A "Hybrid" Approach for Synthesizing Optimal Controllers of Hybrid Systems: A Case Study of the Oil Pump Industrial Example

Hengjun Zhao[1,2], Naijun Zhan[2], Deepak Kapur[3], and Kim G. Larsen[4 ⋆]

[1] State Key Lab. of Comput. Sci., Institute of Software, CAS, Beijing, China
[2] Graduate University of Chinese Academy of Sciences, Beijing, China
[3] Dept. of Comput. Sci., University of New Mexico, Albuquerque, NM, USA
[4] CISS, CS, Aalborg University, Denmark

Abstract. We propose an approach to reduce the optimal controller synthesis problem of hybrid systems to quantifier elimination; furthermore, we also show how to combine quantifier elimination with numerical computation in order to make it more scalable but at the same time, keep arising errors due to discretization manageable and within bounds. A major advantage of our approach is not only that it avoids errors due to numerical computation, but it also gives a better optimal controller. In order to illustrate our approach, we use the real industrial example of an oil pump provided by the German company HYDAC within the European project *Quasimodo* as a case study throughout this paper, and show that our method improves (up to 7.5%) the results reported in [4] based on game theory and model checking.

Keywords: Hybrid System, Optimal Control, Quantifier Elimination, Numerical Computation

1 Introduction

Hybrid systems such as physical devices controlled by computer software, are systems that exhibit both continuous and discrete behaviors. Controller synthesis for hybrid systems is an important area of research in both academia and industry. A synthesis problem focuses on designing a controller that ensures the given system will satisfy a safety requirement, a liveness requirement (e.g. reachability to a given set of states), or meet an optimality criterion, or a desired combination of these requirements.

Numerous work have been done on controller synthesis for safety and/or reachability requirements. For example, in [1, 28], a general framework for synthesizing controllers based on hybrid automata to meet a given safety requirement was proposed, which relies on *backward reachable set* computation and

⋆ The first and second authors are supported by NSFC projects 91118007 and 60970031; the third author is supported by NSF CCF-0729097 and CNS-0905222; the fourth author is supported by The Danish VKR Center of Excellence MT-LAB and The Sino-Danish Basic Research Center IDEA4CPS.

fixed point iteration; while in [25], a symbolic approach based on templates and constraint solving to the same problem was proposed, and in [26], the symbolic approach is extended to meet both safety and reachability requirements.

However, the optimal controller synthesis problem is more involved, also quite important in the design of hybrid systems. In the literature, few work has been done on the problem. Larsen et al proposed an approach based on energy automata and model-checking [4], while Jha, Seshia and Tiwari gave a solution to the problem using unconstrained numerical optimization and machine learning [15]. However, in [4], allowing control only to be exercised at discrete time points certainly limits the opportunity of synthesizing the optimal controller (though one can get arbitrarily close). Moreover, discretizing could cause an incorrect controller to be synthesized — which therefore requires a posterior analysis (e.g. in [4], PHAVER [10] is used for the purpose). The approach of [15] suffers from imprecision caused by numerical computation, and cannot synthesize a really optimal controller sometimes because the machine learning technique cannot guarantee its completeness.

In this paper, we propose a "hybrid" approach for synthesizing optimal controllers of hybrid systems subject to safety requirements. The basic idea is as follows. Firstly, we reduce optimal controller synthesis subject to safety requirements to quantifier elimination (QE for short). Secondly, in order to make our approach scalable, we discuss how to combine QE with numerical computation, but at the same time, keep arising errors due to discretization manageable and within bounds. A major advantage of our approach is not only that it avoids errors due to numerical computation, but also it gives a better optimal controller.

Application of QE in controller synthesis of hybrid systems is not new. The tool HyTech was the first symbolic model checker that can do parametric analysis [13] for linear hybrid automata, but for the oil pump example it will abort soon due to arithmetic overflow. Recently, verification and synthesis of switched dynamical systems using QE were discussed in [24], where the authors gave principles and heuristics for combining different tools, to solve QE problems that are out of the scope of each component tool.

