvette Cedex, France

ARTICLE INFO

Article history:
Received 14 April 2015
Received in revised form
21 July 2016
Accepted 30 November 2016
Available online 27 January 2017

Keywords:
Formal verification
Dynamic systems
Intervals
Constraint satisfaction problem

ABSTRACT

Recently, barrier certificates have been introduced to prove the safety of continuous or hybrid dynamical systems. A barrier certificate needs to exhibit some barrier function, which partitions the state space in two subsets: the safe subset in which the state can be proved to remain and the complementary subset containing some unsafe region. This approach does not require any reachability analysis, but needs the computation of a valid barrier function, which is difficult when considering general nonlinear systems and barriers. This paper presents a new approach for the construction of barrier functions for nonlinear dynamical systems. The proposed technique searches for the parameters of a parametric barrier function using interval analysis. Complex dynamics with bounded perturbations can be considered without needing any relaxation of the constraints to be satisfied by the barrier function.

1. Introduction

Formal verification aims at proving that a certain behavior or property is fulfilled by a system. Verifying, *e.g.*, the safety property for a system consists in ensuring that it will never reach a dangerous or an unwanted configuration. Safety verification is usually translated into a reachability analysis problem (Asarin, Bournez, Dang, & Maler, 2000; Chutinan & Krogh, 1999; Frehse et al., 2011; Sun, Ge, & Lee, 2002; Tiwari, 2003). Starting from an initial region, a system must not reach some unsafe region. Different methods have been considered to address this problem. One may explicitly compute the reachable region and determine whether the system reaches the unsafe region (Gulwani & Tiwari, 2008). An alternative idea is to compute an invariant for the system, *i.e.*, a region in which the system is guaranteed to stay (Chutinan & Krogh, 1999). This paper considers a class of invariants determined by *barrier functions*.

A barrier function (Prajna, 2006; Prajna & Jadbabaie, 2004) partitions the state space and isolates an unsafe region from the part of the state space containing the initial region. In Prajna and Jadbabaie (2004), polynomial barriers are considered for polynomial systems and semi-definite programming is used to find satisfying barrier functions. Our aim is to extend the class of considered problems to non-polynomial systems and to non-polynomial barriers. This paper focuses on continuous-time systems.

The design of a barrier function is formulated as a quantified constraints satisfaction problem (QCSP) (Benhamou & Goualard, 2000; Ratschan, 2006). Interval analysis is then used to find the parameters of a barrier function such that the QCSP is satisfied. More specifically, the algorithm presented in Jaulin and Walter (1996) for robust controller design is adapted and supplemented with some of the pruning schemes found in Chabert and Jaulin (2009) to solve the QCSP associated to the barrier function design.

The paper is organized as follows. Section 2 introduces some related work. Section 3 defines the notion of barrier functions and formulates the design of barrier functions as a QCSP. Section 4 presents the framework developed to solve the QCSP. Design examples are presented in Section 5. Section 6 concludes the work.

In what follows, small italic letters x represent real variables while real vectors $\mathbf{x}$ are in bold. Intervals $[x]$ and interval vectors (boxes) $[\mathbf{x}]$ are represented between brackets. We denote by $\mathbb{IR}$

☆ This research was partially supported by Labex DigiCosme (project ANR-11-LABEX-0045-DIGICOSME) operated by ANR as part of the program "Investissement d'Avenir" Idex Paris-Saclay (ANR-11-IDEX-0003-02), by the ANR INS Project CAFEIN (ANR-12-INSE-0007), and by the iCODE Institute, research project of the IDEX Paris-Saclay. The material in this paper was partially presented at the 53rd IEEE Conference on Decision and Control, December 15–17, 2014, Los Angeles, CA, USA. This paper was recommended for publication in revised form by Associate Editor Zhihua Qu under the direction of Editor Andrew R. Teel.

