Synthesizing Switching Controllers for Hybrid Systems by Generating Invariants*

Hengjun Zhao[1,2], Naijun Zhan[1], and Deepak Kapur[3]

[1] State Key Lab. of Comput. Sci., Institute of Software, CAS, Beijing, China
[2] University of Chinese Academy of Sciences, Beijing, China
[3] Dept. of Comput. Sci., University of New Mexico, Albuquerque, NM, USA
{zhaohj,znj}@ios.ac.cn, kapur@cs.unm.edu

Abstract. We extend a template-based approach for synthesizing switching controllers for semi-algebraic hybrid systems, in which all expressions are polynomials. This is achieved by combining a QE (quantifier elimination)-based method for generating invariants with a qualitative approach for predefining templates. Our synthesis method is relatively complete with regard to a given family of predefined templates. Using qualitative analysis, we discuss heuristics to reduce the numbers of parameters appearing in the templates. To avoid too much human interaction in choosing templates as well as the high computational complexity caused by QE, we further investigate applications of the SOS (sum-of-squares) relaxation approach and the template polyhedra approach in invariant generation, which are both supported by modern numerical solvers.

1 Introduction

Hybrid systems, in which computations proceed by continuous evolutions as well as discrete jumps simulating transitions from one mode to another mode, are often used to model devices controlled by computers in many application domains [1]. Combining ideas from state machines in computer science and control theory, formal analysis, verification and synthesis of hybrid systems have been an important area of active research. In verification problems, a given hybrid system is required to satisfy a desired safety property e.g. that the temperature of a nuclear reactor will never go beyond a maximum threshold, as it may cause serious economic, human and/or environmental damage, thus implying that the system will never enter any unsafe state. A synthesis problem is harder given that the focus is on designing a controller that ensures the given system will satisfy a safety requirement, reach a given set of states, or meet an optimality criterion, or a desired combination of these requirements.

Automata-theoretic and logical approaches have been primarily used for verification and synthesis of hybrid systems [2,4,45]. In [4,45], a general framework

* This work has been supported in part by the projects NSFC-91118007, National Science and Technology Major Project of China (Grant No. 2012ZX01039-004), NSF CCF 1248069 and NSF DMS 1217054.

Z. Liu, J. Woodcock, and H. Zhu (Eds.): Jifeng Festschrift, LNCS 8051, pp. 354–373, 2013.

for controller synthesis based on hybrid automata was proposed. This approach relies on *backward reachable set* computation and *fixed point iteration*, and thus has two main restrictions: (i) the computation of backward reachable set is hard for most continuous dynamics, and (ii) termination of the fixed point iteration process cannot be guaranteed, even for those hybrid systems whose backward reachable sets are effectively computable. Therefore most of the research, e.g. [14], focuses on overcoming the above two restrictions.

Recently, a deductive approach for verification and synthesis based on constraint solving was proposed in [11,28,27,41,21,40]. The central idea is to reduce verification and synthesis problems of hybrid systems to invariant generation problems, much like verification of programs. As proposed in [16,15,34], if invariants are hypothesized to be of certain shapes, then corresponding templates with associated parameters can be used and the invariant generation problem can be reduced to constraint solving over parameters by quantifier elimination. This methodology is used in [41] for synthesizing switching controllers meeting safety requirements, while in [43], the approach is extended for satisfying both safety and reachability requirements. A common problem with template-based method is that it heavily relies on a user specifying the shape of invariants that are of interest, thus making it interactive and user driven, raising doubts about its scalability and automation. Besides, the inference rules for inductive invariants in [42,41,43] are sound and complete for several classes of invariants, e.g. smooth, quadratic and convex invariants, but are not complete for generic semi-algebraic sets[1].

Inspired by [17,4,41] and [22], we extend in this paper the template-based invariant generation approach for synthesizing switching controllers of hybrid systems to meet given safety requirements. The paper makes the following contributions:

- We propose a method for synthesizing switching controllers for hybrid systems using a family of invariants, which could be different for different modes of a hybrid system. We use templates for invariant generation based on quantifier-elimination (QE) techniques combined with numerical methods.
- In the QE-based synthesis approach, we adopt the invariant generation method proposed in [22], which is proved to be sound and relatively complete with respect to a given shape of semi-algebraic invariants (i.e. a given family of predefined semi-algebraic templates). The advantage is that, compared to the invariant generation methods used in [42,41,43], there is more possibility of discovering invariants of the given shape.
- Using the qualitative approach proposed in [17] for analyzing continuous evolution in each mode of a hybrid system, we can identify those continuous states at which even small perturbation would lead to continuous evolution violating the safety requirement. Such continuous states are called critical control points, using which we can determine a more precise shape of

[1] A subset $A \subseteq \mathbb{R}^n$ is called *semi-algebraic* if there is a quantifier-free polynomial formula φ s.t. $A = \{\mathbf{x} \in \mathbb{R}^n \mid \varphi(\mathbf{x}) \text{ is true}\}$.

templates to be used as invariants, thus reducing the number of parameters appearing in templates.
- Quantifier elimination techniques have high complexity especially for cases when templates have lots of parameters. Even though qualitative analysis is helpful in bringing down the number of parameters and thus the complexity of QE, the paper also explores two kinds of predefined templates where numerical techniques can be exploited to improve the degree of automation and scalability. In particular, (i) for polynomial templates, using *sum-of-squares* (SOS) relaxation, the constraint on parameters appearing in templates is transformed into a *semi-definite program*(SDP), which is convex and can be solved efficiently; (ii) for linear systems and a special type of templates—template polyhedra, the invariant generation problem can be reduced to a BMI (*bilinear matrix inequality*) feasibility problem, which is also easier to solve (numerically) than QE.

Paper Structure. The rest of this paper is structured as follows. In Section 2 we formally define the switching controller synthesis problem for safety of hybrid automata. In Section 3 we introduce the notion of invariant in the context of hybrid system, and formulate an abstract solution to the controller synthesis problem using invariants; then we extend a template-based method for invariant generation to solve the controller synthesis problem, by combining quantifier elimination (QE) techniques and qualitative analysis. In Section 4 we investigate the application of two numerical approaches in switching controller synthesis by generating invariants numerically. We finally conclude the paper with some discussions by Section 5.

1.1 Related Work

Our work in this paper resembles [41] but differs in that: i) our method is cast in the setting of hybrid automata, and therefore rather than generating a single global controlled invariant, we searches for a family of invariants that refine the domain of each mode of the original hybrid automata; ii) a sound and complete criterion is used in invariant generation; iii) various techniques are applied for scalability.

The SOS relaxation approach has been successfully used in safety verification of hybrid systems. In [31,32], the authors used the SOSTOOLS software package [33] to compute *barrier certificates* for polynomial hybrid systems. In [20,48], the authors proposed a hybrid symbolic-numeric approach to compute exact inequality invariants of hybrid systems, by first solving (bilinear) SOS programming numerically and then applying *rational vector recovery* techniques.

A necessary and sufficient condition for positive invariance of convex polyhedra for linear continuous systems was provided in [7]. This condition is extended to linear systems with open polyhedral domain for our need in this paper. Template polyhedra were used in [36,35] to compute positive invariants of hybrid systems by *policy iteration*, which differs from our treatment of the problem using BMI; besides, unlike [35], we do not require the polyhedral invariant to be

generated to have the same shape as the domain. Recently, a method for computing polytopic invariants for polynomial dynamical systems using template polyhedra and linear programming was proposed [38].

Mathematical programming techniques and relevant numerical solvers have also been widely applied to static program analysis. Actually, the template polyhedra abstract domain was first proposed in [37] to generate linear program invariants using linear programming. In [8], to verify invariance and termination of semi-algebraic programs, verification conditions are abstracted into numerical constraints using Lagrangian relaxation or SOS relaxation, which are then resolved by efficient SDP solvers.

In our recent work [49], we studied an optimal switching controller synthesis problem arising from an industrial oil pump system with piece-wise constant continuous dynamics. A hybrid approach combining symbolic computation with numerical computation was developed to synthesize safe controllers with better optimal values.

2 Hybrid Systems and Switching Controller Synthesis Problem

We use hybrid automata to model hybrid systems.

