

A Class of Control Certificates to Ensure Reach-While-Stay for Switched Systems

Hadi Ravanbakhsh and Sriram Sankaranarayanan

Department of Computer Science
University of Colorado, Boulder
Boulder, CO, USA

firsname.lastname@colorado.edu

In this article, we consider the problem of synthesizing switching controllers for temporal properties through the composition of simple primitive reach-while-stay (RWS) properties. Reach-while-stay properties specify that the system states starting from an initial set I , must reach a goal (target) set G in finite time, while remaining inside a safe set S . Our approach synthesizes switched controllers that select between finitely many modes to satisfy the given RWS specification. To do so, we consider control certificates, which are Lyapunov-like functions that represent control strategies to achieve the desired specification. However, for RWS problems, a control Lyapunov-like function is often hard to synthesize in a simple polynomial form. Therefore, we combine control barrier and Lyapunov functions with an additional compatibility condition between them. Using this approach, the controller synthesis problem reduces to one of solving quantified nonlinear constrained problems that are handled using a combination of SMT solvers. The synthesis of controllers is demonstrated through a set of interesting numerical examples drawn from the related work, and compared with the state-of-the-art tool SCOTS. Our evaluation suggests that our approach is computationally feasible, and adds to the growing body of formal approaches to controller synthesis.

1 Introduction

The problem of synthesizing switching controllers for reach-while-stay (RWS) specifications is examined in this article. RWS properties are an important class, since we may decompose more complex temporal specifications into a sequence of RWS specifications [11]. The plant model is a switched system that consists of finitely many (controllable) modes, and the dynamics for each mode are specified using ODEs. Furthermore, we consider nonlinear ODEs for each mode, including rational, trigonometric, and exponential functions. The goal of the controller is to switch between the appropriate modes, so that the resulting closed loop traces satisfy the specification.

RWS properties specify that a goal set G must be reached by all behaviors of the closed-loop system while staying inside a safe set S . Specifically, the state of the system is assumed to be initialized to any state in the set S . RWS properties include safety properties (stay inside a safe set S), reachability properties (reach a goal set G), and “control-to-facet” problems [7, 8].

The controller synthesis is addressed in two phases: (a) formulating a *control certificate* whose existence guarantees the existence of a non-Zeno switching control law for the given RWS specification, and (b) solving for a certificate of a particular form as a feasibility problem. The control certificates are control Lyapunov-like functions which represent a strategy for the controller to satisfy the specifications. Additionally, this strategy can be effectively implemented as a feedback law using a controller that respects min dwell time constraints. In the second phase, a counterexample guided inductive synthesis (CEGIS) framework [26], — an approach that uses SMT solvers at its core — is used to discover such control certificates. However, this procedure is used off the shelf, building upon the previous work of

Ravanbakhsh et al. [22]. This procedure uses a specialized solver for finding a certificate of a given parametric form that handles quantified formulas by alternating between a series of quantifier free formulas using existing SMT solvers [18, 3].

The contributions of the paper are as follows: first, we show that a straightforward formulation of the control certificate for the RWS problem yields an exponential number of conditions, and hence can be computationally infeasible. Next, we introduce a class of control certificates which (i) has a concise logical structure that makes the problem of discovering the certificates computationally feasible; and (ii) we show that such certificates yield corresponding switching strategies with a min-dwell time property unlike the conventional control certificates. Next, we extend our approach to the initialized RWS (IRWS) property that additionally restricts the set of initial conditions of the system using a class of “control zero-ing” barrier functions [33, 36]. Also, a suitable formulation for these functions is provided within our framework. Finally, we provide numerical examples to demonstrate the effectiveness of the method, including comparisons with recently developed state-of-art automatic control synthesis tool SCOTS [25].

1.1 Related Work

The broader area of temporal logic synthesis seeks to synthesize formally guaranteed controllers from the given plant model and specifications. The dominant approach is to build a discrete abstraction of the given plant that is related to the original system [35, 15, 10, 25, 17]. Once a suitable abstraction is found, these approaches use a systematic temporal logic-based controller design approach over the abstraction [32]. The properties of interest in these systems include the full linear temporal logic (LTL) and an efficiently synthesizable subset such as GR(1) [35, 15]. These approaches differ in how the abstraction can be constructed in a guaranteed manner. One class of approaches works by fixing a time step, gridding the state-space, and simulating one point per cell [17, 10, 25, 37, 31]. The resulting abstraction, however, is not always approximately bisimilar to the original system. Nevertheless, conditions such as open loop incremental stability of the plant can be used to obtain bisimilarity [5]. Alternatively, the abstraction can be built without time discretization [20, 15] by considering infeasible transitions. And furthermore, the abstraction can be iteratively refined through a counter-example refinement scheme [19]. Our work here does not *directly* focus on building abstractions. Rather, our focus is on deductive approaches for a narrow class of temporal logic properties namely RWS properties. Using our approach, control systems for richer properties can be built from solving a series of RWS problems.

Our approach is closely related to work of Habets et al. [6] and Kloetzer et al. [11]. In these methods, an abstraction is obtained by solving local control-to-facet problems instead of reachability analysis. However, continuous feedback is synthesized for each control-to-facet problem. The key difference in this paper is that the control-to-facet problems themselves are solved using switching. Furthermore, we consider initialized problems, where the initial states are also restricted to belong to a set. We find that IRWS problems can often be realized through a controller even when the corresponding RWS problem (for which the initial condition is not restricted) cannot be synthesized.

Another related class of solutions is based on synthesizing “a deductive proof of correctness” simultaneously with “a control strategy”. The goal of these approaches also consists of finding a control certificate, which yields a (control) strategy to guarantee the property. This typically takes the form of a control Lyapunov-like function. The idea of control Lyapunov functions goes back to Artstein [1] and Sontag [27]. The problem of *discovering* a control Lyapunov function is usually formulated using bilinear matrix inequalities (BMI) [30]. Also, instead of solving such NP-hard problems, usually alternating optimization (V-K iteration or policy iteration) is used to conservatively find a solution [30, 4].

Wongpiromsarn et al. [34] discuss *verification* of temporal logic properties using barrier certificates.

For synthesis, Xu et al. [36] discuss conditions for the so-called “control zeroing” barrier functions for safety and their properties. They also, consider their combination with control Lyapunov functions. In this article, we provide an alternative condition that is based on “exponential condition” barrier functions [12] and enforcing a compatibility condition between the control actions suggested by the control barrier and control Lyapunov functions. Also, Dimitrova et. al. [2] have shown that control certificates can be extended to address more complicated specifications i.e. parity games. While these results show that constraint solving based methods can be applied on more complicated specification, no method of finding such certificates is provided.

The use of SMT solvers in control synthesis has also been well-studied. Taly et. al [28, 29] use a constraint solving approach to find control certificates for reachability and safety. They adapt a technique known as Counter-Example Guided Inductive Synthesis (CEGIS), originally proposed for program synthesis [26], to solve the control problems using a combination of an SMT solver with numerical simulations. Ravanbakhsh et al. [22] propose a combination of SMT and SDP solvers for finding control certificates. However, their method is only applicable to stability or simple reachability properties, involving the use of a single Lyapunov function. In a subsequent paper, their approach is extended to handle disturbance inputs [23]. The use of SMT solvers to solve for Lyapunov-like functions is used in our paper as well. However, this paper focuses on defining a more tractable class of control certificates for RWS problems. Furthermore, we show how these problems can be composed for more complex temporal objectives. In particular, our use of the CEGIS procedure is not a contribution of this paper. Furthermore, in order to handle nonlinear systems and also to guarantee numerical soundness of these solvers, we use the nonlinear SMT solver dReal [3].

Huang et. al. [9] also propose control certificates to solve the RWS problem for piecewise affine systems, using SMT solvers. Their approach uses piecewise constant functions as control certificates and partitions the state space into small enough cells in order to define such functions. By using this technique, any function can be approximated, which makes the method relatively complete.

As mentioned earlier, past work by Habets et al. and Klutzier et al. [6, 11] build a finite abstraction by repeatedly solving control-to-facet problems. These problems seek to find a feedback law inside a polytope P that guarantees all the resulting trajectories exit P through a specific facet F of P . Habets et. al. [7] show necessary and sufficient conditions for the existence of a control strategy for the control-to-facet problem on simplices. This condition is sufficient but not necessary for polytopes. They extract a unique certificate from each problem instance and check whether the condition holds for the certificate. Subsequently, Roszak et al. [24] and Helwa et al. [8] extend this approach and solve reachability to a set of facets by introducing flow condition, which combined with invariant condition serves as a control certificate similar to those used in this paper. From the published results, these methods are more efficient, but are only applicable to affine systems over polytopes. In contrast, the dynamics in this article can be non-linear involving rational, trigonometric, and exponential functions. In this article, we demonstrate that our method can be used to solve such problems and it can be integrated into other methods which build an abstraction for the system.