Our encoding of a MIN-MAX-MIN optimization problem into a QE problem is inspired by the idea in [8]: minimizing a polynomial objective function $f(x_1, x_2, \ldots, x_n)$ can be done by introducing an additional constraint $z \geq f(x_1, x_2, \ldots, x_n)$ and then eliminating variables $x_1, x_2, \ldots, x_n$, where z is a newly introduced variable. Similar ideas can also be found in [5].

The computation of optimal control strategies in this paper is typically a *parametric optimization* problem, a topic researched extensively in both operation research and control communities. Symbolic methods have advantages in addressing parametric optimization problems [29, 9, 16]. However, we do not find any algorithm suitable for solving a parametric quadratic optimization problem over constraint with complex Boolean structure and hundreds of (or thousands of) atomic formulas as in this paper.

It was shown in [2] that for certain parametric quadratic optimization problems, the closed form solution exists: the optimizer is a piecewise affine function in the parameters, and the optimal value is a piecewise quadratic function in the parameters. Our experiment results confirm this.

In order to illustrate our approach, we use the oil pump industrial example provided by the German company HYDAC within the European project *Quasimodo* as a case study throughout this paper, and show that our method results in a better optimal controller (up to 7.5% improvement) than those reported in [4] based on game theory and model checking. Moreover, we prove that the theoretically optimal controller of the oil pump example can be synthesized and its correctness is also guaranteed with our approach.

Paper Organization: In Section 2 we propose a general framework for optimal controller synthesis of hybrid systems based on QE and numerical computation. We focus on the oil pump case study in Section 3-5: a description of the oil pump control problem is given in Section 3, modeling of the system and safety requirements is shown in Section 4, a "hybrid" approach for performing optimization is presented in Section 5, in which further improvement by increasing activation times of the pump is also discussed. We conclude this paper by Section 6.

2 The Overall Approach

In this section we propose an approach that reduces optimal controller synthesis of hybrid systems subject to safety requirements to QE. Reachable sets of hybrid systems are modeled exactly or approximated using polynomial formulas. Optimality criteria and safety requirements are also modeled in the same way. Existentially quantified formulas can be reduced to a finite set of disjunctions by discretizing the existentially quantified variables over bounded intervals, which often leads to scalability.

Generally, a hybrid system consists of a set of discrete operating modes Q, with each of which a continuous dynamics is associated, specifying the behavior of a set of continuous states $\mathbf{x}$. Discrete jumps between different modes may happen if some *transition conditions* are satisfied by $\mathbf{x}$.

The optimal controller synthesis problem studied in this paper can be stated as follows. Suppose we are given an under-specified hybrid system $\mathcal{H}$, in which the transition conditions are not determined but parameterized by $\mathbf{u}$, a vector of control parameters. Our task is to determine values of $\mathbf{u}$ such that $\mathcal{H}$ can make discrete jumps at desired points, thus guaranteeing that

1) a safety requirement $\mathcal{S}$ is satisfied, that is, $\mathbf{x}$ stays in a designated safe region at any time point; and
2) an optimization goal $\mathcal{G}$, possibly

$$\min_{\mathbf{u}} g(\mathbf{u}),\ \max_{\mathbf{u}_2}\min_{\mathbf{u}_1} g(\mathbf{u})\,,\text{ or }\ \min_{\mathbf{u}_3}\max_{\mathbf{u}_2}\min_{\mathbf{u}_1} g(\mathbf{u})\,,^{1}$$

where $g(\mathbf{u})$ is an objective function in parameters $\mathbf{u}$, is achieved.

[1] We assume that $\mathbf{u}$ is chosen from a compact (i.e. bounded closed) set, and the elements of $\mathbf{u}$ are divided into groups $\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3, \ldots$ according to their roles in $\mathcal{G}$.

Our approach for solving the synthesis problem can be described as the following steps.