Definition 1 (Hybrid Automaton). *A hybrid automaton (HA) is a system* $\mathcal{H} \widehat{=} (Q, X, f, D, E, G)$*, where*

- $Q = \{q_1, \ldots, q_m\}$ *is a finite set of modes;*
- $X = \{x_1, \ldots, x_n\}$ *is a finite set of continuous state variables, with* $\mathbf{x} = (x_1, \ldots, x_n)$ *ranging over* $\mathbb{R}^n$*;* $Q \times \mathbb{R}^n$ *is the state space of* $\mathcal{H}$*;*
- $f : Q \to (\mathbb{R}^n \to \mathbb{R}^n)$ *assigns to each mode* $q \in Q$ *a vector field* $\mathbf{f}_q$*;*
- $D : Q \to 2^{\mathbb{R}^n}$ *assigns to each mode* $q \in Q$ *a domain* $D_q \subseteq \mathbb{R}^n$*;*
- $E \subseteq Q \times Q$ *is a set of discrete transitions;*
- $G : E \to 2^{\mathbb{R}^n}$ *assigns to each transition* $e \in E$ *a switching guard* $G_e \subseteq \mathbb{R}^n$ *.*

Compared with the conventional versions of HA as in [2], in Definition 1 we make the following assumptions:

- for all $q \in Q$, $\mathbf{f}_q$ is a *polynomial* vector function, and thus the existence and uniqueness of solutions to $\dot{\mathbf{x}} = \mathbf{f}_q$ is guaranteed; besides, $\mathbf{f}_q$ is required to be a *complete*[2] vector field, that is, for any $\mathbf{x}_0 \in \mathbb{R}^n$, the solution $\mathbf{x}(t)$ to $\dot{\mathbf{x}} = \mathbf{f}_q$ exists for all $t \in [0, \infty)$; however, unlike [17], no assumption is made about whether a closed form solution to $\dot{\mathbf{x}} = \mathbf{f}_q$ exists;
- for all $q \in Q$ and all $e \in E$, D_q and G_e are *closed* semi-algebraic sets;
- the initial set of each mode is identical with the domain, and thus omitted;
- all resets are assumed to be identity mappings for ease of presentation, but can also be generalized to polynomial functions.

[2] This assumption is used in the proof of Theorem 1, to exclude the possibility that a hybrid system is blocked due to the inextensibility of trajectories defined by $\mathbf{f}$.

We use a nuclear reactor system discussed in [3,12,17] as a running example throughout this paper.

Example 1. The nuclear reactor system consists of a reactor core and a cooling rod which is immersed into and removed out of the core periodically to keep the temperature of the core, denoted by x, in a certain range. Denote the fraction of the rod immersed into the reactor by θ. Then the initial specification of this system can be represented using the hybrid automaton in Fig. 1.

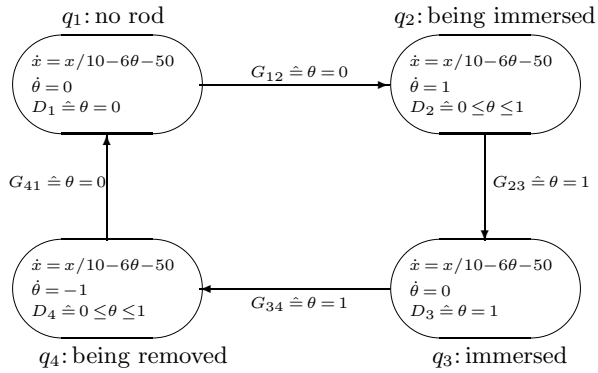

Fig. 1. Nuclear reactor temperature control

The semantics of a hybrid automaton $\mathcal{H}$ can be defined by the set of trajectories it accepts. For the formal definitions of *hybrid time set* and *hybrid trajectory* the readers are referred to [45].

The domain of a hybrid automaton $\mathcal{H}$ is defined as $D_{\mathcal{H}} \widehat{=} \bigcup_{q \in Q}(\{q\} \times D_q)$. We call $\mathcal{H}$ *non-blocking* if for any $(q, \mathbf{x}) \in D_{\mathcal{H}}$, there is a hybrid trajectory from $(q, \mathbf{x})$ which can either be extended to infinite time $t = \infty$ or execute infinitely many discrete transitions; otherwise $\mathcal{H}$ is called *blocking*.

A *safety requirement* $\mathcal{S}$ assigns to each mode $q \in Q$ a safe region $S_q \subseteq \mathbb{R}^n$, i.e. $\mathcal{S} = \bigcup_{q \in Q}(\{q\} \times S_q)$. Alternatively, there could be a global safe region S which all modes are required to satisfy, i.e. $S_q = S$ for all $q \in Q$.

One way of formulating a switching controller synthesis problem for meeting a safety requirement can be precisely defined as follows [4].

Problem 1 (Controller Synthesis for Safety). Given a hybrid automaton $\mathcal{H}$ and a safety property S, find a hybrid automaton $\mathcal{H}' = (Q, X, f, D', E, G')$ such that

(r1) Refinement: for any $q \in Q$, $D'_q \subseteq D_q$, and for any $e \in E$, $G'_e \subseteq G_e$;
(r2) Safety: for any trajectory ω that $\mathcal{H}'$ accepts, if $(q, \mathbf{x})$ is on ω, then $\mathbf{x} \in S_q$;
(r3) Non-blocking: $\mathcal{H}'$ is non-blocking.

If such $\mathcal{H}'$ exists, then $\mathcal{SC} \widehat{=} \{G'_e \subseteq \mathbb{R}^n \mid e \in E\}$ is a safe *switching controller* associated with the set of transitions E, and $D_{\mathcal{H}'} \widehat{=} \bigcup_{q \in Q}(\{q\} \times D'_q)$ is the *controlled invariant set* rendered by $\mathcal{SC}$.

Informally, the switching controller synthesis problem reduces to identifying a set of continuous states for each transition, only at which the system is allowed

to switch from one mode to another, guaranteeing that the system satisfies the safety requirements imposed on every mode as well as can run forever.

3 A QE-Based Approach

In this section, we propose a quantifier elimination (QE) based approach for synthesizing a switching controller for a hybrid automaton by integrating heuristics based on qualitative analysis [17] for predefining templates of invariants, into a relatively complete method for generating semi-algebraic invariants for polynomial continuous dynamical systems with domain [22]. Below we first review the concept of invariant used in [22] based on a related concept in [28].

3.1 Invariants for Continuous Dynamical System with Domain

The notion of *positively invariant set* plays a very important role in the study of continuous dynamical systems [5].

Definition 2. *A subset $P \subseteq \mathbb{R}^n$ is called a positively invariant set for a system $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$, if for all $\mathbf{x}_0 \in P$, the solution $\mathbf{x}(t)$ to $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$ starting from $\mathbf{x}_0$ satisfies $\mathbf{x}(t) \in P$ for all $t > 0$.*

However, the above concept of invariant is not suitable for the study of hybrid systems. The reason is that each mode of a hybrid automaton $\mathcal{H}$ can be abstracted as a pair $(D, \mathbf{f})$, where D and $\mathbf{f}$ are the domain and vector field of a certain mode of $\mathcal{H}$; for any trajectory $\mathbf{x}(t)$ of $\mathbf{f}$, only the part of $\mathbf{x}(t)$ that lies in D is meaningful to the behavior of $\mathcal{H}$, rather than the complete trajectory with all $t > 0$ as in Definition 2. Therefore the following concept of invariant is proposed for systems like $(D, \mathbf{f})$.

Definition 3. *A subset $P \subseteq \mathbb{R}^n$ is called an invariant of $(D, \mathbf{f})$, if for all $\mathbf{x}_0 \in P$ and all $T \geq 0$, the solution $\mathbf{x}(t)$ of $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$ over $[0, T]$ with $\mathbf{x}(0) = \mathbf{x}_0$ satisfies*

$$(\forall t \in [0,T].\, \mathbf{x}(t) \in D) \longrightarrow (\forall t \in [0,T].\, \mathbf{x}(t) \in P) \ .$$

Intuitively, P is an invariant of $(D, \mathbf{f})$ if any continuous evolution starting from P stays in P as long as it stays in D. If $D = \mathbb{R}^n$, then an invariant of $(D, \mathbf{f})$ is a positively invariant set of $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$ as defined in Definition 2; otherwise if D is a proper subset of $\mathbb{R}^n$, then generally the notion of invariant in Definition 3 is weaker, and thus allows a broader class of sets to be invariants.