E-mail addresses: adel.djaballah@ensta-paristech.fr (A. Djaballah), alexandre.chapoutot@ensta-paristech.fr (A. Chapoutot), michel.kieffer@lss.supelec.fr (M. Kieffer), olivier.bouissou@cea.fr (O. Bouissou).

http://dx.doi.org/10.1016/j.automatica.2016.12.013
0005-1098/

the set of closed intervals over $\mathbb{R}$, the set of real numbers. Data structures or sets $\mathcal{S}$ are in upper-case calligraphic. The derivative of a function x with respect to time t is denoted by $\dot{x}$.

2. Related work

The verification of the safety properties for dynamical systems has been an active field of research in the last years. This related work review focuses on methods involving the computation of *invariants* for dynamical systems. Alternative methods based on the computation of reachable sets are described in Chen, Abrahám, and Sankaranarayanan (2012), Frehse et al. (2011) and in the references therein.

An invariant is a part of the state space in which the state of a dynamical system can be proved to remain. Invariants are very useful to prove the safety of dynamical systems. If an invariant does not contain any unsafe regions, then the dynamical system is safe. Methods to characterize invariants have been intensively studied for linear and polynomial dynamics but significant work has still to be done for non-linear dynamical systems.

Using ideas from the community interested in *hybrid systems*, a set of methods has been defined to compute invariants for various classes of systems, for example for linear or affine systems (Tiwari, 2003) or for polynomial systems (Gulwani & Tiwari, 2008; Kapinski, Deshmukh, Sankaranarayanan, & Arechiga, 2014; Sankaranarayanan, Sipma, & Manna, 2004; Yang, Lin, & Wu, 2015). These methods introduce a candidate parametric function, which parameter vector has to be adjusted to define an invariant of the considered dynamical system. Various techniques are then employed to determine satisfying parameter vectors. For example, in Sankaranarayanan et al. (2004), the theory of ideals over polynomials and Gröbner bases are used to define constraints to be satisfied by the parameter vector of interest. These constraints are then solved numerically using tools such as those introduced in Collins and Hong (1991). In Gulwani and Tiwari (2008), quantified parametric polynomial constraints are considered. Then, satisfying parameter vectors are found using Farkas' Lemma and solvers from sat-modulo theory (Barrett & Tinelli, 2017). Sum-of-Squares (SoS) polynomials are used in Kapinski et al. (2014). The design involves various system simulations and selection of candidate parameter vectors using linear programming. A final validation of the selected parameter vectors is then performed with Mathematica and using interval analysis with dReal (Gao, Kong, & Clarke, 2013). Note that our algorithm presented in Section 4.3 could also be used as validation method for the approach presented in Kapinski et al. (2014). Bilinear SoS programming is considered in Yang et al. (2015).

An alternative way to find such an invariant is by considering tools such as Lyapunov functions to prove the stability properties of dynamical systems (Genesio, Tartaglia, & Vicino, 1985). For example, Parrilo (2003) considers parametric functions to find a Lyapunov function for a system with polynomial dynamics formed by SoS polynomials and employs semidefinite programming (SdP) for the parameter synthesis. In Ratschan and She (2006), Lyapunov functions are designed via a branch-and-relax approach and linear programming to solve the induced constraints. In Goubault, Jourdan, Putot, and Sankaranarayanan (2014) Darboux polynomials are used to design specific forms of Lyapunov functions involving rational functions, logarithmic, and exponential terms. Similar invariants have been also considered in Rebiha, Matringe, and Moura (2012).