Step 1. *Derive constraint $D(\mathbf{u})$ on $\mathbf{u}$ from the safety requirements of the system.*

If the reachable set R (parameterized by $\mathbf{u}$) of $\mathcal{H}$ can be exactly computed (e.g. for very simple linear hybrid automata), then we just require that R should be contained in the safe region. Otherwise we have to approximate R (with sufficient precision) by automatically generating inductive invariants of $\mathcal{H}$ (e.g. for general linear or nonlinear hybrid systems). The notion of *inductive invariant* is crucial in safety verification of hybrid systems [11, 22], and constraint-based approaches have been proposed for automatic generation of inductive invariants [23, 11, 21, 17].

Step 2. *Encode the optimization problem $\mathcal{G}$ over constraint $D(\mathbf{u})$ into a quantified first-order formula $\mathbf{Qu}.\varphi(\mathbf{u}, z)$, where z is a fresh variable.*

Our encoding is based on the following proposition, in which we discuss all the aforementioned optimization functions together.

Proposition 1. *Suppose $g_1(\mathbf{u}_1)$, $g_2(\mathbf{u}_1, \mathbf{u}_2)$, $g_3(\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3)$ are polynomials, and $D_1(\mathbf{u}_1)$, $D_2(\mathbf{u}_1, \mathbf{u}_2)$, $D_3(\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3)$ are nonempty compact semi-algebraic sets*[2]*. Then there exist c_1, c_2, $c_3 \in \mathbb{R}$ s.t.*

$$\exists \mathbf{u}_1.(D_1 \wedge g_1 \leq z) \iff z \geq c_1\,, \quad (1)$$

$$\forall \mathbf{u}_2.\big(\exists \mathbf{u}_1.D_2 \longrightarrow \exists \mathbf{u}_1.(D_2 \wedge g_2 \leq z)\big) \iff z \geq c_2\,, \quad (2)$$

$$\exists \mathbf{u}_3.\big((\exists \mathbf{u}_1 \mathbf{u}_2.D_3) \wedge \forall \mathbf{u}_2.\big(\exists \mathbf{u}_1.D_3 \longrightarrow \exists \mathbf{u}_1.(D_3 \wedge g_3 \leq z)\big)\big) \iff z \rhd c_3\,, \quad (3)$$

where $\rhd \in \{>, \geq\}$, and c_1, c_2, c_3 satisfy

$$c_1 = \min_{\mathbf{u}_1} g_1(\mathbf{u}_1) \quad \text{over } D_1(\mathbf{u}_1)\,, \quad (4)$$

$$c_2 = \sup_{\mathbf{u}_2} \min_{\mathbf{u}_1} g_2(\mathbf{u}_1, \mathbf{u}_2) \quad \text{over } D_2(\mathbf{u}_1, \mathbf{u}_2)\,, \quad (5)$$

$$c_3 = \inf_{\mathbf{u}_3} \sup_{\mathbf{u}_2} \min_{\mathbf{u}_1} g_3(\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3) \quad \text{over } D_3(\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3)\,. \quad (6)$$

We omit the proof of this proposition due to space limitation. All the proofs, as well as the formulas generated by QE, can be found in the full version [30] of the paper.

Step 3. *Eliminate quantifiers in $\mathbf{Qu}.\varphi(\mathbf{u}, z)$ and from the result we can retrieve the optimal value of $\mathcal{G}$ and the corresponding optimal controller $\mathbf{u}$.*

By Proposition 1, the optimal value of a MIN, MAX-MIN or MIN-MAX-MIN[3] problem can be obtained by applying QE to the left hand side (LHS) formulas in

[2] A semi-algebraic set is defined by Boolean combinations of polynomial equations and inequalities.

[3] By Proposition 1, the MIN (MAX) notation can really be INF (SUP) sometimes.

(1)-(3) respectively. Although QE for the first-order theory of real closed fields is a complete decision procedure [27], due to the inherent doubly exponential complexity [6], direct QE would fail on big formulas with many alternations of quantifiers, as in LHS of (3). It is then necessary to devise heuristics to do QE more efficiently for such special formulas.