Example 2. Suppose $D \mathrel{\widehat{=}} x > 0$ and $\mathbf{f} = (-y, x)$. It can be shown (please to the full version [18]) that $P \mathrel{\widehat{=}} y \geq 0$ is not a positively invariant set of $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$, but is an invariant of $(D, \mathbf{f})$.

The above arguments show that when dealing with continuous evolutions in the context of hybrid system, it is necessary to study invariants for augmented systems $(D, \mathbf{f})$, rather than pure continuous dynamical systems $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$.

3.2 The Abstract Synthesis Procedure

Solving Problem 1 amounts to refining the domains and guards of $\mathcal{H}$ by removing so-called *bad* states. A state $(q, \mathbf{x}) \in D_{\mathcal{H}}$ is *bad* if the hybrid trajectory starting from $(q, \mathbf{x})$ either blocks $\mathcal{H}$ or violates $\mathcal{S}$; otherwise, it is called a *good* state. From Definition 3, we observe that the set of good states of $\mathcal{H}$ can be approximated using invariants, which results in the following solution to Problem 1.

Theorem 1. *Let $\mathcal{H}$ and $\mathcal{S}$ be as in Problem 1. Further, for each $q \in Q$, let D'_q be a closed subset of $\mathbb{R}^n$ such that $\bigcup_{q\in Q} D'_q$ is non-empty (to imply at least one D'_q is non-empty). If we have*

(c1) for all $q \in Q$, $D'_q \subseteq D_q \cap S_q$;
(c2) for all $q \in Q$, D'_q is an invariant of $(H_q, \mathbf{f}_q)$ with

$$H_q \widehat{=} \Big(\bigcup_{e=(q,q')\in E} G'_e \Big)^c,$$

where $G'_e \widehat{=} G_e \cap D'_{q'}$ for $e = (q, q')$, and A^c denotes the complement of A in $\mathbb{R}^n$, then the HA $\mathcal{H}' = (Q, X, f, D', E, G')$ is a solution to Problem 1.

Proof. Please refer to the full version of this paper [18]. □

In Theorem 1, condition (c1) ensures that D'_q is a refinement of D_q and mode q satisfies its safety condition, thus guaranteeing (r1) and (r2) of Problem 1; condition (c2) requires that any trajectory starting in mode q will either remain in mode q or jump to another mode q' when the associated guard is satisfied, thus guaranteeing (r3) of Problem 1.

Based on Theorem 1, we give below the steps of a template-based method for synthesizing a switching controller.

(s1) **Template assignment:** assign to each $q \in Q$ a template parametrically specifying D'_q, which will be required (see step (s3)) to be a refinement of D_q, as well as the invariant to be generated at mode q;
(s2) **Guard refinement:** refine guard G_e for each $e = (q, q') \in E$ by setting $G'_e \widehat{=} G_e \cap D'_{q'}$;
(s3) **Deriving synthesis conditions:** encode (c1) and (c2) in Theorem 1 into constraints on parameters appearing in the templates;
(s4) **Constraint solving:** solve the constraints derived from (s3) in terms of the parameters;
(s5) **Parameters instantiation:** find an appropriate instantiation of D'_q and G'_e such that D'_q are closed[3] sets for all $q \in Q$, and D'_q is non-empty[4] for at least one $q \in Q$; if such an instantiation is not found, we choose a new set of templates and go back to (s1).

[3] This can be enforced by restricting to $\geq, \leq, =$ and $\vee, \wedge$ symbols in templates.
[4] To avoid trivially generating an empty set, some additional constraints can be encoded in step (s3).

We have assumed the hybrid automata to be specified by polynomial expressions. If in addition we restrict the form of safety requirements and templates to polynomial formulas, then computability of the above abstract procedure is guaranteed by Tarski's result [44].

In (s3), condition (c1) can be encoded into a first-order polynomial formula straightforwardly; encoding of (c2) into first-order polynomial constraints is based on our previous work in [22] on a relatively complete method for generating invariants (see Section 3.3). We use *quantifier elimination* (QE) to solve the first-order polynomial constraints obtained in (s4).

The shape of chosen templates in (s1) determines the likelihood of success of the above procedure, as well as the complexity of QE in (s4). In Section 3.5, we discuss heuristics for choosing appropriate templates using the *qualitative analysis* discussed in [17].

3.3 A Relatively Complete Method for Generating Invariants

In [22] we presented a sound and relatively complete approach for generating *semi-algebraic* invariants for $(D, \mathbf{f})$ with semi-algebraic domain D and polynomial vector function $\mathbf{f}$. For self-containedness, we introduce below the main result in the simplest case. For details the readers can consult [22].

Given a polynomial $p(\mathbf{x})$ and a polynomial vector field $\mathbf{f}(\mathbf{x})$, define the *Lie derivatives* of p along $\mathbf{f}$, $L^k_{\mathbf{f}}p : \mathbb{R}^n \longrightarrow \mathbb{R}$ for $k \in \mathbb{N}$, as follows:

- $L^0_{\mathbf{f}}p(\mathbf{x}) = p(\mathbf{x})$;
- $L^k_{\mathbf{f}}p(\mathbf{x}) = \langle \nabla L^{k-1}_{\mathbf{f}}p(\mathbf{x}), \mathbf{f}(\mathbf{x})\rangle$, for $n > 0$,

where $\nabla g(\mathbf{x})$ denotes the *gradient* vector of a scalar function $g(\mathbf{x})$, and $\langle \cdot, \cdot \rangle$ is the *inner product* of two vectors.

Example 3. Suppose $\mathbf{f} = (1, 1)$ and $p(x, y) = -x^2 + y$. Then $L^0_{\mathbf{f}}p(x, y) = -x^2 + y$, $L^1_{\mathbf{f}}p(x, y) = -2x + 1$, $L^2_{\mathbf{f}}p(x, y) = -2$, and $L^k_{\mathbf{f}}p(x, y) = 0$ for all $k \geq 3$.

The importance of Lie derivatives is that they can be used to predict continuous evolutions of $\mathbf{f}$ in terms of a polynomial p. To illustrate this, look at Fig. 2 showing the vector field $\mathbf{f}$ (small arrows) and the semi-algebraic set $P \mathrel{\widehat{=}} p \geq 0$ (grey area), with $\mathbf{f}$ and p defined in Example 3. At point $A_0(-1, 1)$ on the boundary of P, the first-order Lie derivative $L^1_{\mathbf{f}}p(-1, 1) = 3 > 0$, indicating that the angle between the vector field $(1, 1)$ (arrow $\overrightarrow{A_0A_1}$), and the gradient $\nabla p(-1, 1) = (2, 1)$ (arrow $\overrightarrow{A_0A_2}$) is less that $\frac{\pi}{2}$, which further indicates that the trajectory of $\mathbf{f}$ from A_0 (arrow $\overrightarrow{A_0A_3}$) would move towards the $p > 0$ side.

At point $B_0(\frac{1}{2}, \frac{1}{4})$, $L^1_{\mathbf{f}}p(\frac{1}{2}, \frac{1}{4}) = 0$ indicates that the vector field $\overrightarrow{B_0B_1}$ is orthogonal to the gradient $\overrightarrow{B_0B_2}$, from which we cannot tell how the trajectory from B_0 (arrow $\overrightarrow{B_0B_3}$) evolves with respect to P. However, if we resort to higher order Lie derivatives, then from $L^2_{\mathbf{f}}p(\frac{1}{2}, \frac{1}{4}) = -2 < 0$ we assert that $\overrightarrow{B_0B_3}$ would go out of P into the $p < 0$ side immediately.

Generally, given p and $\mathbf{f}$, to make predictions as above at a point $\mathbf{x} \in \mathbb{R}^n$, we need to compute $L^0_{\mathbf{f}}p(\mathbf{x}), L^1_{\mathbf{f}}p(\mathbf{x}), \ldots$ to get the first $k \in \mathbb{N}$ s.t. $L^k_{\mathbf{f}}p(\mathbf{x}) \neq 0$.