2 Background

2.1 Notation

Given a function $f(t)$, let $f^+(t)$ ($f^-(t)$) be the right (left) limit of f at t , and $\dot{f}(t)$ represent the *right derivative* of f at time t . For a set $S \subseteq \mathbb{R}^n$, ∂S and $int(S)$ are its boundary and interior, respectively.

Definition 1 (*Nondegenerate Basic Semialgebraic Set*):

A nondegenerate basic semialgebraic set K is a nonempty set defined by a conjunction polynomial inequalities:

$$K : \{ \mathbf{x} \mid p_{K,1}(\mathbf{x}) \leq 0 \wedge \cdots \wedge p_{K,i}(\mathbf{x}) \leq 0 \},$$

where $\mathbf{x} \in \mathbb{R}^n$. For each $j \in [1, i]$, we define

$$H_{K,j} = \{ \mathbf{x} \mid \mathbf{x} \in K \wedge p_{K,j}(\mathbf{x}) = 0 \}.$$

It is required that (a) each $H_{K,j}$ is nonempty, (b) the boundary ∂K and the interior $\text{int}(K)$ are given by $\bigvee_{j=1}^i H_{K,j}$ and $\bigwedge_{j=1}^i p_{K,j}(\mathbf{x}) < 0$, respectively, and (c) the interior is nonempty. We use “basic semialgebraic” and “nondegenerate basic semialgebraic” interchangeably.

2.2 Switched Systems

We consider continuous-time switched system plants, controlled by a memoryless controller that provides continuous-time switching feedback. The state of the plant \mathcal{P} is defined by n continuous variables \mathbf{x} in a state space $X \subseteq \mathbb{R}^n$, along with a finite set of modes $Q = \{q_1, \dots, q_m\}$. The trace of the system $(q(t), \mathbf{x}(t))$ maps time to mode $q(\cdot) : \mathbb{R}^+ \rightarrow Q$, and state $\mathbf{x}(\cdot) : \mathbb{R}^+ \rightarrow X$. The mode $q \in Q$ is controlled by an external switching input $q(t)$. The state of the plant inside each mode evolves according to (time invariant) dynamics:

$$\dot{\mathbf{x}}(t) = f_{q(t)}(\mathbf{x}(t)), \quad (1)$$

wherein $f_q : X \rightarrow \mathbb{R}^n$ is a Lipschitz continuous function over X , describing the vector field of the plant for mode q .

The controller \mathcal{C} is defined as a function $\mathcal{K} : Q \times X \rightarrow Q$, which given the current mode and state of the plant, decides the mode of the plant at the next time instant. Formally:

$$q^+(t) = \mathcal{K}(q(t), \mathbf{x}(t)). \quad (2)$$

The closed loop $\langle \mathcal{P}, \mathcal{C} \rangle$ produces traces $(q(t), \mathbf{x}(t))$ defined jointly by equations (1) and (2). However, care must be taken to avoid *Zenoness*, wherein the controller can switch infinitely often in a finite time interval. Such controllers are physically unrealizable. Therefore, we will additionally ensure that the \mathcal{K} function satisfies a *minimum dwell time* requirement that guarantees a minimum time $\delta > 0$ between mode switches.

Definition 2 (*Minimum Dwell Time*): A controller \mathcal{C} has a minimum dwell time $\delta > 0$ with respect to a plant \mathcal{P} iff for all traces and for all switch times T ($q(T) \neq q^+(T)$), the controller does not switch during the times $t \in [T, T + \delta)$: i.e., $\mathcal{K}(q(t), \mathbf{x}(t)) = q^+(T)$ for all $t \in [T, T + \delta)$.

Once the function \mathcal{K} is defined with a minimum dwell time guarantee, given initial mode $(q(0))$, and initial state $(\mathbf{x}(0))$, a unique trace is defined for the system.

Specifications: Generally, specifications describe desired sequences of plant states $\mathbf{x}(t)$ over time $t \geq 0$ that we wish to control for. In this paper, we focus on *reach-while-stay* (RWS) specifications involving three sets: *initial set* $I \subseteq X$, *safe set* $S \subseteq X$ and *goal set* $G \subseteq X$.

Definition 3 (*Initialized Reach-While-Stay (RWS) Specification*): A trace $\mathbf{x}(t)$ for $t \in [0, \infty)$ satisfies a reach-while-stay (RWS) specification w.r.t sets $\langle I, S, G \rangle$ iff whenever $\mathbf{x}(0) \in I$, there exists a time $T \geq 0$ s.t. for all $t \in [0, T)$, $\mathbf{x}(t) \in S$, and $\mathbf{x}(T) \in G$.

In other words, whenever the system is initialized inside the set I , it stays inside the safe set S until it reaches the goal set G . Alternatively, we may express the specification in temporal logic as $I \implies (S \mathcal{U} G)$, where \mathcal{U} is the temporal operator ‘‘until’’.

We will assume that set S is a compact basic semialgebraic set. Typical examples include polytopes defined by linear inequalities or ellipsoids, that can be easily checked for the properties such as compactness and nondegeneracy. Also, sets I and G are compact semialgebraic sets.

The special case when $I = S$ will be called *uninitialized* RWS. Such a property simply states that the system initialized inside the set S continues to remain in S until it reaches a goal state $\mathbf{x} \in G$ at some finite time instant T . This case is suitable for building a finite abstraction as mentioned in Sec. 1.

2.3 Control Certificates

Encoding verification and synthesis problems into (control) certificates, which are defined by a set of conditions, is a standard approach. For example Lyapunov functions have been used for ensuring stability and barrier functions are employed to reason about safety properties. However, these functions are not usually known in advance. To discover such a function in the first place, we solve a constrained problem in which certificates are parameterized. Usually, certificates are defined over polynomials with unknown coefficients and the problem reduces to finding proper coefficients for polynomials [21, 2]. For example, to find a Lyapunov function, first, a template for Lyapunov function V is chosen: $V = \sum_i c_\alpha \mathbf{x}^\alpha$, where \mathbf{x}^α is a monomial with degree greater than zero. Then, solving the following constrained problem yields a Lyapunov function for proving stability to origin: $(\exists \mathbf{c}) (\forall \mathbf{x} \neq 0) (V(\mathbf{x}) > 0 \wedge \dot{V}(\mathbf{x}) < 0)$, where \dot{V} is $\nabla V \cdot f(\mathbf{x})$. In these techniques, it is essential to define *control* certificate with a simple structure that can be discovered automatically. In the subsequent we combine the certificates for safety and liveness to obtain a certificate for RWS properties.

3 RWS for Basic Semialgebraic Safe Sets

In this section, we first focus on the uninitialized RWS problem ($I = S$) and provide solutions for the case when S is a basic nondegenerate semialgebraic set (see Def. 1).

Let S be a nondegenerate basic semialgebraic sets, as in Def. 1. Let ∂S be partitioned into nonempty facets F_1, \dots, F_{l_k} . Each facet F_k is, in turn, defined by two sets of polynomial inequalities $F_k^<$ of inactive constraints and $F_k^=$ of active constraints: $F_k = \{\bigwedge_{p_{S,j} \in F_k^<} p_{S,j}(\mathbf{x}) < 0 \wedge \bigwedge_{p_{S,j} \in F_k^=} p_{S,j}(\mathbf{x}) = 0\}$.

For each state on a facet and not in G , we require the existence of a mode q , whose vector field points inside S . Additionally, we will require a certificate V to decrease everywhere in $S \setminus G$. For any polynomial p , let $\dot{p}_q : (\nabla p) \cdot f_q(\mathbf{x})$. By combining conditions for safety and liveness, one can obtain the following conditions:

$$\left\{ \begin{array}{l} \mathbf{x} \in \text{int}(S) \setminus G \implies (\exists q) \dot{V}_q(\mathbf{x}) < -\varepsilon \\ \mathbf{x} \in F_1 \setminus G \implies (\exists q) \left(\dot{V}_q(\mathbf{x}) < -\varepsilon \wedge \bigwedge_{p \in F_1^=} \dot{p}_q(\mathbf{x}) < -\varepsilon \right) \\ \vdots \\ \mathbf{x} \in F_{l_k} \setminus G \implies (\exists q) \left(\dot{V}_q(\mathbf{x}) < -\varepsilon \wedge \bigwedge_{p \in F_{l_k}^=} \dot{p}_q(\mathbf{x}) < -\varepsilon \right). \end{array} \right. \quad (3)$$

The first condition in Eq. (3) states that V must strictly decrease everywhere in the set $\text{int}(S) \setminus G$. The subsequent conditions treat each facet F_j of the set S and posit the existence of a mode q for each state that causes the active constraints and the function V to decrease.

However, we note that as the number of state variables increases, the number of facets can be exponential in the number of inequalities that define S [8]. This poses a serious limitation to the applicability of Eq. (3).