Note that in (3), any instantiation of the outmost quantified variables $\mathbf{u}_3$ would result in a simpler formula, whose quantifier-free equivalence gives an upper bound of c_3. If in some way we know the bounds of $\mathbf{u}_3$, i.e. $l_i \le \mathbf{u}_3^i \le u_i$, for $1 \le i \le \dim(\mathbf{u}_3)$, then by discretizing $\mathbf{u}_3$ over all $[l_i, u_i]$ with certain granularity Δ, and using the set of discretized values to instantiate the outmost existential quantifiers of (3), we can get a finite set of simplified formulas, each of which produces an upper approximation of c_3. Finally, through an exhaustive search in this set we can select such an approximation that is closest to c_3. Finer granularity yields better approximation of the optimal value, so one can seek for a good balance between timing and optimality by tuning the granularity Δ. Furthermore, the above computation is well suited for parallelization to make full use of available computing resources, because the intervals $[l_i, u_i]$ and corresponding instantiations can be divided into subgroups and allocated to different processes.

3 Description of the Oil Pump Control Problem

The oil pump example [4] was a real industrial case provided by the German company HYDAC ELECTRONICS GMBH, and studied at length within the European research project *Quasimodo*. The whole system, depicted by Fig. 1, consists of a machine, an accumulator, a reservoir and a pump. The machine consumes oil periodically out of the accumulator with a period of $20s$ (second) for one consumption cycle. The profile of consumption rate is shown in Fig. 2. The pump adds oil from the reservoir into the accumulator with power $2.2l/s$ (liter/second).

Control objectives for this system are: by switching on/off the pump at certain time points

$$0 \le t_1 \le t_2 \le \cdots \le t_n \le t_{n+1} \le \cdots, \tag{7}$$

ensuring that

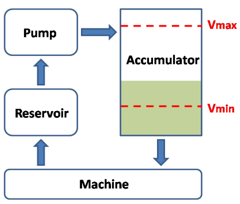

Fig. 1. The oil pump system. (This picture is based on [4].)

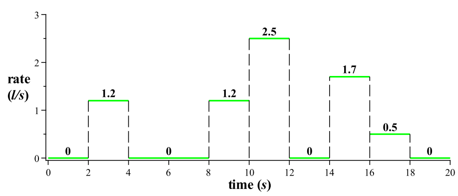

Fig. 2. Consumption rate of the machine in one cycle.

- $\mathrm{R_s}$ (*safety*): the system can run arbitrarily long while maintaining $v(t)$ within $[V_{\min}, V_{\max}]$ for any time point t, where $v(t)$ denotes the oil volume in the accumulator at time t, $V_{\min} = 4.9l$ (liter) and $V_{\max} = 25.1l$;

and considering the energy cost and wear of the system, a second objective:

- $\mathrm{R_o}$ (*optimality*): minimizing the average accumulated oil volume in the limit, i.e. minimizing
$$\lim_{T\to\infty} \frac{1}{T} \int_{t=0}^{T} v(t)\mathrm{d}t \ .$$

Both objectives should be achieved under two additional constraints:

- $\mathrm{R_{pl}}$ (*pump latency*): there must be a latency of at least $2s$ between any two consecutive operations of the pump; and
- $\mathrm{R_r}$ (*robustness*): uncertainty of the system should be taken into account:
  - fluctuation of consumption rate (if it is not 0), up to $f = 0.1l/s$;
  - imprecision in the measurement of oil volume, up to $\epsilon = 0.06l$;
  - imprecision in the measurement of time, up to $\delta = 0.015s$.[4]

In [4], the authors used timed game automata to model the above system, and applied the tool UPPAAL-TIGA to synthesize near-optimal controllers. Due to discretization made in the timed-game model, an incorrect controller might be synthesized. Therefore the correctness and robustness of the synthesized controllers are checked using the tool PHAVER. Through simulations with SIMULINK, it was shown that the controller synthesized by UPPAAL-TIGA provides big improvement (about 40%) over the *Bang-Bang Controller* and *Smart Controller* that are currently used at the HYDAC company. We will show how further improvement can be achieved using our approach.

4 Deriving Constraints from Safety Requirements

Following [4], the determination of control points (7) can be localized by exploiting the periodicity of oil consumption. That is, decisions on when to switch on/off the pump in one cycle can be made *locally* by measuring the initial oil volume v_0 at the beginning of each cycle. Accordingly, the safety requirement $\mathrm{R_s}$ in Section 3 can be reformulated as: find an interval $[L, U] \subseteq [V_{\min}, V_{\max}]$ s.t.