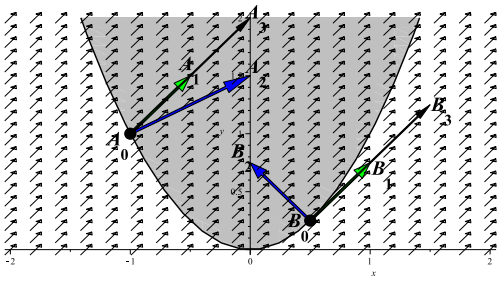

Fig. 2. Using Lie derivatives to predict continuous evolution

Furthermore, we can compute an integer $N_{p,\mathbf{f}}$ from p and $\mathbf{f}$ such that if all Lie derivatives with order $\leq N_{p,\mathbf{f}}$ evaluate to 0 at $\mathbf{x}$, then $L^k_{\mathbf{f}}(\mathbf{x}) = 0$ for all $k \in \mathbb{N}$. As a result, it suffices to compute Lie derivatives up to the $N_{p,\mathbf{f}}$-th order.

Formally, we have

Theorem 2. *Given a system* $(D, \mathbf{f})$ *with* $D \mathrel{\widehat{=}} h(\mathbf{x}) > 0$*, it has an invariant of the form* $P \mathrel{\widehat{=}} p(\mathbf{x}) \geq 0$ *if and only if* $\forall \mathbf{x}.\big(p(\mathbf{x}) = 0 \wedge h(\mathbf{x}) > 0 \longrightarrow \psi(p, \mathbf{f})\big)$*, where*

$$\psi(p,\mathbf{f}) \mathrel{\widehat{=}} \begin{array}{ll} & L^1_{\mathbf{f}}p(\mathbf{x}) > 0 \\ \vee & L^1_{\mathbf{f}}p(\mathbf{x}) = 0 \wedge L^2_{\mathbf{f}}p(\mathbf{x}) > 0 \\ \vee & \cdots \\ \vee & L^1_{\mathbf{f}}p(\mathbf{x}) = 0 \wedge \cdots \wedge L^{N_{p,\mathbf{f}}-1}_{\mathbf{f}}p(\mathbf{x}) = 0 \wedge L^{N_{p,\mathbf{f}}}_{\mathbf{f}}p(\mathbf{x}) > 0 \\ \vee & L^1_{\mathbf{f}}p(\mathbf{x}) = 0 \wedge \cdots \wedge L^{N_{p,\mathbf{f}}-1}_{\mathbf{f}}p(\mathbf{x}) = 0 \wedge L^{N_{p,\mathbf{f}}}_{\mathbf{f}}p(\mathbf{x}) = 0 \end{array}.$$

Proof. Please refer to [22]. □

The above theorem can be generalized for parametric polynomials $p(\mathbf{u}, \mathbf{x})$, thus enabling us to use polynomial templates and QE to automatically discover invariants. Such a method for invariant generation is relatively complete, that is, if there exists invariants in the form of the predefined template, then we are able to find one.

3.4 Comparison with Other Invariant Generation Approaches

In Platzer et al's work [28,27,30,29] and Tiwari et al's work [11,42,41], various criteria are proposed for checking invariants for systems $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$ or $(D, \mathbf{f})$. We will show the strength of our criterion due to its completeness through the following comparison.

Consider the system $(\mathbb{R}^2, \mathbf{f})$ from [42] with $\mathbf{f} \mathrel{\widehat{=}} (1-y, x)$. It can be shown that this system has an invariant $p \geq 0$ with $p \mathrel{\widehat{=}} -(-x^2 - y^2 + 2y)^2$. Geometrically, the trajectories of $\mathbf{f}$ are all circles centered at $(0, 1)$. The set $p \geq 0$, or equivalently $p = 0$ is actually one of these circles with radius 1, thus an invariant.

In [42], sound and complete inference rules are given for invariants that are linear, quadratic, smooth or convex. However, it was also pointed out in [42] that all these rules failed to prove the invariance property of $p \geq 0$, for in this example $p \geq 0$ is not linear or quadratic, nor is it smooth or convex. Furthermore, by a simple computation we get $L_{\mathbf{f}}^k p \equiv 0$ for all $k \geq 1$, so the sound but incomplete rule in [42,41] which involves only strict inequalities over finite-order Lie derivatives is also inapplicable. However, from $L_{\mathbf{f}}^1 p \equiv 0$ we get[5] $N_{p,\mathbf{f}} = 0$, and then according to Theorem 2, $p \geq 0$ can be verified since $\forall x \forall y.\ \big(-(-x^2 - y^2 + 2y)^2 = 0 \longrightarrow \mathsf{true}\big)$ holds trivially.

Although the rule in [28] can also be used to check the invariant $p \geq 0$, generally it only works on very restricted invariants. Even for linear systems like $(\mathbb{R}, \dot{x} = x)$, it cannot prove the invariant $x \geq 0$ because the verification condition $\forall x.x \geq 0$ is obviously *false*; whereas our approach requires $\forall x.(x = 0 \rightarrow \mathsf{true})$ (for this example $N_{p,\mathbf{f}}$=0 so $\psi(p, \mathbf{f})$ in Theorem 2 is true), which is trivially *true*.

Intuitively, to prove that $p \geq 0$ is an invariant of $(D, \mathbf{f})$, the rule in [28] requires $L_{\mathbf{f}}^1 p \geq 0$ over the whole domain D, while the rule in [42,41] requires that on the boundary of $p \geq 0$ inside D, the first non-zero high order Lie derivative, say $L_{\mathbf{f}}^k p$, is strictly positive. Completeness is lost either because non-boundary points are unnecessarily examined, or an upper bound on the order of Lie derivatives to be considered (the number $N_{p,\mathbf{f}}$ in our rule) is not given.

The above analysis shows the generality of our approach, using which it is possible to generate invariants in many general cases, and hence gives more possibility to synthesize a controller based on our understandings of the kind of controllers that can be synthesized using methods in [41,43,40].

3.5 Heuristics for Predefining Templates

The key steps of the qualitative analysis used in [17] are as follows.

1. Infer the evolution behavior (increasing or decreasing) of continuous variables in each mode from the differential equations (using first or second order derivatives).
2. Identify modes at which the evolution behavior (increasing or decreasing) of a continuous variable changes, and thus the maximal (or minimal) value of this continuous variable can be achieved. Such modes are called *control critical* modes.
3. At a control critical mode, equate the maximal (or minimal) value of a continuous variable to the corresponding safety upper (or lower) bound to obtain a specific continuous state, called a *critical point*.
4. At a control critical mode, use the critical point as the initial value to compute a closed form solution of the differential equation at that mode; then backtrack along this solution to compute a switching point which evokes a transition leading to the control critical mode.
5. The above obtained switching point is chosen as a new critical point, which is then backward propagated to other modes in a similar way.

[5] How to compute $N_{p,\mathbf{f}}$ for polynomial functions p and $\mathbf{f}$ can be found in [22].

Next, we illustrate how such an analysis helps in predefining templates for the running example.

Example 4 (Nuclear Reactor Temperature Control). Our goal is to synthesize a switching controller for the system in Example 1 with the global safety requirement that the temperature of the core lies between 510 and 550, i.e. $S_i \mathrel{\widehat{=}} 510 \leq x \leq 550$ for $i = 1, 2, 3, 4$. Please refer to Fig. 3 to get a better understanding of following discussions.

