Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 50-1 (2017) 7205–7210

Formal Controller Synthesis via Genetic Programming

C.F. Verdier * M. Mazo, Jr. *

* *Department of Delft Center of Systems and Control, Delft University of Technology, 2628 CD Delft, The Netherlands (e-mail: {c.f.verdier, m.mazo}@tudelft.nl).*

Abstract: This paper presents an automatic controller synthesis method for nonlinear systems with reachability and safety specifications. The proposed method consists of genetic programming in combination with an SMT solver, which are used to synthesize both a control Lyapunov function and the modes of a switched state feedback controller. The resulting controller consists of a set of analytic expressions and a switching law based on the control Lyapunov function, which together guarantee the imposed specifications. The effectiveness of the proposed approach is shown on a 2D pendulum.

Keywords: Formal methods, Lyapunov methods, Genetic programming

1. INTRODUCTION

Complex controller specifications of modern cyber-physical systems can often be formulated as (linear) temporal properties, i.e. propositions that are qualified in terms of time. In this paper we limit ourselves to a very specific temporal property, i.e. the *reach and stay while stay* (RSWS) property: all trajectories starting in an initial set I *eventually* reach and stay within the goal set G, while *always* staying in the safe set S. The aim of this paper is to automate the synthesis of switched state feedback controllers for nonlinear systems, such that the RSWS specification is met.

Two popular correct-by-design controller synthesis approaches for nonlinear systems with temporal specifications are 1) abstraction and simulation, and 2) control Lyapunov functions (CLFs) and control barrier functions (CBFs). The first approach consists of finding a symbolic abstraction of the system that (bi)simulates the original system, for which it is easier to construct and verify a controller to satisfy temporal logic specifications (Tabuada (2009)). Tools implementing this methodology for nonlinear systems are e.g. PESSOA (Mazo Jr et al. (2010)), SCOTS (Rungger and Zamani (2016)), and CoSyMa (Mouelhi et al. (2013)). Drawbacks of these methods are that they require a discretization of the state space and the controllers often take the form of enormous tables, hence these methods suffer from the curse of dimensionality.

Control Lyapunov functions (CLF) (Artstein (1983)) and control barrier functions (CBF) (Wieland and Allgöwer (2007)) are design tools for stabilization and safety respectively. The benefit of these methods is that there is no need to compute the exact solution of the system. Attempts to unify CLFs with CBFs can be found in e.g. Romdlony and Jayawardhana (2016) and Xu et al. (2015). A popular approach to automatic synthesis of (control) Lyapunov functions and (control) barrier functions is to pose the problem as a sum of squares (SOS) problem, which reduces it to a convex optimization problem, see e.g. Papachristodoulou and Prajna (2002). This approach is restricted to polynomial systems, although extensions exist such that some non-polynomial functions can be used by recasting them as polynomials, see e.g. Chesi (2009) and Hancock and Papachristodoulou (2013). Nevertheless, polynomial Lyapunov functions can be too restrictive. As shown in Ahmadi et al. (2011), if a polynomial system is globally asymptotically stable, there might exist no polynomial Lyapunov function.

To overcome the limitations of the two discussed approaches, we propose to use genetic programming (GP). GP is an evolutionary algorithm capable of evolving encoded representations of symbolic functions, until a satisfactory solution is found (Koza (1992)). The evolution is driven by a fitness function, which scores solutions on how well they satisfy desired specifications. GP distinguishes itself from other optimization methods in that it is able to search over the function space, rather than over a parameter space. Due to this nature, genetic programming (and variants) have been used to synthesize Lyapunov functions, see e.g. Grosman and Lewin (2009), and controllers, see e.g. Koza et al. (2003), Sekaj and Perkacz. (2007), Diveev and Shmalko (2015), and Chen and Lu (2011). In these works, fitness is based on specific samples and/or simulations, hence no formal guarantee can be given on the behavior of the system, other than for the specific test cases. In this work we propose the combination of genetic programming and a Satisfiability Modulo Theories (SMT) solver (Barrett et al. (2009)), which uses a combination of background theories to determine whether a first-order logic formula can be satisfied or not. This solver is used to provide formal guarantees on the behavior of the system.

In our approach, the used control strategy is a switching law that switches between different controller modes based on a CLF. Genetic programming is used to automatically

⋆ This work is supported by NWO Domain TTW, the Netherlands, under the project CADUSY- TTW#13852.

10.1016/j.ifacol.2017.08.1362

generate both candidate CLFs and the controller modes. Subsequently, the candidate solutions are verified using the SMT solver. In this paper, the SMT solver dReal is used, which is capable of providing formal guarantees on the satisfiability of nonlinear inequalities over the real numbers (Gao et al. (2013)). By using GP, we allow ourselves to search for solutions that include non-polynomial functions. Furthermore, as opposed to abstraction methods, the synthesized controllers are expressed as analytic expressions, that are in general more compact than a binary decision diagram (BDD) or a lookup table. Finally, the proposed method provides formal guarantees on stability and safety, as opposed to previous attempts using GP.

A similar approach is found in Ravanbakhsh and Sankaranarayanan (2015). Here, counter-example guided synthesis is used to synthesize a CLF for a switched system with a *reach-while-stay* specification. The verification is also done using dReal. However, only the controller modes are prefixed and only the CLF is synthesized.