Our solution to this problem, is based partly on the idea of exponential barriers discussed by Kong et al. [12]. Rather than force the vector field to point inwards at each facet, we simply ensure that each polynomial inequality $p_{S,j} \leq 0$ that defines S , satisfies a decrease condition outside set G . Thus, Eq. (3) is replaced by a simpler (relaxed) condition:

$$\mathbf{x} \in S \setminus G \implies (\exists q) \dot{V}_q(\mathbf{x}) < -\varepsilon \wedge \bigwedge_j \left((p_{S,j,q}(\mathbf{x}) + \lambda p_{S,j}(\mathbf{x})) < -\varepsilon \right). \quad (4)$$

Here $\lambda > 0$ is a user specified parameter. This rule is a relaxation of (3). The rule is made stronger for larger values of λ . However, larger values of λ can cause numerical difficulties in practice while searching for a control certificate.

For safety constraints, we require \dot{p}_q to be numerically $\leq -\varepsilon$ mainly, to avoid numerical issues. This can be restrictive for cases where $p_{S,j,q}$ is simply zero. To go around this, we define a set of facets $J_q = \{j | (\exists x) p_{S,j,q}(x) > 0\}$ for each mode q . Informally speaking, J_q is set of all facets for which change of $p_{S,j}$ must be considered when mode q is selected. Because for each facet $j \notin J_q$, $p_{S,j}$ will never increase as long as mode q is selected. Then, the conditions become:

$$\mathbf{x} \in S \setminus G \implies (\exists q) \dot{V}_q(\mathbf{x}) < -\varepsilon \wedge \bigwedge_{j \in J_q} \left((p_{S,j,q}(\mathbf{x}) + \lambda p_{S,j}(\mathbf{x})) < -\varepsilon \right). \quad (5)$$

As mentioned earlier, the problem of control synthesis consists of two phases. The first phase deals with the problem of *finding a control certificate* $V(\mathbf{x})$ that satisfies (5). We use a *counter-example guided inductive synthesis* (CEGIS) framework to find such certificates. In the second phase, a switching strategy is extracted from the control certificate to design the final controller. We now examine each phase, in turn.

3.1 Discovering Control Certificates

We now explain the CEGIS framework that searches for a suitable control certificate V . To synthesize a control certificate, we start with a parametric form $V_{\mathbf{c}}(\mathbf{x}) = V(\mathbf{c}, \mathbf{x}) : \sum_{i=1}^N c_i g_i(\mathbf{x})$ with some (nonlinear) basis functions $g_1(\mathbf{x}), \dots, g_N(\mathbf{x})$ chosen by the user, and unknown coefficients $\mathbf{c} : (c_1, \dots, c_N)$, s.t. $\mathbf{c} \in C$ for a compact set $C \subseteq \mathbb{R}^N$. The certificate V is a linear function over \mathbf{c} .

The constraints from Eq. (5) become as follows:

$$(\exists \mathbf{c} \in C) (\forall \mathbf{x} \in X) \mathbf{x} \in S \setminus G \implies \bigvee_q \left(\dot{V}_q < -\varepsilon \wedge \bigwedge_{j \in J_q} (p_{S,j,q}(\mathbf{x}) + \lambda p_{S,j}(\mathbf{x}) < -\varepsilon) \right). \quad (6)$$

The constraints in Eq. (6) has a complex quantifier alternation structure involving the $\exists \mathbf{c}$ quantifier nested outside the $\forall \mathbf{x}$ quantifier. First, we note that J_q is computed separately and here we assume it is given. Next, we modify an algorithm commonly used for program synthesis problems to the problem of synthesizing the coefficients $\mathbf{c} \in C$ [26].

The counterexample guided inductive synthesis (CEGIS) approach has its roots in program synthesis, wherein it was proposed as a general approach to solve $\exists \forall$ constraints that arise in such problems [26].

The key idea behind the CEGIS approach is to find solutions to such constraints while using a satisfiability (feasibility) solver for quantifier-free formulas that check whether a given set of constraints without quantifiers have a feasible solution.

Solvers like Z3 allow us to solve many different classes of constraints with extensive support for linear arithmetic constraints [18]. On the other hand, general purpose nonlinear delta-satisfiability solvers like dReal, support the solving of quantifier-free nonlinear constraints involving polynomials, trigonometric, and rational functions [3]. However, the presence of quantifiers drastically increases the complexity of solving these constraints. Here, we briefly explain the idea of CEGIS procedure for $\exists\forall$ constraints of the form

$$(\exists \mathbf{c} \in C) (\forall \mathbf{x} \in X) \Psi(\mathbf{c}, \mathbf{x}).$$

Here, \mathbf{c} represents the unknown coefficients of a control certificate and \mathbf{x} represents the state variables of the system. Our goal is to find one witness for \mathbf{c} that makes the overall quantified formula true. The overall approach constructs, maintains, and updates two sets iteratively:

1. $X_i \subseteq X$ is a finite set of *witnesses*. This is explicitly represented as $X_i = \{\mathbf{x}_1, \dots, \mathbf{x}_i\}$.
2. $C_i \subseteq C$ is a (possibly infinite) subset of available *candidates*. This is implicitly represented by a constraint $\psi_i(\mathbf{c})$, s.t. $C_i : \{\mathbf{c} \in C \mid \psi_i(\mathbf{c})\}$.

In the beginning, $X_0 = \{\}$ and $\psi_0 : \text{true}$ representing the set $C_0 : C$.

At each iteration, we perform the following steps:

- (a) *Choose* a candidate solution $\mathbf{c}_{i+1} \in C_i$. This is achieved by checking the feasibility of the formula ψ_i . Throughout this paper, we will maintain ψ_i as a *linear arithmetic* formula that involves boolean combinations of linear inequality constraints. Solving these problems is akin to solving linear optimization problems involving disjunctive constraints. Although the complexity is NP-hard, solvers like Z3 integrate fast LP solvers with Boolean satisfiability solvers to present efficient solutions [18].
- (b) *Test* the current candidate. This is achieved by testing the satisfiability of $\neg\Psi(\mathbf{c}, \mathbf{x})$ for fixed $\mathbf{c} = \mathbf{c}_{i+1}$. In doing so, we obtain a set of nonlinear constraints over \mathbf{x} . We wish to now check if it is feasible.

If $\neg\Psi(\mathbf{c}_{i+1}, \mathbf{x})$ has no feasible solutions, then $\Psi(\mathbf{c}_{i+1}, \mathbf{x})$ is true (valid) for all \mathbf{x} . Therefore, we can stop with $\mathbf{c} = \mathbf{c}_{i+1}$ as the required solution for \mathbf{c} .

Otherwise, if $\neg\Psi(\mathbf{c}, \mathbf{x})$ is feasible for some $\mathbf{x} = \mathbf{x}_{i+1}$, we add it back as a witness: $X_{i+1} : X_i \cup \{\mathbf{x}_{i+1}\}$. The formula ψ_{i+1} is given by

$$\psi_{i+1} : \psi_i \wedge \Psi(\mathbf{c}, \mathbf{x}_{i+1}).$$

Note that $\psi_{i+1} \implies \psi_i$, and \mathbf{c}_{i+1} is no longer a feasible point for ψ_{i+1} . The set C_{i+1} described by ψ_{i+1} is:

$$C_{i+1} : \{\mathbf{c} \in C \mid \Psi(\mathbf{c}, \mathbf{x}_i) \text{ holds for each } \mathbf{x}_i \in X_{i+1}\}.$$

The CEGIS procedure either (i) runs forever, or (ii) terminates after i iterations with a solution $\mathbf{c} : \mathbf{c}_i$, or (iii) terminates with a set of witness points X_i proving that no solution exists.

We now provide further details of the CEGIS procedure adapted to find a certificate that satisfies Eq. (6). In the CEGIS procedure, the formula $\Psi(\mathbf{c}, \mathbf{x})$ will have the following form:

$$\left\{ \begin{array}{l} \mathbf{x} \in R_1 \implies \varphi_1(\mathbf{c}, \mathbf{x}) \\ \mathbf{x} \in R_2 \implies \varphi_2(\mathbf{c}, \mathbf{x}) \\ \dots \\ \mathbf{x} \in R_{N_j} \implies \varphi_{N_j}(\mathbf{c}, \mathbf{x}), \end{array} \right. \quad (7)$$

and each φ_j for $j = 1, \dots, N_j$ has the form

$$\bigvee_k \bigwedge_l p_{j,k,l}(\mathbf{c}, \mathbf{x}) > 0, \quad (8)$$

where $p_{j,k,l}(\mathbf{c}, \mathbf{x})$ is a function linear in \mathbf{c} and possibly nonlinear in \mathbf{x} , depending on the dynamics and template used for the control certificate.

The CEGIS procedure involves two calls to solvers: (a) Testing satisfiability of $\Psi_i(\mathbf{c})$ and (b) Testing the satisfiability of $\neg\Psi(\mathbf{c}_{i+1}, \mathbf{x})$. We shall discuss each of these problems in the following paragraphs.