1) **Refine domains.** Using the safety requirement, domains D_i for $i = 1, 2, 3, 4$ are refined by $D_i^s \mathrel{\widehat{=}} D_i \cap S_i$, e.g. $D_1^s \mathrel{\widehat{=}} \theta = 0 \wedge 510 \leq x \leq 550$.
2) **Infer continuous evolutions.** Let $l_1 \mathrel{\widehat{=}} x/10 - 6\theta - 50 = 0$ be the *zero-level* set of $\dot{x}$ and check how x and θ evolve in each mode. For example, in D_2^s, $\dot{x} > 0$ on the left of l_1 and $\dot{x} < 0$ on the right; since θ increases from 0 to 1, x first increases then decreases and achieves maximal value when crossing l_1.
3) **Identify control critical modes.** By 2), q_2 and q_4 are control critical modes at which the evolution direction of x changes and the maximal (or minimal) value of x is achieved.
4) **Generate critical points.** By 3), we can get a critical point $E(5/6, 550)$ at q_2 by taking the intersection of l_1 and the safety upper bound $x = 550$; and $F(1/6, 510)$ can be obtained similarly at q_4.
5) **Propagate critical points.** E is backward propagated to $A(0, a)$ using the trajectory $\widehat{AE}$ through E defined by $\mathbf{f}_{q_2}$, and then to $C(1, c)$ using the trajectory $\widehat{CA}$ through A defined by $\mathbf{f}_{q_4}$; similarly, by propagating F we get D and B.
6) **Construct templates.** For brevity, we only show how to construct D_2'. Intuitively, $\theta = 0$, $\theta = 1$, $\widehat{AE}$ and $\widehat{BD}$ form the boundaries of D_2'. In order to get a semi-algebraic template, we need to fit $\widehat{AE}$ and $\widehat{BD}$ (which are generally not polynomial curves) by polynomials using points A, E and B, D respectively. By the inference of 2), $\widehat{AE}$ has only one extreme point (also the maximum point) E in D_2^s, and is tangential to $x = 550$ at E. A simple algebraic curve that can exhibit a shape similar to $\widehat{AE}$ is the parabola through A, E opening downward with $l_2 \mathrel{\widehat{=}} \theta = \frac{5}{6}$ the axis of symmetry. Therefore to minimize the degree of terms appearing in templates, we do not resort to polynomials with degree greater than 2. This parabola can be computed using the coordinates of A, E as: $x - 550 - \frac{36}{25}(a - 550)(\theta - \frac{5}{6})^2 = 0$, with a the parameter to be determined.

Through the above analysis, we generate the following templates:

- $D_1' \mathrel{\widehat{=}} \theta = 0 \wedge 510 \leq x \leq a$;
- $D_2' \mathrel{\widehat{=}} 0 \leq \theta \leq 1 \wedge x - b \geq \theta(d - b) \wedge x - 550 - \frac{36}{25}(a - 550)(\theta - \frac{5}{6})^2 \leq 0$;
- $D_3' \mathrel{\widehat{=}} \theta = 1 \wedge d \leq x \leq 550$;
- $D_4' \mathrel{\widehat{=}} 0 \leq \theta \leq 1 \wedge x - a \leq \theta(c - a) \wedge x - 510 - \frac{36}{25}(d - 510)(\theta - \frac{1}{6})^2 \geq 0$,

in which a, b, c, d are parameters satisfying

$$510 \leq b \leq a \leq 550 \wedge 510 \leq d \leq c \leq 550\,.$$

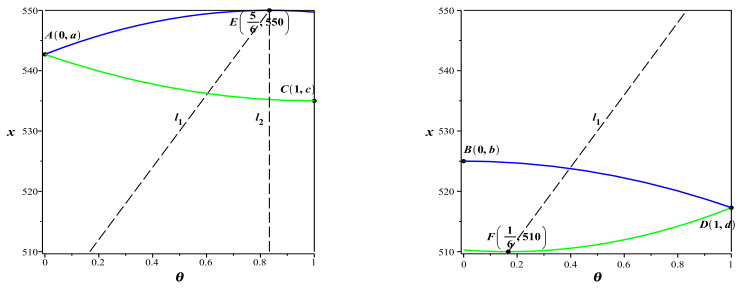

Fig. 3. Predefining templates via qualitative analysis

Note that without qualitative analysis, a single generic *quadratic* polynomial over θ and x would require $\binom{2+2}{2} = 6$ parameters.

Based on the framework presented in Section 3.2, we show below how to synthesize a switching controller for the system in Example 4 step by step.

Example 5 (Nuclear Reactor Temperature Control Contd.).

(s1) The four templates are defined in Example 4.

(s2) The four guards are refined by $G'_{ij} \mathrel{\widehat{=}} G_{ij} \cap D'_j$ and then simplified to:
- $G'_{12} \mathrel{\widehat{=}} \theta = 0 \wedge b \leq x \leq a$;
- $G'_{23} \mathrel{\widehat{=}} \theta = 1 \wedge d \leq x \leq 550$;
- $G'_{34} \mathrel{\widehat{=}} \theta = 1 \wedge d \leq x \leq c$;
- $G'_{41} \mathrel{\widehat{=}} \theta = 0 \wedge 510 \leq x \leq a$.

(s3) Based on Theorem 1 and a general version of Theorem 2 [22], we can derive the synthesis condition, which is a first-order polynomial formula in the form of $\phi \mathrel{\widehat{=}} \forall x \forall \theta . \varphi(a, b, c, d, x, \theta)$. We do not include ϕ here due to its big size.

(s4) By applying QE to ϕ, we get the following solution to the parameters:[6]

$$a = \frac{6575}{12} \wedge b = \frac{4135}{8} \wedge c = \frac{4345}{8} \wedge d = \frac{6145}{12} \ . \tag{1}$$

(s5) Instantiate D'_i and G'_{ij} by (1). It is obvious that all D'_i are nonempty closed sets. According to Theorem 1, we get a safe switching controller for the nuclear reactor system, i.e.
- $G'_{12} \mathrel{\widehat{=}} \theta = 0 \wedge 4135/8 \leq x \leq 6575/12$;
- $G'_{23} \mathrel{\widehat{=}} \theta = 1 \wedge 6145/12 \leq x \leq 550$;
- $G'_{34} \mathrel{\widehat{=}} \theta = 1 \wedge 6145/12 \leq x \leq 4345/8$;
- $G'_{41} \mathrel{\widehat{=}} \theta = 0 \wedge 510 \leq x \leq 6575/12$.

In [17], an upper bound $x = 547.97$ for G_{12} and a lower bound $x = 512.03$ for G_{34} are obtained by solving the differential equations at mode q_2 and q_4 respectively.

[6] The process of applying QE and selecting a sample solution demands some human effort which can be found in the full version of this paper [18].

By (1), the corresponding bounds generated here are $x \leq \frac{6575}{12} = 547.92$ and $x \geq \frac{6145}{12} = 512.08$.

As should be evident from the above results, in contrast to [17], where differential equations are solved to get closed-form solutions, we are able to get good results without requiring closed-form solutions. This indicates that our approach should work well for hybrid automata where differential equations for modes need not have closed form solutions.

4 Speeding Computations Using Numerical Methods on Specialized Templates

The QE-based approach crucially depends upon quantifier elimination techniques. It is well known that the complexity of a general purpose QE method over the full theory of real-closed fields is *doubly exponential* in the number of variables [9]. Therefore it is desirable to develop heuristics to do QE more efficiently. As shown in Section 3.5, qualitative analysis helps in reducing the number of parameters in templates. Another possible way to address the issue of high computational cost is resorting to numerical methods. In this section, we will discuss the application of two such approaches on specialized templates.

4.1 The SOS Relaxation Approach

Let $\mathbb{R}[x_1, x_2, \ldots, x_n]$, or $\mathbb{R}[\mathbf{x}]$ for short, denote the polynomial ring over variables $x_1, x_2, \ldots, x_n$ with real coefficients. A *monomial* is an expression in the form of $x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_n^{\alpha_n}$ with $(\alpha_1, \alpha_2, \ldots, \alpha_n) \in \mathbb{N}^n$. A *polynomial* $p(\mathbf{x}) \in \mathbb{R}[\mathbf{x}]$ of degree d can be written as a linear combination of $\binom{n+d}{d}$ monomials, i.e.

$$p(\mathbf{x}) = \sum_{\alpha_1+\alpha_2+\cdots+\alpha_n \leq d} c_{(\alpha_1,\alpha_2,\ldots,\alpha_n)} \cdot x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_n^{\alpha_n} \quad .$$

We call p an SOS (sum-of-squares) if there exist s polynomials $q_1, q_2, \ldots, q_s$ s.t.

$$p = \sum_{1 \leq i \leq s} q_i^2 \quad .$$

It is obvious that any SOS p is non-negative, i.e. $\forall \mathbf{x} \in \mathbb{R}^n . p(\mathbf{x}) \geq 0$.