To the best knowledge of the authors, this is the first work combining genetic programming and formal verification for controller synthesis. Furthermore, a special CLF is designed for the RSWS specification and such that the verification is decidable.

2. PROBLEM DEFINITION

Notation: Given a set A, let us denote the boundary as ∂A and the interior as $int(A)$. The Euclidean norm is denoted by $\|\cdot\|$ and the natural logarithm by $\ln(\cdot)$. The temporal logic operators *always* and *eventually* are denoted by $\Box$ and $\Diamond$ respectively. The predicate defining set A is denoted by ϕ_A. A system satisfies $\Box\phi_A$ if and only if $\forall t \geq 0, \xi(t) \in A$ and a system satisfies $\Diamond\phi_A$ if and only if $\exists t \geq 0, \xi(t) \in A$.

In this work we consider the class of nonlinear dynamical systems described by

$$\begin{cases} \dot{\xi}(t) &= f(\xi(t), u(t)) \\ \xi(0) &\in I \end{cases} \tag{1}$$

where I is a compact set and the variables $\xi(t) \in \mathbb{R}^n$ and $u(t) \in \mathbb{R}^m$ denote the state and input respectively.

The controller is designed such that the composition of the system (1) and control input $u(t)$ for $t \geq 0$ results in a system that satisfies

S1) *Reach and stay while stay (RSWS)*: given a compact safe set S, all trajectories starting from the compact initial set $I \subset int(S)$ eventually reach and stay in the compact goal set $G \subset int(S)$, while staying within the safe set S. This corresponds to the temporal logic formula $\Box\phi_S \wedge \Diamond\Box\phi_G$.

S2) There occurs no Zeno behavior.

We say a there occurs no Zeno behavior if there are no infinitely many switches in a finite time interval.

This paper addresses the following problem:

Problem 1. Given the compact sets S, I, G and system (1), synthesize a control law $u(t)$ such that specifications S1) and S2) are guaranteed.

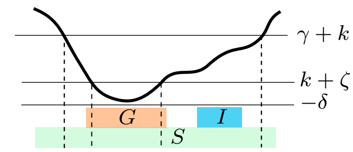

Fig. 1. Example of a RSWS CLF.

3. CONTROL STRATEGY

Consider the index set $Q = \{1, \ldots, M\}$, controller mode $q \in Q$ and vector fields $H = \{h_q(\xi(t)) | q \in Q\}$ from $\mathbb{R}^n$ to $\mathbb{R}^m$. In this work, we consider controllers of the form

$$u(t) = h_q(\xi(t)). \tag{2}$$

A switching law based on a CLF determines the next mode $q(\xi(t^+))$. The CLF is designed such that in combination with the switching law the desired safety and reachability specifications of S1) are enforced. This specific CLF is referred to as the RSWS CLF.

3.1 RSWS control Lyapunov function

Classical (control) Lyapunov functions satisfy a "tight" inequality, i.e. $(\forall x \in D) V(x) \geq 0$ and $\exists x \in D$ such that $V(x) = 0$, where $\{0\} \subseteq D$, see e.g. Khalil (2002). Similarly, the first derivative also satisfies a tight inequality. As stated in Gao et al. (2012), satisfiability of inequalities over the reals for transcendental functions is not decidable, hence they introduced the δ-decision problem, which is decidable. However, the δ-decision problem can be problematic for tight inequalities, as will be shown in Section 5. To circumvent the occurrence of tight inequalities, we introduce a perturbation variable δ in our definition of the RSWS CLF, yielding a more general CLF-like function, coined the *relaxed RSWS CLF*. The term relaxed is used to indicate that the bounds are picked to be more conservative compared to the nominal RSWS CLF (i.e. $\delta = 0$).

Let us denote the Lie derivative of $g(x)$ along the flow of $f(x, h_q(x))$ as $L_{f_q} = \frac{\partial g(x)}{\partial x} f(x, h_q(x)))$.

Definition 2. (Relaxed RSWS control Lyapunov function). A function $V \in \mathcal{C}^2(S, \mathbb{R})$ is a relaxed RSWS control Lyapunov function w.r.t. the compact sets (S, I, G) and system (1), if there exists real numbers $\alpha, \beta, \gamma, \varepsilon, \zeta > 0$, and $\delta \geq 0$, such that

$$\begin{array}{ll} V(x) \geq k + \zeta & \forall x \in S \backslash G \\ V(x) \geq -\delta & \forall x \in G \\ V(x) > \gamma + k & \forall x \in \partial S \\ V(x) \leq \gamma + k & \forall x \in I \\ \exists q \in Q \text{ x.t. } \dot{V}_q(x) \leq -\alpha V(x) + \delta & \forall x \in S \\ \ddot{V}_q(x) \leq \varepsilon & \forall q \in Q, \forall x \in S \end{array} \tag{3}$$

where $\dot{V}_q(x) = L_{f_q} V$, $\ddot{V}_q(x) = L_{f_q} \dot{V}_q(x)$ and

$$k = \beta + \frac{\delta}{\alpha}. \tag{4}$$

Remark 3. Note that the relaxed control Lyapunov function is not a CLF in the strict sense if $\delta > 0$, as it and its time derivative are not necessarily positive definite and negative definite respectively.

To illustrate the first four conditions, an example RSWS CLF is shown in Figure 1. For the sake of brevity, we use CLF to refer to the relaxed RSWS CLF throughout this paper.