Finding Candidate Solutions: Given a finite set of witnesses X_i , a solution exists for Ψ_{i+1} iff there exists $\mathbf{c} \in C$ s.t.

$$\bigwedge_{\mathbf{x} \in X_i} \bigwedge_{j=1}^{N_j} \left(\mathbf{x} \in R_j \implies \bigvee_k \bigwedge_l p_{j,k,l}(\mathbf{c}, \mathbf{x}) > 0 \right),$$

and since $p_{j,k,l}$ is a linear function in \mathbf{c} , such \mathbf{c} can be found by solving a formula in Linear Arithmetic Theory ($\mathcal{L}\mathcal{A}$).

Finding Witnesses: Finding a witness for a given candidate solution \mathbf{c}_i involves checking the satisfiability $\neg\Psi$. Whereas Ψ is a conjunction of N_j clauses, $\neg\Psi$ is a disjunction of clauses. The j^{th} clause in $\neg\Psi$ ($1 \leq j \leq N_j$) has the form

$$\mathbf{x} \in R_j \wedge \bigwedge_k \bigvee_l p_{j,k,l}(\mathbf{c}_i, \mathbf{x}) \leq 0. \quad (9)$$

We will test each clause separately for satisfiability. Assuming that $p_{j,k,l}$ is a general nonlinear function over \mathbf{x} , SMT solvers like dReal [3] can be used to solve this over a compact set R_j . Numerical SMT solvers like dReal can either conclude that the given formula is unsatisfiable or provide a solution to a “nearby” formula that is δ close. The parameter δ is adjusted by the user. As a result, dReal can correctly conclude that the current candidate yields a valid certificate. On the other hand, its witness may not be a witness for the original problem. In this case, using the spurious witness may cause the CEGIS procedure to potentially continue (needlessly) even when a solution \mathbf{c}_i has been found. Nevertheless, the overall procedure produces a correct result whenever it terminates with an answer.

Example 1 *This example is adopted from [19]. There are two variables and three control modes with the dynamics given below:*

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -x_2 - 1.5x_1 - 0.5x_1^3 \\ x_1 \end{bmatrix} + B_q, \quad B_{q1} = \begin{bmatrix} 0 \\ -x_2^2 + 2 \end{bmatrix}, \quad B_{q2} = \begin{bmatrix} 0 \\ -x_2 \end{bmatrix}, \quad B_{q3} = \begin{bmatrix} 2 \\ 10 \end{bmatrix}.$$

The goal is to reach the target set $G : (x_1 + 0.75)^2 + (x_2 - 1.75)^2 \leq 0.25^2$, a circle centered at $(-0.75, 1.75)$, as shown in Figure 1a, while staying in the safe region given by the rectangle $S_0 : [-2, 2] \times [-2, 3]$:

$$S_0 : \{\mathbf{x} | (x_1 + 2)(x_1 - 2) \leq 0 \wedge (x_2 + 2)(x_2 - 3) \leq 0\}.$$

First, we shift co-ordinates to transform $(-0.75, 1.75)$ as the new origin. Then, we use a quadratic template for $V(c_1x_1^2 + c_2x_1x_2 + c_3x_2^2)$, $\varepsilon = 1$, $\lambda = 5$. The solution V is found in 5 iterations. Then, we translate the function back to the original co-ordinates:

$$V(x_1, x_2) : 37.782349x_1^2 - 2.009762x_1x_2 + 60.190607x_1 + 4.415093x_2^2 - 16.960145x_2 + 37.411604.$$

Example 2 A unicycle [37] has three variables. x and y are position of the car and θ is its angle. The dynamics of the system is $\dot{x} = u_1 \cos(\theta)$, $\dot{y} = u_1 \sin(\theta)$, $\dot{\theta} = u_2$, where u_1 and u_2 are inputs. Assuming a switched system, we consider $u_1 \in \{-1, 0, 1\}$ and $u_2 \in \{-1, 0, 1\}$. The safe set is $[-1, 1] \times [-1, 1] \times [-\pi, \pi]$ and the target facet is $x = 1$. We use a template that is linear in (x, y) and quadratic in θ . Using $\varepsilon = 0.1$ and $\lambda = 0.5$, the following CLF is found after 22 iterations:

$$V(\mathbf{x}) : -x - y - 0.5881\theta + \theta^2 - 0.1956\theta x + \theta y.$$

Example 3 This example is adopted from [7]. There are four variables and two control inputs. The dynamic is as follows:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} x_1 + x_2 + 8 \\ -x_2 + x_3 + 1 \\ -2x_3 + 2x_4 + 1 \\ -3x_4 + 1 \end{bmatrix} + \begin{bmatrix} u_1 \\ -u_2 \\ -2u_1 \\ u_2 \end{bmatrix}.$$

The region of interest S is hyper-box $[-1, 1]^4$ and the input belongs to set $[0, 1] \times [0, 2]$. The goal is to reach facet $x_1 = 1$, while staying in S as the safe region.

First, we discretize the control input to model the system as a switched system. For this purpose, we assume $u_1 \in \{0, 1\}$ and $u_2 \in \{0, 0.5, 1, 1.5, 2\}$. Then, we use a linear template for the CLF ($c_1 x_1 + c_2 x_2 + c_3 x_3 + c_4 x_4$), $\varepsilon = 0.1$, $\lambda = 5$. CEGIS framework finds certificate $V(\mathbf{x}) : -0.13333344(x_1 + x_2 + x_3 + x_4)$.

3.2 Control Design

Thus far, we discussed the CEGIS framework for finding a control certificate. Extracting the \mathcal{H} function from the certificate is now considered. Given a control certificate V satisfying Eq. (5), the choice of a switching mode is dictated by a function $\eta_q(\mathbf{x})$ defined for each state $\mathbf{x} \in X$ and mode $q \in Q$ as follows:

$$\eta_q(\mathbf{x}) : \max \left(\dot{V}_q(\mathbf{x}), \eta_{S,1,q}(\mathbf{x}), \dots, \eta_{S,k,q}(\mathbf{x}) \right),$$

where for all $j \in J_q$, $\eta_{S,j,q}$ is $p_{S,j,q} + \lambda p_{S,j}$ and for $j \notin J_q$, $\eta_{S,j,q} = -\infty$ or equivalently $\eta_{S,j,q} = -L$ for some large constant L .

The idea is that whenever (at time t) the controller chooses a mode q s.t. $\eta_q(\mathbf{x}(t)) < -\varepsilon$, one can guarantee that $\eta_q(\mathbf{x}(t)) < 0$ holds for all $t \in [T, T + \delta)$, for some minimum time δ . Therefore, for some fixed ε_s ($0 < \varepsilon_s < \varepsilon$), the function \mathcal{H} for any $\mathbf{x} \in S \setminus G$ can be defined as

$$\mathcal{H}(q, \mathbf{x}) := \begin{cases} \hat{q} & \text{if } \left(\eta_q(\mathbf{x}) \geq -\varepsilon_s \wedge \eta_{\hat{q}}(\mathbf{x}) < -\varepsilon \right) \\ q & \text{otherwise.} \end{cases} \quad (10)$$

In other words, the controller state persists in a given mode q until $\eta_q(\mathbf{x}) \geq -\varepsilon_s$. Then, given that $\mathbf{x} \in S \setminus G$, Eq. (5) will provide us a new control mode \hat{q} that satisfies $\eta_{\hat{q}}(\mathbf{x}) < -\varepsilon$. This mode is chosen as the next mode to switch to.

Example 4 Consider once again, the problem from Ex. 1. Using the defined function $V(x_1, x_2)$, Eq. (10) yields a controller. Figure 1b shows some of the simulation traces of this closed loop system, demonstrating the RWS property.