- $\mathrm{R_{lu}}$ (*constraint for* L, U): for all $v_0 \in [L, U]$, there is a finite sequence of time points $\mathbf{t} = (t_1, t_2, \ldots, t_n)$,[5] where $0 \le t_1 \le t_2 \le \ldots \le t_n \le 20$ satisfy $\mathrm{R_{pl}}$, for turning on/off the pump so that the resulting $v(t)$ with $v(0) = v_0$ satisfies
  - $\mathrm{R_i}$ (*inductiveness*): $v(20) \in [L, U]$; and

[4] In [4], δ is assumed to be 0.01. Here we include an extra rounding error of 0.005 due to floating point calculations in the implementation of our control strategy.

[5] The choice of n will be made later (in this paper n can be $0, 2, 4, 6$), but larger n's obviously will have the potential of allowing improved controllers.

- R_{ls} (*local safety*): $v(t) \in [V_{\min}, V_{\max}]$ for all $t \in [0, 20]$

under the constraint R_{r}.

Definition 1 (Local Controller). *The above* $\mathbf{t}$ *corresponding to* v_0 *is called a local controller; the interval* $[L, U]$ *is called a stable interval.*

Basically, R_{lu} says that there is a stable interval $[L, U]$ and a corresponding family of local control strategies which can be repeated for arbitrarily many cycles and guarantee safety in each cycle.

Modeling Oil Consumption. Let $V_{out}(t)$ with $V_{out}(0) = 0$ denote the amount of oil consumed by time t in one cycle, and modify the consumption rate in Fig. 2 by f in (R_{r}). Then by simply integrating the lower and upper bounds of the consumption rate over the time interval $[0, 20]$ we can get

$$C_1 \mathrel{\widehat{=}} \begin{array}{ll} (0 \le t \le 2 & \longrightarrow V_{out} = 0) \\ \wedge\, (2 \le t \le 4 & \longrightarrow 1.1(t-2) \le V_{out} \le 1.3(t-2)) \\ \wedge\, (4 \le t \le 8 & \longrightarrow 2.2 \le V_{out} \le 2.6) \\ \wedge\, (8 \le t \le 10 & \longrightarrow 2.2 + 1.1(t-8) \le V_{out} \le 2.6 + 1.3(t-8)) \\ \wedge\, (10 \le t \le 12 & \longrightarrow 4.4 + 2.4(t-10) \le V_{out} \le 5.2 + 2.6(t-10)) \\ \wedge\, (12 \le t \le 14 & \longrightarrow 9.2 \le V_{out} \le 10.4) \\ \wedge\, (14 \le t \le 16 & \longrightarrow 9.2 + 1.6(t-14) \le V_{out} \le 10.4 + 1.8(t-14)) \\ \wedge\, (16 \le t \le 18 & \longrightarrow 12.4 + 0.4(t-16) \le V_{out} \le 14 + 0.6(t-16)) \\ \wedge\, (18 \le t \le 20 & \longrightarrow 13.2 \le V_{out} \le 15.2) \end{array} .^{6}$$

Actually, if the machine consuming oil is regarded as a hybrid system $\mathcal{H}$ with state variable V_{out} and continuous dynamics subject to *box* constraints, then C_1 is the exact *reachable set* of $\mathcal{H}$ from initial point $V_{out} = 0$ within 20 time units. Therefore we do not need to approximate the reachable set of $\mathcal{H}$ by generating inductive invariants. This is also the case with the following pump system. However, if the consumption profile is more complicated, say piecewise polynomial, then approximations are indeed necessary.