The basic idea of SOS relaxation is as follows: to prove that a polynomial p is nonnegative, it suffices to show that p can be decomposed into a sum of squares, a trivially sufficient condition for non-negativity (but generally not necessary); similarly, to prove $p \geq 0$ on the semi-algebraic set $q \geq 0$, it is sufficient to find two SOS r_1, r_2 such that $p = r_1 + r_2 \cdot q$.

SOS relaxation is attractive because SOS decomposition can be reduced to a semi-definite programming (SDP) problem according to the following equivalence [26]:

> A polynomial p of degree $2d$ is an SOS if and only if there exists a semi-definite matrix Q such that $p = \mathbf{q} \cdot Q \cdot \mathbf{q}^T$, where $\mathbf{q}$ is a $\binom{n+d}{d}$-dimensional row vector of monomials with degree $\leq d$.

SDP is a convex programming that is solvable in *polynomial* time using numerical methods such as the interior point method [47]. Therefore the searching for SOS is a tractable problem.

We now show how SOS can be related to invariant generation. Let $p \geq 0$ be a parametric template defined for the system $(h > 0, \mathbf{f})$. By Theorem 2, a sufficient condition for $p \geq 0$ to be an invariant of $(h > 0, \mathbf{f})$ is

$$\forall \mathbf{x}. \Big(p(\mathbf{x}) = 0 \wedge h(\mathbf{x}) > 0 \longrightarrow L^1_{\mathbf{f}} p(\mathbf{x}) > 0 \Big) ,$$

of which a sufficient condition is

$$\forall \mathbf{x}. \Big(h(\mathbf{x}) > 0 \longrightarrow L^1_{\mathbf{f}} p(\mathbf{x}) > 0 \Big) ,$$

and again of which a sufficient condition given by SOS relaxation is

$$L^1_{\mathbf{f}} p = s_1 + s_2 \cdot h + \varepsilon \ , \tag{2}$$

where s_1, s_2 are SOS and ε is a positive constant. By expressing s_1, s_2 via unknown semi-definite matrices and equating the parametric coefficients of monomials on both sides of (2), we can obtain an SDP problem, the solution of which gives an invariant $p \geq 0$.

As shown above, in general, a constraint that possesses easy SOS relaxation encoding has the form $\forall \mathbf{x}.(\bigwedge_{i=1}^{m} g_i \triangleright 0 \longrightarrow g_{m+1} \triangleright 0)$, where all g_i's are polynomials in $\mathbf{x}$ and $\triangleright \in \{\geq, >\}$. Handling arbitrary Boolean combinations, which is common case in Theorem 1 and 2, is not the strength of the SOS approach. For the particular purpose of facilitating SOS encodings of controller synthesis conditions, we propose specialized use of both Theorem 1 and 2, which can be found in the full version of this paper [18].

For the nuclear reactor example, we define two general *quartic* templates in the form of

$$\theta \geq 0 \wedge \theta \leq 1 \wedge \sum_{\alpha_1 + \alpha_2 \leq 4} c_{(\alpha_1, \alpha_2)} \cdot \theta^{\alpha_1} x^{\alpha_2} \leq 0$$

for mode q_2 and q_4. The idea is to fit the two boundaries of D'_2 (or D'_4), i.e. $\widehat{AE}$ and $\widehat{BD}$ in Fig. 3 simultaneously by one quartic polynomial, rather than two polynomials with lower degrees. The high efficiency of SOS solvers gives such possibility of using generic templates with higher degrees. Using the SOS relaxation techniques discussed above, the following switching controller is obtained:

- $G'_{12} \mathrel{\widehat{=}} \theta = 0 \wedge 519.10 \leq x \leq 547.86$;
- $G'_{23} \mathrel{\widehat{=}} \theta = 1 \wedge 512.09 \leq x \leq 550.00$;
- $G'_{34} \mathrel{\widehat{=}} \theta = 1 \wedge 512.09 \leq x \leq 546.15$;
- $G'_{41} \mathrel{\widehat{=}} \theta = 0 \wedge 510.00 \leq x \leq 547.86$.

For more details on the issues of defining templates, encoding constraints, applying numerical solvers etc, please refer to the full version [18].

4.2 The Template Polyhedra Approach

Convex polyhedra are a popular class of (positive) invariant sets of linear (continuous or discrete) systems [5]. A *convex polyhedron* in $\mathbb{R}^n$ can be represented using linear inequality constraints as $Q\mathbf{x} \leq \rho$, where $Q \in \mathbb{R}^{r\times n}$ is an $r \times n$ matrix, and $\mathbf{x} \in \mathbb{R}^{n\times 1}, \rho \in \mathbb{R}^{r\times 1}$ are column vectors.

Given a linear continuous dynamical system $\dot{\mathbf{x}} = A\mathbf{x}$ with $A \in \mathbb{R}^{n\times n}$, the following result on (positive) polyhedral invariant set is established in [7].

Proposition 1. *The polyhedron $Q\mathbf{x} \leq \rho$ is a positive invariant set of $\dot{\mathbf{x}} = A\mathbf{x}$ if and only if there exists an essentially non-negative*[7] *matrix $H \in \mathbb{R}^{r\times r}$ satisfying $HQ = QA$ and $H\rho \leq 0$.*

By simply applying the famous *Farkas' lemma* [11], we can generalize Proposition 1 and give a sufficient condition for polyhedral invariants of linear dynamics with *open* polyhedral domain (below we use a simple domain for ease of presentation).

Proposition 2. *Let $\mathbf{f} \mathrel{\widehat{=}} A\mathbf{x} + \mathbf{b}$ and $D \mathrel{\widehat{=}} \mathbf{c}\mathbf{x} < a$, where $a \in \mathbb{R}$, $\mathbf{b} \in \mathbb{R}^{n\times 1}$ is a column vector, and $\mathbf{c} \in \mathbb{R}^{1\times n}$ is a row vector. Then the polyhedron $Q\mathbf{x} \leq \rho$ is an invariant of the system $(D, \mathbf{f})$, if there exists an essentially non-negative matrix $H \in \mathbb{R}^{r\times r}$ and a non-negative column vector $\lambda \geq 0$ in $\mathbb{R}^{r\times 1}$ s.t.*

(1) $HQ = \mathsf{diag}(\lambda)QA - \mathsf{ones}^{(r,1)}\mathbf{c}$ *; and*
(2) $H\rho \leq -\mathsf{diag}(\lambda)Q\mathbf{b} - \mathsf{ones}^{(r,1)}a$ *,*

where $\mathsf{diag}(\lambda)$ *denotes the $r \times r$ diagonal matrix whose main diagonal corresponds to the vector λ, and* $\mathsf{ones}^{(r,1)}$ *denotes the $r \times 1$ column vector with all entries* 1*.*

Proof. Please refer to the full version of this paper [18]. □

Proposition 2 serves as the basis of automatic generation of polyhedral invariants for linear systems. To reduce the number of parameters in a polyhedral template, we propose the use of *template polyhedra*. The idea is to partly fix the shape of the invariant polyhedra by fixing the orientation of their facets. Formally, a template polyhedron is of the form $Q\mathbf{x} \leq \rho$ where Q is fixed a priori and ρ is to be determined. Any instantiation of ρ from $\mathbb{R}^{r\times 1}$ produces a concrete polyhedron. In this paper, since the system is planar, we choose Q in such a way that its row vectors form a set of uniformly distributed directions on a unit circle, i.e.

$$\mathbf{q}_i = \Big(\cos(\frac{i-1}{r}2\pi), \sin(\frac{i-1}{r}2\pi)\Big)$$

for $1 \leq i \leq r$, where $\mathbf{q}_i$ denotes the i-th row of Q. It is easy to see that $Q\mathbf{x} \leq \rho$ is just a rectangle when $r = 4$, and an octagon when $r = 8$.

To determine ρ, we need to find such matrices H and λ satisfying Proposition 2. Note that since both H and ρ are indeterminate, the constraint (2) in

[7] A square matrix is *essentially non-negative* if all its entries are non-negative except for those on the diagonal. Besides, given a matrix M, in this paper the inequalities or equations $M \geq 0$, $M > 0$, $M = 0$ should be interpreted entry-wise.