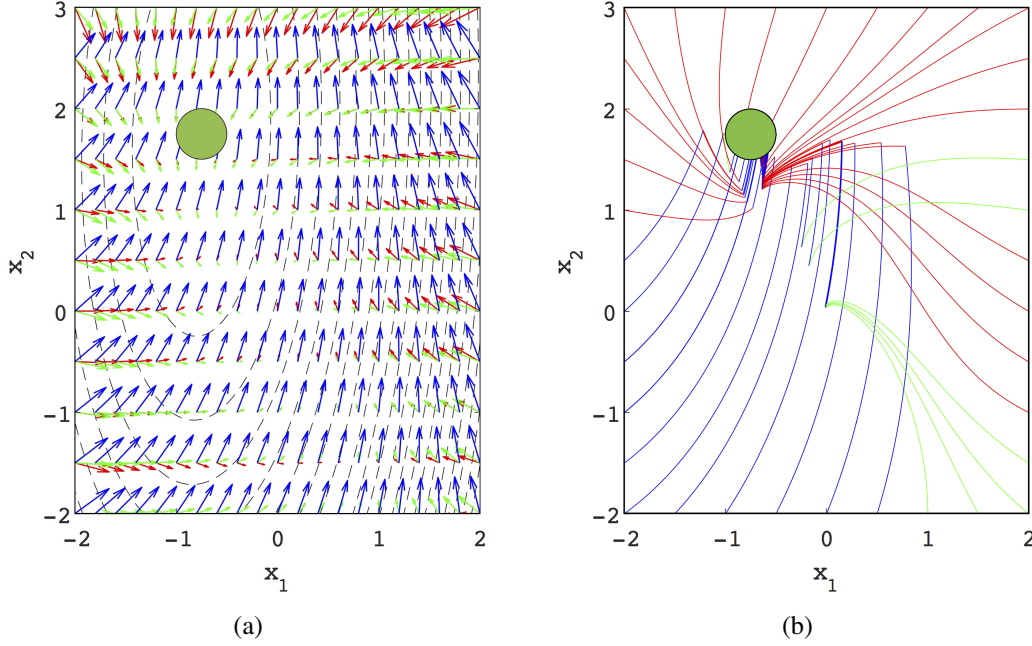


Figure 1: (a) Region G for Example 1 is shown shaded in the center, and the vector fields for modes q_1, q_2 and q_3 are shown in red, green and blue, respectively. Level-sets of V are shown with black dashed lines. (b) Closed loop trajectories for Example 1 using the controller defined by Eq. (10). The segments shown in colors red, green and blue correspond to the modes q_1, q_2 and q_3 , respectively.

We now establish the key result that provides a minimum dwell time guarantee.

Lemma 1 *There exists a $\delta > 0$ s.t. for all initial conditions $\mathbf{x}(T) \in S \setminus G$, if $\eta_q(\mathbf{x}(T)) < -\varepsilon$, and if the mode of the system is set to q at time T , then*

$$(\forall t \in [T, T + \delta]) (\mathbf{x}(t) \in S \setminus G) \implies \eta_q(\mathbf{x}(t)) \leq -\varepsilon_s.$$

Proof *Let $T + \delta$ be the earliest time instant, where $\eta_q(\mathbf{x}(T + \delta)) \geq -\varepsilon_s$ while at the same time*

$$(\forall t \in [T, T + \delta]) q(t) = q, \mathbf{x}(t) \in S \setminus G.$$

At time T , $\eta_q(\mathbf{x}(T)) < -\varepsilon$ and at time $T + \delta$, $\eta_q(\mathbf{x}(T + \delta)) = -\varepsilon_s$. Note that $\eta_q(\mathbf{x})$ is defined as $\max(\alpha_1(\mathbf{x}), \dots, \alpha_m(\mathbf{x}))$ for some smooth functions $\alpha_1, \dots, \alpha_m$. As a result, Since S is a bounded set, and p, f_q , and V are bounded over S , there exists a constant $\Lambda > 0$ s.t.

$$(\forall \mathbf{x} \in S) \dot{\alpha}_{i,q} \leq \Lambda. \tag{11}$$

Therefore, for each α_i , we have

$$\alpha_i(\mathbf{x}(T + \delta)) = \alpha_i(\mathbf{x}(T)) + \int_{t=T}^{T+\delta} \dot{\alpha}_{i,q}(\mathbf{x}(t)) dt \leq \alpha_i(\mathbf{x}(T)) + \Lambda \delta.$$

As a result, we conclude that

$$\eta_q(\mathbf{x}(T + \delta)) = \max_i \alpha_i(\mathbf{x}(T + \delta)) = \alpha_{j^*}(\mathbf{x}(T + \delta)) \leq \alpha_{j^*}(\mathbf{x}(T)) + \Lambda \delta \leq \eta_q(T) + \Lambda \delta.$$

Therefore, we can conclude $-\varepsilon_s < -\varepsilon + \Lambda\delta \implies \frac{\varepsilon - \varepsilon_s}{\Lambda} < \delta$ and there exists a fixed $\delta > \frac{\varepsilon - \varepsilon_s}{\Lambda} > 0$ s.t.

$$(\forall t \in [T, T + \delta)) \eta_q(\mathbf{x}(t)) < -\varepsilon_s. \blacksquare$$

Eq. (10) gives a switching strategy which respects the min-dwell time and as long as $\mathbf{x}(t) \in S$, the controller guarantees $\eta_{q(t)}(\mathbf{x}(t)) \leq -\varepsilon_s$. I.e. for all $j \in J_q$, $\dot{V}_q(\mathbf{x}(t)) \leq -\varepsilon_s$ and $p_{S,j,q}(\mathbf{x}(t)) + \lambda p_{S,j}(\mathbf{x}(t)) \leq -\varepsilon_s$.

Theorem 1 *Given nondegenerate basic semialgebraic set S , a semialgebraic set G , and a function V (satisfying Equation (5)), the control strategy defined by Eq. (10) respects the min-dwell time property and guarantees the RWS property defined by $S, G: S \implies S \mathcal{U} G$.*

Proof *As discussed, there exists a controller which respects the min-dwell time property. Also, the controller guarantees $\dot{V}_q(\mathbf{x}) \leq -\varepsilon_s$ and $(p_{S,j,q}(\mathbf{x}) + \lambda p_{S,j}(\mathbf{x})) \leq -\varepsilon_s$ (for all $j \in J_q$), as long as $\mathbf{x} \in S \setminus G$.*

Assume $\mathbf{x}(t)$ is on the boundary of S (and not in G) at some time t . Because S is assumed to be a nondegenerate basic semialgebraic set, there exists at least one j s.t. $p_{S,j}(\mathbf{x}(t)) = 0$. If $j \notin J_q$, by definition, $p_{S,j,q}$ is negative for all states and $p_{S,j,q}$ remains ≤ 0 as long as mode q is selected. Otherwise ($j \in J_q$), we obtain $p_{S,j,q}(\mathbf{x}(t)) \leq -\varepsilon_s < 0$. Therefore, there exists $\tau_j > 0$, s.t. $s \in (t, t + \tau_j)$, we conclude that $p_{S,j,q}(\mathbf{x}(s)) < 0$. As a result, the trajectory cannot leave the set S .

Thus, the trace cannot leave S , unless it reaches G . Now, we show that the trajectory cannot stay inside $S \setminus G$ forever. By the construction of the controller, we can conclude time diverges (because the controller respects the min-dwell time property) and that V decreases ($\dot{V}_q(\mathbf{x}(t)) \leq -\varepsilon_s$). However, the value of V is bounded on bounded set $S \setminus G$. Therefore, \mathbf{x} cannot remain in $S \setminus G$ and the only possible outcome for the trace is to reach G . \blacksquare

4 RWS for Semialgebraic Safe Set

As Habets et al [6] discussed, control-to-facet problems can be used to build an abstraction. Here, we demonstrate that the method described so far can be integrated in this framework to tackle more complicated problems. First, we briefly explain how the method works. For a more detailed discussion, the reader can refer to [6] or [11].

First, state space is decomposed into polytopes according to the specifications. Here, we can use basic semialgebraic sets instead of polytopes. Then, for each such a set u , we consider an abstract state $\mathcal{A}(u)$. Furthermore, for each of its $n - 1$ dimensional facet F , a control-to-facet problem is solved. The corresponding problem is to find a control strategy to reach F starting from u . If the control-to-facet problem is solved successfully, then for each basic semialgebraic set v with a $n - 1$ dimensional facet $F' \subseteq F$, an edge from $\mathcal{A}(u)$ to $\mathcal{A}(v)$ (with label/action F) is added to the abstraction. Also for each basic semialgebraic set u , one can check if u is a control invariant to build self loops. However, for RWS properties, self loops are redundant and we skip them here. After building the abstract system, we use standard techniques to solve the problem for finite systems. If the problem could be solved for the abstract system, then, one can design a controller.

First, for each abstract state $\mathcal{A}(u)$, there is at least one action F that agrees with the winning strategy for the abstract system. Let that action be $\mathcal{F}(\mathcal{A}(u))$. The idea is to implement transition $\mathcal{F}(\mathcal{A}(u))$, using controller $\mathcal{H}_{u, \mathcal{F}(\mathcal{A}(u))}$ for the corresponding control-to-facet problem [6]. Formally, the controller

can be defined as follows:

$$\mathcal{K}(q, \mathbf{x}) = \begin{cases} \mathcal{K}_{u_1, \mathcal{F}(\mathcal{A}(u_1))}(q, \mathbf{x}) & \mathbf{x} \in u_1 \\ \vdots \\ \mathcal{K}_{u_s, \mathcal{F}(\mathcal{A}(u_s))}(q, \mathbf{x}) & \mathbf{x} \in u_s. \end{cases} \quad (12)$$

When \mathbf{x} belongs to multiple sets, one can break the tie by some ordering, where states in the winning set have priorities. It is worth mentioning that combining these controllers together, does not produce any Zeno behavior as it is guaranteed that each abstract state is visited only once for RWS properties. However, superdense switching is possible as two facets of a polytope can get arbitrarily close.