Safety properties may also be directly verified in the design phase, instead of being verified *a posteriori*, as done in the previous approaches. In Platzer (2007, 2010), theorem-proving approaches are employed using symbolic–numeric techniques to synthesize invariants for differential (continuous and hybrid) systems. In particular, quantifier elimination techniques are intensively used and more recently a combination with the approach presented in Kapinski et al. (2014) has been considered in Aréchiga, Kapinski, Deshmukh, Platzer, and Krogh (2015). Alternatively, techniques searching for *barrier certificates* aim at determining a parametric function, called *barrier*, defining an hyper-surface in the state-space which is never crossed by the dynamics of the system, see Dai, Gan, Xia, and Zhan (2016), Prajna (2005), Prajna and Jadbabaie (2004), Prajna and Rantzer (2005) and Sloth, Pappas, and Wisniewski (2012). A parameter vector for this barrier has to be found such that the barrier separates the part of the state-space in which the initial state belongs from the unsafe region. In Prajna (2005), Prajna and Jadbabaie (2004) and Prajna and Rantzer (2005), polynomial dynamics and barrier functions are considered and parameters are designed with SdP, which require some relaxation to obtain a convex design problem. In Dai et al. (2016), two candidate functions are combined to define more sophisticated barriers, which parameters are again found via SdP. In Sloth et al. (2012), linear matrix inequalities and SoS are used to generate the barrier functions for hybrid dynamical systems with polynomial dynamics.

Our work follows this approach for non-linear and possibly non-polynomial continuous-time dynamical systems with bounded perturbations and uses interval analysis for the barrier parameter vector search phase.

3. Formulation

This section recalls the safety characterization introduced in Prajna and Jadbabaie (2004) for continuous-time systems using barrier functions.

3.1. Safety for continuous-time systems

Consider the autonomous continuous-time perturbed dynamical system

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{d}), \tag{1}$$

where $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^n$ is the state vector and $\mathbf{d} \in \mathcal{D}$ is a constant and bounded disturbance. The set of possible initial states at $t = 0$ is denoted $\mathcal{X}_0 \subset \mathcal{X}$. There is some unsafe subset $\mathcal{X}_u \subseteq \mathcal{X}$ that shall not be reached by the system, given any $\mathbf{x}_0 \in \mathcal{X}_0$ at time $t = 0$ and any $\mathbf{d} \in \mathcal{D}$. We assume that classical hypotheses (see, *e.g.,* Bellman & Cooke, 1963) on f are satisfied so that (1) has a unique solution $\mathbf{x}(t, \mathbf{x}_0, \mathbf{d}) \in \mathcal{X}$ for any given initial value $\mathbf{x}_0 \in \mathcal{X}_0$ at time $t = 0$ and any $\mathbf{d} \in \mathcal{D}$.

Definition 1. The dynamical system (1) is *safe* if $\forall \mathbf{x}_0 \in \mathcal{X}_0, \forall \mathbf{d} \in \mathcal{D}$ and $\forall t \geqslant 0, \mathbf{x}(t, \mathbf{x}_0, \mathbf{d}) \notin \mathcal{X}_u$.

3.2. Barrier certificates

A way to prove that (1) is safe is by the barrier certificate approach introduced in Prajna and Jadbabaie (2004). A barrier is a differentiable function $B : \mathcal{X} \to \mathbb{R}$ that partitions the state space $\mathcal{X}$ into $\mathcal{X}_-$ where $B(\mathbf{x}) \leqslant 0$ and $\mathcal{X}_+$ where $B(\mathbf{x}) > 0$ such that $\mathcal{X}_0 \subseteq \mathcal{X}_-$ and $\mathcal{X}_u \subseteq \mathcal{X}_+$. Moreover, B has to be such that $\forall \mathbf{x}_0 \in \mathcal{X}_0, \forall \mathbf{d} \in \mathcal{D}, \forall t \geqslant 0, B(\mathbf{x}(t, \mathbf{x}_0, \mathbf{d})) \leqslant 0$.

Proving that $B(\mathbf{x}(t, \mathbf{x}_0, \mathbf{d})) \leqslant 0$ requires an evaluation of the solution of (1) for all $\mathbf{x}_0 \in \mathcal{X}_0$ and $\mathbf{d} \in \mathcal{D}$. Alternatively, Prajna and Jadbabaie (2004) provides some sufficient conditions a barrier function has to satisfy to prove the safety of a dynamical system, see Theorem 1.

Theorem 1. *Consider the dynamical system (1) and the sets $\mathcal{X}$, $\mathcal{D}$, $\mathcal{X}_0$ and $\mathcal{X}_u$. If there exists a function $B : \mathcal{X} \to \mathbb{R}$ such that*