Proposition 2 becomes *bilinear*, making the problem NP-hard [46] to solve. It is however still tractable using modern BMI solvers.

Using octagonal templates[8] for mode q_2 and q_4 in the nuclear reactor example, we obtain the following switching controller:

- $G'_{12} \mathrel{\widehat{=}} \theta = 0 \wedge 519.90 \leq x \leq 545.70$;
- $G'_{23} \mathrel{\widehat{=}} \theta = 1 \wedge 514.40 \leq x \leq 550.00$;
- $G'_{34} \mathrel{\widehat{=}} \theta = 1 \wedge 514.40 \leq x \leq 538.24$;
- $G'_{41} \mathrel{\widehat{=}} \theta = 0 \wedge 510.00 \leq x \leq 545.70$.

As in Section 4.1, the readers are referred to [18] for more details about the application of the template polyhedra approach.

5 Conclusion and Discussion

We have extended a template-based approach for synthesizing switching controllers for semi-algebraic hybrid systems by combining symbolic invariant generation methods using quantifier elimination with qualitative methods to determine the likely shape of invariants. We have also investigated the application of numerical methods to gain more scalability and automation. A summary comparison of the three proposed approaches, as well as their advantages and problems, are given in the following aspects based on our experience.

- **Applicability.** The QE-based and SOS relaxation approaches can be applied to semi-algebraic systems which do not have closed-form solutions, while the template-polyhedra approach is only applicable to linear systems.
- **Design of Templates.** The QE-based approach demands much heuristics, which currently works well only on systems with low dimension, in determining templates, while the other two require less human effort.
- **Derivation of Synthesis Conditions.** For the QE-based method, derivation of synthesis conditions is a routine work because of the power of first-order formulas in formulating problems; the other two approaches are not good at handling complex logical structures and require the problems to be of specific forms, which usually demands some human work in practice.
- **Quality of Results.** The QE-based and SOS relaxation approaches can generate (non-convex) semi-algebraic invariants, while the template-polyhedra approach can only generate convex polyhedral invariants. Figure 4 demonstrates the synthesized D'_2 for the nuclear reactor example by the three approaches in turn: the first one is formed by straight lines and a parabola, the second one by straight lines and a quartic polynomial curve, and the third one is an octagon. For the switching controller synthesis problem, it's desirable to generate as large as possible invariants to gain more possibility of further refinement of the controllers based on other criteria. In this sense it's difficult to judge the merits of three approaches (e.g. in Fig. 4 the synthesized invariants have similar sizes).

[8] We search for polyhedral invariants using templates with $4, 8, 12, \ldots$ facets, and could not get a solution using templates with 4 facets.

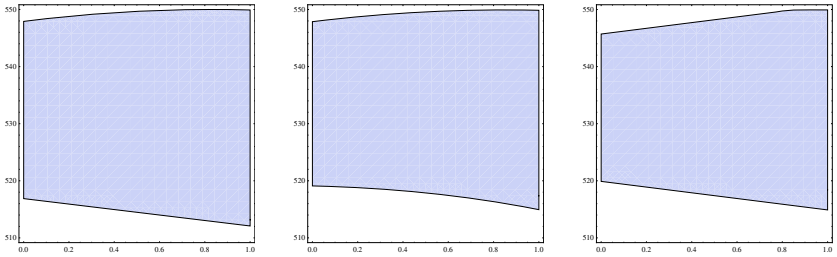

Fig. 4. Shapes of synthesized invariants by three approaches

– **Computational Issues.** For the QE-based approach, we have used the algebraic tool Redlog [10] to perform QE, and run several rounds of QEPCAD [6] (the slfq function) with human interactions to simplify the output of Redlog. For the numerical approaches, we use the MATLAB optimization toolbox YALMIP [24,25] as a high-level modeling environment and the interfaced external solvers SeDuMi [39] and PENBMI [19] (the TOMLAB [13] version) to solve the underlying SDP and BMI problems respectively. In our experiments, the SDP solver exhibits consistently good performances, but the BMI solver frequently runs into numerical problems. To make the BMI solver work we have to adjust the input constraints and the solver options a lot with trials and errors[9]. Table 1 shows the time cost of three approaches

Table 1. Templates and time cost of three controller synthesis approaches

Approach		QE-based	SOS-relaxation	template-polyhedra
Tool		Redlog + slfq	YALMIP + SeDuMi	YALMIP + PENBMI
Template	NR	quadratic, #PARMS = 4	generic quartic	8 facets
	TS	quadratic, #PARMS = 2	generic quartic	12 facets
Time (sec)	NR	12.663	1.969	0.578
	TS	7.092	1.609	1.703

on the nuclear reactor (NR) example as well as a thermostat (TS) example from [14]. All computations are done on a desktop with the Intel Q9400 2.66GHz CPU and 4GB RAM running Ubuntu Linux. We can see that for these two examples the QE-based approach is more expensive in time compared to numerical approaches. However, according to our experience, the template-polyhedra approach does not scale well due to the NP-hardness of BMI problems, so its superiority to QE-based approach may not always be the case. For a detailed explanation of Table 1 as well as the description of the TS example please refer to [18].

[9] Another way is to use the global nonlinear optimization solver BMIBNB in YALMIP, which would cause an increase of time cost by dozens (even hundreds) of times for the same two examples.

– **Soundness.** The QE-based approach is exact while the other two approaches suffer from numerical errors which would cause the synthesis of unsafe controllers. To partly address this problem, we have directly encoded some tolerance of numerical errors into the synthesis conditions to increase robustness and reduce the risk of synthesizing bad controllers. The justification for adopting numerical methods is that verification is much easier than synthesis. For example, for the nuclear reactor example, we have verified posteriorly and symbolically the controllers synthesized by both numerical approaches, and the verification process is very efficient.

Our preliminary analysis suggests the effectiveness of the three proposed approaches. We are currently experimenting with these methods on more examples, especially nonlinear ones which do not possess closed-form solutions. We plan to extend these approaches for reachability and/or optimality requirements as well, by incorporating our previous results on *asymptotic stability* analysis [23] and a case study in optimal control [49].

Acknowledgements. We thank Dr. Jiang Liu for his great contribution to our previous joint work on invariant generation. We also thank Dr. Matthias Horbach, Mr. ThanhVu Nguyen, and the anonymous reviewers for their valuable comments, which help to improve the quality of our paper greatly.

References

1. Alur, R.: Formal verification of hybrid systems. In: EMSOFT 2011, pp. 273–278. ACM (2011)
2. Alur, R., Couroubetis, C., Henzinger, T., Ho, P.H.: Hybrid automata: an algorithmic approach to the specification and verification of hybrid systems. In: Grossman, R.L., Ravn, A.P., Rischel, H., Nerode, A. (eds.) HS 1991 and HS 1992. LNCS, vol. 736, pp. 209–229. Springer, Heidelberg (1993)
3. Alur, R., Courcoubetis, C., Halbwachs, N., Henzinger, T.A., Ho, P.H., Nicollin, X., Olivero, A., Sifakis, J., Yovine, S.: The algorithmic analysis of hybrid systems. Theor. Comput. Sci. 138(1), 3–34 (1995)
4. Asarin, E., Bournez, O., Dang, T., Maler, O., Pnueli, A.: Effective synthesis of switching controllers for linear systems. Proc. of the IEEE 88(7), 1011–1025 (2000)
5. Blanchini, F.: Set invariance in control. Automatica 35(11), 1747–1767 (1999)
6. Brown, C.W.: QEPCAD B: A program for computing with semi-algebraic sets using CADs. SIGSAM Bulletin 37, 97–108 (2003)
7. Castelan, E., Hennet, J.: On invariant polyhedra of continuous-time linear systems. IEEE Trans. Autom. Control 38(11), 1680–1685 (1993)
8. Cousot, P.: Proving program invariance and termination by parametric abstraction, Lagrangian relaxation and semidefinite programming. In: Cousot, R. (ed.) VMCAI 2005. LNCS, vol. 3385, pp. 1–24. Springer, Heidelberg (2005)
9. Davenport, J.H., Heintz, J.: Real quantifier elimination is doubly exponential. J. Symb. Comput. 5(1-2), 29–35 (1988)
10. Dolzmann, A., Seidl, A., Sturm, T.: Redlog User Manual (November 2006), `http://redlog.dolzmann.de/downloads/`, edition 3.1, for redlog Version 3.06 (reduce 3.8)