If one is interested in LTL properties (not just reach-while-stay) or min-dwell time property, one possible solution is to use *fat* facets, where the target sets are n dimensional goal sets. This extends the domain of the control-to-facet problem to adjacent basic semialgebraic sets as well. Also, it allows the controller to continue using current sub-controller for some minimum time (if min-dwell time requirement is not met), before changing the sub-controller (at the switch time).

Example 5 Consider again the system from Example 1, with the addition of some obstacles [19]. More precisely, as shown in Fig 2a, safe set is $S = S_0 \setminus (O_1 \cup O_2)$. First, the safe set is decomposed into four basic semialgebraic sets, which are shown with R_0 to R_3 in Fig. 2a.

R_0 is the target set. Next, we build a transition relation between four abstract states, representing four basic semialgebraic sets. This is done by solving seven RWS problems for basic semialgebraic sets. For R_1 to R_0 , we use a quadratic template for V , and for other problems, we use linear template. The abstract system is shown in Fig. 2b. Next, the problem is solved for the abstract system. The solution to the abstract system is simple: if the state is in R_2 , the controller uses the left facet to reach R_1 or R_3 . Otherwise, if the state is in R_3 , the controller uses the upper facet to reach R_1 and finally, if the state is in R_1 , the controller makes sure the state reaches R_0 .

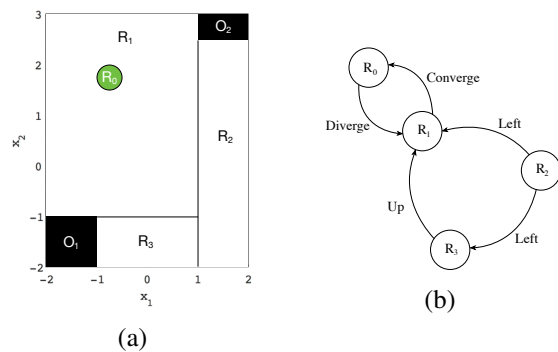


Figure 2: (a) Schematic view of state decomposition. (b) Finite abstraction for the original problem.

Example 6 This example is a path planning problem for the unicycle [25]. Projection of safe set on x and y yields a maze. The target set is placed at the right bottom corner of the maze (Fig. 3). Using specification-guided technique, we modeled the system with 53 polyhedra. Each polyhedron is treated as a single state and a transition relation is built by solving 113 control-to-facet problems. Then, the problem is solved over the finite graph. The total computation took 1484 seconds. The figure also shows a single trajectory of the closed loop system.

Example 7 This example is similar to Example 6, except for the fact that there is no direct control over the angular velocity. More precisely, only the angular acceleration is controllable and the system would have the following dynamics $\dot{x} = u_1 \cos(\theta)$, $\dot{y} = u_1 \sin(\theta)$, $\dot{\theta} = \omega$, $\dot{\omega} = u_2$. Also, we assume $\omega \in [-1, 1]$. By changing the coordinates one can use $r = \sqrt{x^2 + y^2}$, $z_1 = x \cos(\theta) + y \sin(\theta)$ and $z_2 = y \cos(\theta) - x \sin(\theta)$ to define position and angle of the car (cf. [13] for details). Then, we use the following template $V(x, y, \theta, \omega) = c_1 r^2 + c_2 z_1 + c_3 z_2 \omega + c_4 \omega^2$, where the origin is located just outside of the target facet. Using this template, we find control certificates for all 113 control-to-facet problems in 5296 seconds.

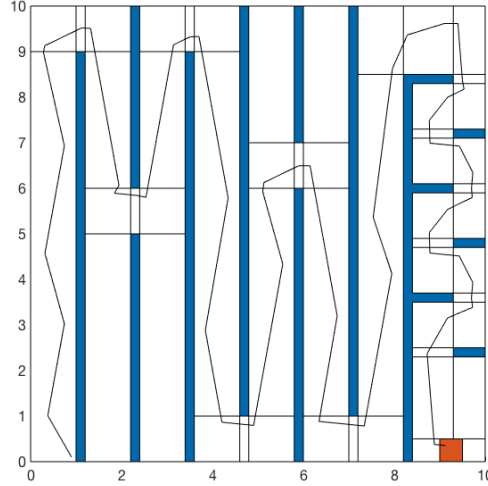


Figure 3: Region G is shown shaded in Orange, and unsafe regions are shown in blue. An execution trace of the car is shown for x and y variables.

5 Initialized Reach-While-Stay

So far, we discussed uninitialized RWS specifications ($S \implies S \mathcal{U} G$). In these systems, we use boundary of safe set as barrier. However, as pointed out by Lin et al. [14], this may not be the case. Now, we consider the initialized problem for a given initial set I ($I \implies S \mathcal{U} G$). To avoid technical difficulties, we assume that $I \subseteq \text{int}(S)$. The solution is to create a composite barrier that is formed by the boundary of S as well as other a priori unknown barrier functions.

Barrier Functions: We recall that for a control barrier function [29, 36], the following conditions are considered

$$\begin{aligned}
 \mathbf{x} \in \partial S &\implies B(\mathbf{x}) > 0 \\
 \mathbf{x} \in I &\implies B(\mathbf{x}) < 0 \\
 \mathbf{x} \in S &\implies \left(B(\mathbf{x}) = 0 \implies (\exists q) \dot{B}_q(\mathbf{x}) < -\varepsilon \right).
 \end{aligned} \tag{13}$$

This ensures that $B(\mathbf{x}) = 0$ is a barrier and ∂S is unreachable. Eq. (13), combined with the smoothness of B and f_q ensures that as soon as the state is sufficiently “close” to the barrier, it is possible to choose a control mode that ensures the local decrease of the B .

The condition in Equation (13) can be encoded into the CEGIS framework. However, the presence of the equality $B(\mathbf{x}) = 0$ poses practical problems. In particular, it requires for each candidate B_c , to find a counterexample \mathbf{x} s.t. $B_c(\mathbf{x}) \neq 0$. Unfortunately, such an assertion is easy to satisfy, resulting in the procedure always exceeding the maximum number of iterations permitted.

Again, we find that the following relaxation of the third condition is particularly effective in our experiments

$$\begin{aligned}
 \mathbf{x} \in \partial S &\implies B(\mathbf{x}) > 0 \\
 \mathbf{x} \in I &\implies B(\mathbf{x}) < 0 \\
 \mathbf{x} \in S &\implies \bigvee_q \left(\dot{B}_q(\mathbf{x}) - \lambda B(\mathbf{x}) < -\varepsilon \vee \dot{B}_q(\mathbf{x}) + \lambda B(\mathbf{x}) < -\varepsilon \right),
 \end{aligned} \tag{14}$$

for some constant λ .

Intuitively, by choosing $\lambda = 0$, the condition is similar to that of Lyapunov functions, whereas as $|\lambda| \rightarrow \infty$, the condition gets less conservative and in the limit, it is equivalent to the original condition.

In fact, for smaller $|\lambda|$ CEGIS terminates faster, but at the cost of missing potential solutions. On the other hand, using larger $|\lambda|$, is less conservative at the cost of CEGIS timing out. We also note that this formulation is less conservative than the one introduced by Kong et al. [12] as our formulation uses two exponential conditions which only forces decrease of value of B around $B^* = \{\mathbf{x} \mid B(\mathbf{x}) = 0\}$.

To solve the RWS in general form, we define a finite set of barriers \mathcal{B} with the following conditions:

$$\begin{aligned} \mathbf{x} \in \partial S &\implies \bigvee_{B \in \mathcal{B}} B(\mathbf{x}) > 0 \\ \mathbf{x} \in I &\implies \bigwedge_{B \in \mathcal{B}} B(\mathbf{x}) < 0. \end{aligned} \quad (15)$$

Also for each mode q , \mathcal{B}_q is defined as $\mathcal{B}_q = \{B \in \mathcal{B} \mid (\exists \mathbf{x}) \dot{B}_q(\mathbf{x}) > 0\}$. Then, existence of a proper mode can be encoded as the following:

$$\mathbf{x} \in S \setminus G \implies \left(\left(\dot{V}_q(\mathbf{x}) < -\varepsilon \right) \wedge \bigwedge_{B \in \mathcal{B}_q} \begin{pmatrix} \dot{B}_q(\mathbf{x}) + \lambda B(\mathbf{x}) < -\varepsilon \vee \\ \dot{B}_q(\mathbf{x}) - \lambda B(\mathbf{x}) < -\varepsilon \end{pmatrix} \right). \quad (16)$$

Theorem 2 *Given nondegenerate basic semialgebraic set S , semialgebraic sets I and G , function V , and a non-empty set of functions \mathcal{B} (satisfying Equation (15) and (16)), there is a control strategy that respects the min-dwell time property and guarantees the RWS property: $I \implies S \mathcal{U} G$.*

To simplify these constraints and reduce the number of unknowns, one can use some of $p_{S,i}$'s to fix some of these barriers, which yields conditions similar to the ones used for the uninitialized problem. This trick is demonstrated in the following example.

Example 8 *This example is taken from [16], in which a DC-DC converter is modeled with two variables i and v .*

The system has two modes q_1 and q_2 , with the following dynamics:

$$\begin{aligned} q_1 : \begin{cases} \dot{i} = 0.0167i + 0.3333 \\ \dot{v} = -0.0142v \end{cases} \\ q_2 : \begin{cases} \dot{i} = -0.0183i - 0.0663v + 0.3333 \\ \dot{v} = -0.0711i - 0.0142v. \end{cases} \end{aligned}$$

The safe set is $S : [0.65, 1.65] \times [4.95, 5.95]$ and the goal set is $G : [1.25, 1.45] \times [5.55, 5.75]$. We assume initial set to be $I : [0.85, 0.95] \times [5.15, 5.25]$ (Fig. 4). Then, we use 5 barriers B_0, \dots, B_4 . Using boundaries of S , we choose B_1, \dots, B_4 as follows:

$$\begin{aligned} B_1 &= 0.65 - i + \varepsilon_b & B_2 &= 1.65 - i + \varepsilon_b \\ B_3 &= 4.95 - v + \varepsilon_b & B_4 &= 5.95 - v + \varepsilon_b, \end{aligned}$$

where $\varepsilon_b > 0$ is small enough that $I \subset \text{int}(\bigcap_{i=1}^4 B_i)$. In this case, we choose $\varepsilon_b = 0.01$. Notice that such ε_b always exists by the definition. Next, we assume $B_0 = V$ and both have the following template:

$$B_0 = V : c_1(i - 1.35)^2 + c_2(i - 1.35)(v - 5.65) + c_3(v - 5.65)^2 - 1.$$

This template is chosen in a way that V is a quadratic function with minimum value of -1 for the point of interest $i = 1.35, v = 5.65$. So far, we used these tricks to reduce the number of unknowns for barriers.

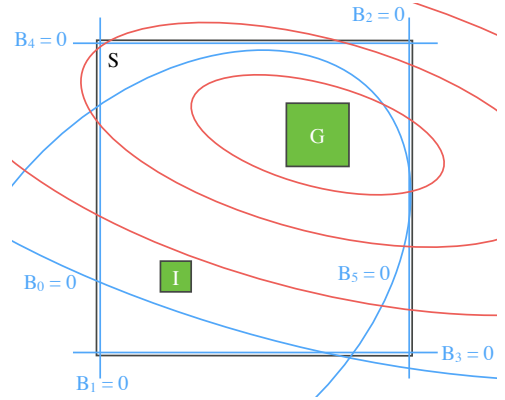


Figure 4: The blue lines are the barriers and the red lines are level-sets of the Lyapunov function.

Table 1: Results of Comparison with SCOTS on examples

Legend: n : # state variables, itr : # iterations, Time: total computation time, η : state discretization step, τ : time step. All timings are in seconds and rounded, TO: timeout (> 10 hours).

Problem		SCOTS				CEGIS	
ID	n	η	τ	itr	Time	δ	Time
Ex. 5	2	0.16^2	0.12	18	0	10^{-4}	3
Ex. 8	2	0.01^2	1.0	106	1	10^{-4}	39
Ex. 6	3	$0.2^2 \times 0.1$	0.3	404	989	10^{-4}	1484
Ex. 3	4	0.03×0.1^3	0.005	48	304	10^{-5}	3
Ex. 7	4	$0.1^2 \times 0.05^2$	0.3	TO		10^{-4}	5296

However, our method fails to find a certificate. Next, we add one more barrier (B_5) to the formulation and we use the following template for B_5 : $c_4(i - 0.9)^2 + c_5(i - 0.9)(v - 5.2) + c_6(v - 5.05)^2 - 1$, which is a quadratic function with minimum value of -1 for initial point $i = 0.9$, $v = 5.2$. This time, we can successfully find a control certificate. The final barriers and level-sets of the Lyapunov function is shown in Fig. 4.

Comparison: While abstraction based methods can provide a near optimal solution (are relatively complete), these methods can be computationally expensive. On the other hand, our method is a Lyapunov-based method and the solution is not necessarily (relatively) complete. For example, our approach assumes that control certificates with a given form (that is given as input by user) exist. As such, the existence of such certificates is not guaranteed and thus, our approach lacks the general applicability of a fixed-point based synthesis. Also, for initialized problems our method needs an initial set as input, while for the abstraction based methods, maximum controllable region can be obtained without the need for specifying the initial set. However, our method is relatively more scalable thanks to recent development in SMT solvers. Here, for the sake of completeness, we provide a brief comparison with SCOTS toolbox [25] for the examples provided in this article. To compare Example 3, we use *fat* facet and assume target set has a volume (otherwise, because of time discretization, SCOTS cannot find a solution). More precisely, we use target set $[1, 1.2] \times [-1, 1]^3$ instead of $[1, 1] \times [-1, 1]^3$.

All the experiments are ran on a laptop with Core i7 2.9 GHz CPU and 16GB of RAM. The results are reported in Table 1. We also note that if we use larger values for SCOTS parameters, SCOTS fails to solve these problems (initial set is not a subset of controllable region). Table 1 shows that SCOTS performs much better for Example 5 and 8 for which there are only 2 state variables. For Example 6, both methods have similar performances. And for Example. 3 and Example.7, which have 4 state variables, our method is faster.

6 Conclusions

In this paper, given a switched system, we addressed controller synthesis problems for RWS with composite barriers. Specifically, we addressed uninitialized problems which are useful for building an abstraction, as well as initialized problems. For each problem, we provided sufficient conditions in terms of “existence of a control certificate”. Also, we demonstrated that searching for a control certificate can be encoded into constrained problems and solving these problems is computationally feasible. In the future,

we wish to investigate how the initialized RWS problems can be extended to be used along fixed-point computation based techniques as it allows more flexible switching strategies.

Acknowledgments

This work was funded in part by NSF under award numbers SHF 1527075 and CPS 1646556. All opinions expressed are those of the authors and not necessarily of the NSF.

References

- [1] Zvi Artstein (1983): *Stabilization with relaxed controls*. *Nonlinear Analysis: Theory, Methods & Applications* 7(11), pp. 1163 – 1173, doi:10.1016/0362-546X(83)90049-4. Available at <http://www.sciencedirect.com/science/article/pii/0362546X83900494>.
- [2] Rayna Dimitrova & Rupak Majumdar (2014): *Deductive control synthesis for alternating-time logics*. In: *2014 International Conference on Embedded Software, EMSOFT 2014, New Delhi, India, October 12-17, 2014*, pp. 14:1–14:10, doi:10.1145/2656045.2656054.
- [3] Sicun Gao, Soonho Kong & Edmund M. Clarke (2013): *dReal: An SMT Solver for Nonlinear Theories over the Reals*. In: *Automated Deduction - CADE-24 - 24th International Conference on Automated Deduction, Lake Placid, NY, USA, June 9-14, 2013. Proceedings*, pp. 208–214, doi:10.1007/978-3-642-38574-2_14. Available at https://doi.org/10.1007/978-3-642-38574-2_14.
- [4] L. El Ghaoui & V. Balakrishnan (1994): *Synthesis of fixed-structure controllers via numerical optimization*. In: *Proceedings of 1994 33rd IEEE Conference on Decision and Control*, 3, pp. 2678–2683 vol.3, doi:10.1109/CDC.1994.411398.
- [5] A. Girard, G. Pola & P. Tabuada (2010): *Approximately Bisimilar Symbolic Models for Incrementally Stable Switched Systems*. *IEEE Transactions on Automatic Control* 55(1), pp. 116–126, doi:10.1109/TAC.2009.2034922.
- [6] L. C. G. J. M. Habets, P. J. Collins & J. H. van Schuppen (2006): *Reachability and control synthesis for piecewise-affine hybrid systems on simplices*. *IEEE Transactions on Automatic Control* 51(6), pp. 938–948, doi:10.1109/TAC.2006.876952.
- [7] L.C.G.J.M. Habets & J.H. van Schuppen (2004): *A control problem for affine dynamical systems on a full-dimensional polytope*. *Automatica* 40(1), pp. 21 – 35, doi:10.1016/j.automatica.2003.08.001. Available at <http://www.sciencedirect.com/science/article/pii/S0005109803002620>.
- [8] M. K. Helwa & M. E. Broucke (2011): *Monotonic reach control on polytopes*, pp. 4741–4746, doi:10.1109/CDC.2011.6160866.
- [9] Z. Huang, Y. Wang, S. Mitra, G. E. Dullerud & S. Chaudhuri (2015): *Controller synthesis with inductive proofs for piecewise linear systems: An SMT-based algorithm*. In: *2015 54th IEEE Conference on Decision and Control (CDC)*, pp. 7434–7439, doi:10.1109/CDC.2015.7403394.
- [10] Manuel Mazo Jr., Anna Davitian & Paulo Tabuada (2010): *PESSOA: A Tool for Embedded Controller Synthesis*. In: *Computer Aided Verification, 22nd International Conference, CAV 2010, Edinburgh, UK, July 15-19, 2010. Proceedings*, pp. 566–569, doi:10.1007/978-3-642-14295-6_49. Available at https://doi.org/10.1007/978-3-642-14295-6_49.
- [11] M. Kloetzer & C. Belta (2008): *A Fully Automated Framework for Control of Linear Systems from Temporal Logic Specifications*. *IEEE Transactions on Automatic Control* 53(1), pp. 287–297, doi:10.1109/TAC.2007.914952.
- [12] Hui Kong, Fei He, Xiaoyu Song, William N. N. Hung & Ming Gu (2013): *Exponential-Condition-Based Barrier Certificate Generation for Safety Verification of Hybrid Systems*. In: *Computer Aided Verification - 25th*