11. Gulwani, S., Tiwari, A.: Constraint-based approach for analysis of hybrid systems. In: Gupta, A., Malik, S. (eds.) CAV 2008. LNCS, vol. 5123, pp. 190–203. Springer, Heidelberg (2008)
12. Ho, P.H.: The algorithmic analysis of hybrid systems. Ph.D. thesis, Cornell University (1995)
13. Holmström, K., Göran, A.O., Edvall, M.M.: User's Guide for TOMLAB/PENOPT. Tomlab Optimization (November 2006), http://tomopt.com/docs/TOMLAB_PENOPT.pdf
14. Jha, S., Gulwani, S., Seshia, S.A., Tiwari, A.: Synthesizing switching logic for safety and dwell-time requirements. In: ICCPS 2010, pp. 22–31. ACM (2010)
15. Kapur, D.: A quantifier-elimination based heuristic for automatically generating inductive assertions for programs. Journal of Systems Science and Complexity 19(3), 307–330 (2006)
16. Kapur, D.: Automatically Generating Loop Invariants Using Quantifier Elimination. Technical Report, Department of Computer Science, University of New Mexico, Albuquerque, USA (December 2003)
17. Kapur, D., Shyamasundar, R.K.: Synthesizing controllers for hybrid systems. In: Maler, O. (ed.) HART 1997. LNCS, vol. 1201, pp. 361–375. Springer, Heidelberg (1997)
18. Kapur, D., Zhan, N., Zhao, H.: Synthesizing switching controllers for hybrid systems by continuous invariant generation. CoRR abs/1304.0825 (2013), http://arxiv.org/abs/1304.0825
19. Kočvara, M., Stingl, M.: PENBMI User's Guide (Version 2.1). PENOPT GbR (March 2006), http://www.penopt.com/doc/penbmi2_1.pdf
20. Lin, W., Wu, M., Yang, Z., Zeng, Z.: Exact safety verification of hybrid systems using sums-of-squares representation. CoRR abs/1112.2328 (2011), http://arxiv.org/abs/1112.2328
21. Liu, J., Lv, J., Quan, Z., Zhan, N., Zhao, H., Zhou, C., Zou, L.: A calculus for hybrid CSP. In: Ueda, K. (ed.) APLAS 2010. LNCS, vol. 6461, pp. 1–15. Springer, Heidelberg (2010)
22. Liu, J., Zhan, N., Zhao, H.: Computing semi-algebraic invariants for polynomial dynamical systems. In: EMSOFT 2011, pp. 97–106. ACM (2011)
23. Liu, J., Zhan, N., Zhao, H.: Automatically discovering relaxed Lyapunov functions for polynomial dynamical systems. Mathematics in Computer Science 6(4), 395–408 (2012)
24. Löfberg, J.: YALMIP: A toolbox for modeling and optimization in MATLAB. In: Proc. of the CACSD Conference, Taipei, Taiwan (2004), http://users.isy.liu.se/johanl/yalmip
25. Löfberg, J.: Pre- and post-processing sum-of-squares programs in practice. IEEE Trans. Autom. Control 54(5), 1007–1011 (2009)
26. Parrilo, P.A.: Structured Semidefinite Programs and Semialgebraic Geometry Methods in Robustness and Optimization. Ph.D. thesis, California Institute of Technology, Pasadena, CA (May 2000), http://thesis.library.caltech.edu/1647/
27. Platzer, A.: Differential-algebraic dynamic logic for differential-algebraic programs. J. Log. and Comput. 20(1), 309–352 (2010)
28. Platzer, A., Clarke, E.M.: Computing differential invariants of hybrid systems as fixedpoints. In: Gupta, A., Malik, S. (eds.) CAV 2008. LNCS, vol. 5123, pp. 176–189. Springer, Heidelberg (2008)
29. Platzer, A.: A differential operator approach to equational differential invariants. In: Beringer, L., Felty, A. (eds.) ITP 2012. LNCS, vol. 7406, pp. 28–48. Springer, Heidelberg (2012)

30. Platzer, A.: The structure of differential invariants and differential cut elimination. Logical Methods in Computer Science 8(4), 1–38 (2012)
31. Prajna, S., Jadbabaie, A.: Safety verification of hybrid systems using barrier certificates. In: Alur, R., Pappas, G.J. (eds.) HSCC 2004. LNCS, vol. 2993, pp. 477–492. Springer, Heidelberg (2004)
32. Prajna, S., Jadbabaie, A., Pappas, G.J.: A framework for worst-case and stochastic safety verification using barrier certificates. IEEE Trans. Autom. Control 52(8), 1415–1428 (2007)
33. Prajna, S., Papachristodoulou, A., Seiler, P., Parrilo, P.: SOSTOOLS and its control applications. In: Henrion, D., Garulli, A. (eds.) Positive Polynomials in Control. LNCIS, vol. 312, pp. 273–292. Springer, Heidelberg (2005)
34. Sankaranarayanan, S., Sipma, H., Manna, Z.: Non-linear loop invariant generation using Gröbner bases. In: POPL 2004 (2004)
35. Sankaranarayanan, S., Dang, T., Ivančić, F.: A policy iteration technique for time elapse over template polyhedra. In: Egerstedt, M., Mishra, B. (eds.) HSCC 2008. LNCS, vol. 4981, pp. 654–657. Springer, Heidelberg (2008)
36. Sankaranarayanan, S., Dang, T., Ivančić, F.: Symbolic model checking of hybrid systems using template polyhedra. In: Ramakrishnan, C.R., Rehof, J. (eds.) TACAS 2008. LNCS, vol. 4963, pp. 188–202. Springer, Heidelberg (2008)
37. Sankaranarayanan, S., Sipma, H.B., Manna, Z.: Scalable analysis of linear systems using mathematical programming. In: Cousot, R. (ed.) VMCAI 2005. LNCS, vol. 3385, pp. 25–41. Springer, Heidelberg (2005)
38. Sassi, M.A.B., Girard, A.: Computation of polytopic invariants for polynomial dynamical systems using linear programming. Automatica 48(12), 3114–3121 (2012)
39. Sturm, J.F.: Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. Optimization Methods and Software 11-12, 625–653 (1999)
40. Sturm, T., Tiwari, A.: Verification and synthesis using real quantifier elimination. In: ISSAC 2011, pp. 329–336. ACM (2011)
41. Taly, A., Gulwani, S., Tiwari, A.: Synthesizing switching logic using constraint solving. International Journal on Software Tools for Technology Transfer 13, 519–535 (2011)
42. Taly, A., Tiwari, A.: Deductive verification of continuous dynamical systems. In: FSTTCS 2009. LIPIcs, vol. 4, pp. 383–394 (2009)
43. Taly, A., Tiwari, A.: Switching logic synthesis for reachability. In: EMSOFT 2010, pp. 19–28. ACM (2010)
44. Tarski, A.: A Decision Method for Elementary Algebra and Geometry. University of California Press, Berkeley (1951)
45. Tomlin, C.J., Lygeros, J., Sastry, S.S.: A game theoretic approach to controller design for hybrid systems. Proc. of the IEEE 88(7), 949–970 (2000)
46. VanAntwerp, J.G., Braatz, R.D.: A tutorial on linear and bilinear matrix inequalities. Journal of Process Control 10(4), 363–385 (2000)
47. Vandenberghe, L., Boyd, S.: Semidefinite programming. SIAM Review 38(1), 49–95 (1996)
48. Yang, Z., Wu, M., Lin, W.: Exact safety verification of hybrid systems based on bilinear SOS representation. CoRR abs/1201.4219 (2012), `http://arxiv.org/abs/1201.4219`
49. Zhao, H., Zhan, N., Kapur, D., Larsen, K.G.: A "hybrid" approach for synthesizing optimal controllers of hybrid systems: A case study of the oil pump industrial example. In: Giannakopoulou, D., Méry, D. (eds.) FM 2012. LNCS, vol. 7436, pp. 471–485. Springer, Heidelberg (2012)