- International Conference, CAV 2013, Saint Petersburg, Russia, July 13-19, 2013. Proceedings*, pp. 242–257, doi:10.1007/978-3-642-39799-8_17. Available at https://doi.org/10.1007/978-3-642-39799-8_17.
- [13] Daniel Liberzon (2012): *Switching in systems and control*. Springer Science & Business Media, doi:10.1007/978-1-4612-0017-8.
- [14] Z. Lin & M. E. Broucke (2007): *Reachability and control of affine hypersurface systems on polytopes*. In: *2007 46th IEEE Conference on Decision and Control*, pp. 733–738, doi:10.1109/CDC.2007.4434805.
- [15] J. Liu, N. Ozay, U. Topcu & R. M. Murray (2013): *Synthesis of Reactive Switching Protocols From Temporal Logic Specifications*. *IEEE Transactions on Automatic Control* 58(7), pp. 1771–1785, doi:10.1109/TAC.2013.2246095.
- [16] Sebti Mouelhi, Antoine Girard & Gregor Gössler (2012): *CoSyMA: A Tool for Controller Synthesis Using Multi-scale Abstractions*. Research Report RR-8108, INRIA. Available at <https://hal.inria.fr/hal-00743982>.
- [17] Sebti Mouelhi, Antoine Girard & Gregor Gößler (2013): *CoSyMA: a tool for controller synthesis using multi-scale abstractions*. In: *Proceedings of the 16th international conference on Hybrid systems: computation and control, HSCC 2013, April 8-11, 2013, Philadelphia, PA, USA*, pp. 83–88, doi:10.1145/2461328.2461343.
- [18] Leonardo Mendonça de Moura & Nikolaj Bjørner (2008): *Z3: An Efficient SMT Solver*. In: *Tools and Algorithms for the Construction and Analysis of Systems, 14th International Conference, TACAS 2008, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2008, Budapest, Hungary, March 29-April 6, 2008. Proceedings*, pp. 337–340, doi:10.1007/978-3-540-78800-3_24. Available at https://doi.org/10.1007/978-3-540-78800-3_24.
- [19] P. Nilsson & N. Ozay (2014): *Incremental synthesis of switching protocols via abstraction refinement*. In: *53rd IEEE Conference on Decision and Control*, pp. 6246–6253, doi:10.1109/CDC.2014.7040368.
- [20] N. Ozay, J. Liu, P. Prabhakar & R. M. Murray (2013): *Computing augmented finite transition systems to synthesize switching protocols for polynomial switched systems*. In: *2013 American Control Conference*, pp. 6237–6244, doi:10.1109/ACC.2013.6580816.
- [21] S. Prajna, A. Papachristodoulou & P. A. Parrilo (2002): *Introducing SOSTOOLS: a general purpose sum of squares programming solver*. In: *Proceedings of the 41st IEEE Conference on Decision and Control, 2002.*, 1, pp. 741–746 vol.1, doi:10.1109/CDC.2002.1184594.
- [22] H. Ravanbakhsh & S. Sankaranarayanan (2015): *Counter-Example Guided Synthesis of control Lyapunov functions for switched systems*. In: *2015 54th IEEE Conference on Decision and Control (CDC)*, pp. 4232–4239, doi:10.1109/CDC.2015.7402879.
- [23] Hadi Ravanbakhsh & Sriram Sankaranarayanan (2016): *Robust Controller Synthesis of Switched Systems Using Counterexample Guided Framework*. In: *Proceedings of the 13th International Conference on Embedded Software, EMSOFT '16, ACM*, pp. 8:1–8:10, doi:10.1145/2968478.2968485.
- [24] Bartek Roszak & Mireille E. Broucke (2006): *Necessary and sufficient conditions for reachability on a simplex*. *Automatica* 42(11), pp. 1913 – 1918, doi:10.1016/j.automatica.2006.06.003. Available at <http://www.sciencedirect.com/science/article/pii/S0005109806002445>.
- [25] Matthias Rungger & Majid Zamani (2016): *SCOTS: A Tool for the Synthesis of Symbolic Controllers*. In: *Proceedings of the 19th International Conference on Hybrid Systems: Computation and Control, HSCC 2016, Vienna, Austria, April 12-14, 2016*, pp. 99–104, doi:10.1145/2883817.2883834.
- [26] Armando Solar Lezama (2008): *Program Synthesis By Sketching*. Ph.D. thesis, EECS Department, University of California, Berkeley. Available at <http://www2.eecs.berkeley.edu/Pubs/TechRpts/2008/EECS-2008-177.html>.
- [27] Eduardo D. Sontag (1989): *A universal construction of Artstein's theorem on nonlinear stabilization*. *Systems & Control Letters* 13(2), pp. 117 – 123, doi:10.1016/0167-6911(89)90028-5. Available at <http://www.sciencedirect.com/science/article/pii/0167691189900285>.

- [28] Ankur Taly, Sumit Gulwani & Ashish Tiwari (2011): *Synthesizing switching logic using constraint solving*. *STTT* 13(6), pp. 519–535, doi:10.1007/s10009-010-0172-8. Available at <https://doi.org/10.1007/s10009-010-0172-8>.
- [29] Ankur Taly & Ashish Tiwari (2010): *Switching logic synthesis for reachability*. In: *Proceedings of the 10th International conference on Embedded software, EMSOFT 2010, Scottsdale, Arizona, USA, October 24-29, 2010*, pp. 19–28, doi:10.1145/1879021.1879025.
- [30] Weehong Tan & Andrew Packard (2004): *Searching for control Lyapunov functions using sums of squares programming*. In: *Allerton conference on communication, control and computing*, pp. 210–219.
- [31] Y. Tazaki & J. i. Imura (2012): *Discrete Abstractions of Nonlinear Systems Based on Error Propagation Analysis*. *IEEE Transactions on Automatic Control* 57(3), pp. 550–564, doi:10.1109/TAC.2011.2161789.
- [32] Wolfgang Thomas, Thomas Wilke et al. (2002): *Automata, logics, and infinite games: a guide to current research*. 2500, Springer Science & Business Media, doi:10.1007/3-540-36387-4.
- [33] Peter Wieland & Frank Allgwer (2007): *CONSTRUCTIVE SAFETY USING CONTROL BARRIER FUNCTIONS*. *IFAC Proceedings Volumes* 40(12), pp. 462 – 467, doi:10.3182/20070822-3-ZA-2920.00076. Available at <http://www.sciencedirect.com/science/article/pii/S1474667016355690>. 7th IFAC Symposium on Nonlinear Control Systems.
- [34] T. Wongpiromsarn, U. Topcu & A. Lamperski (2016): *Automata Theory Meets Barrier Certificates: Temporal Logic Verification of Nonlinear Systems*. *IEEE Transactions on Automatic Control* 61(11), pp. 3344–3355, doi:10.1109/TAC.2015.2511722.
- [35] Tichakorn Wongpiromsarn, Ufuk Topcu, Necmiye Ozay, Huan Xu & Richard M. Murray (2011): *TuLiP: A Software Toolbox for Receding Horizon Temporal Logic Planning*. In: *Proceedings of the 14th International Conference on Hybrid Systems: Computation and Control, HSCC '11, ACM, New York, NY, USA*, pp. 313–314, doi:10.1145/1967701.1967747.
- [36] Xiangru Xu, Paulo Tabuada, Jessy W. Grizzle & Aaron D. Ames (2015): *Robustness of Control Barrier Functions for Safety Critical Control***This work is partially supported by the National Science Foundation Grants 1239055, 1239037 and 1239085. *IFAC-PapersOnLine* 48(27), pp. 54 – 61, doi:10.1016/j.ifacol.2015.11.152. Available at <http://www.sciencedirect.com/science/article/pii/S2405896315024106>. Analysis and Design of Hybrid Systems ADHS.
- [37] M. Zamani, G. Pola, M. Mazo & P. Tabuada (2012): *Symbolic Models for Nonlinear Control Systems Without Stability Assumptions*. *IEEE Transactions on Automatic Control* 57(7), pp. 1804–1809, doi:10.1109/TAC.2011.2176409.