Modeling Pump. In [4] it is assumed that the number of activations of pump in one cycle is at most 2. We will adopt this assumption at first and increase this number later on. With this assumption, there will be at most four time points to switch the pump on/off in one cycle, denoted by $0 \le t_1 \le t_2 \le t_3 \le t_4 \le 20$. If the pump is started only once or zero times, then we just set $t_3 = t_4 = 20$ or $t_1 = t_2 = t_3 = t_4 = 20$ respectively. Then the 2-second latency requirement (R_{pl}) can be modeled by

$$C_2 \mathrel{\widehat{=}} \begin{array}{l} (t_1 \ge 2 \wedge t_2 - t_1 \ge 2 \wedge t_3 - t_2 \ge 2 \wedge t_4 - t_3 \ge 2 \wedge t_4 \le 20) \\ \vee\, (t_1 \ge 2 \wedge t_2 - t_1 \ge 2 \wedge t_2 \le 20 \wedge t_3 = 20 \wedge t_4 = 20) \\ \vee\, (t_1 = 20 \wedge t_2 = 20 \wedge t_3 = 20 \wedge t_4 = 20) \end{array} .$$

[6] In the sequel when a function $\gamma(t)$ appears in a formula, the argument t is dropped and γ is taken as a real-valued variable.

Let $V_{in}(t)$ with $V_{in}(0) = 0$ denote the amount of oil introduced into the accumulator by time t in one cycle. Then we have

$$C_3 \widehat{=} \begin{array}{ll} (0 \le t \le t_1 & \longrightarrow V_{in}=0) \\ \wedge\,(t_1 \le t \le t_2 & \longrightarrow V_{in}=2.2(t-t_1)) \\ \wedge\,(t_2 \le t \le t_3 & \longrightarrow V_{in}=2.2(t_2-t_1)) \\ \wedge\,(t_3 \le t \le t_4 & \longrightarrow V_{in}=2.2(t_2-t_1)+2.2(t-t_3)) \\ \wedge\,(t_4 \le t \le 20 & \longrightarrow V_{in}=2.2(t_2+t_4-t_1-t_3)) \end{array} .$$

Encoding Safety Requirements. Denote the oil volume in the accumulator at the beginning of one cycle by v_0, and the volume at time t by $v(t)$. Then for any $0 \le t \le 20$ we have:

$$C_4 \widehat{=} v = v_0 + V_{in} - V_{out} \ .$$

According to (R_r), the measurement of t_i $(1 \le i \le 4)$ and v_0 may deviate from their actual values, so $v(t)$ will deviate from its predicted value as stated in the constraint C_4. Nevertheless, we have the following estimation of the deviation of $v(t)$.

Lemma 1. *Let $\tilde{v}(t)$ denote the actual oil volume in the accumulator at time t. Then for any $0 \le t \le 20$, $|v(t) - \tilde{v}(t)| \le 8.8\,\delta + \epsilon < 0.2$.*

Please refer to [30] for the proof of Lemma 1. By Lemma 1, it is sufficient to rectify the safety bounds in (R_i) and (R_{ls}) by an amount of 0.2. Let

$$\begin{array}{lll} C_5 & \widehat{=} & t = 20 \longrightarrow L + 0.2 \le v \le U - 0.2 \\ C_6 & \widehat{=} & 0 \le t \le 20 \longrightarrow V_{\min} + 0.2 \le v \le V_{\max} - 0.2 \ . \end{array}$$

Then (R_i) and (R_{ls}) can be expressed as

$$\mathcal{S} \widehat{=} \forall t, v, V_{in}, V_{out}.(C_1 \wedge C_3 \wedge C_4 \longrightarrow C_5 \wedge C_6)\,.$$

Deriving Constraints. To find $[L, U]$ such that for every $v_0 \in [L, U]$ there is a local control strategy satisfying R_i and R_{ls}, let

$$C_7 \widehat{=} L \le v_0 \le U \ ,$$

and then R_{lu} can be encoded into

$$C_8 \widehat{=} \forall v_0.\Big(C_7 \longrightarrow \exists t_1 t_2 t_3 t_4.\big(C_2 \ \wedge \mathcal{S}\big)\Big)\ .$$

We use the tool Mjollnir [19] to do QE on C_8 and the following result is returned:

$$C_9 \widehat{=} L \ge 5.1 \wedge U \le 24.9 \wedge U - L \ge 2.4 \ .$$

Then the relation between L, U, v_0 and the corresponding local control strategy $\mathbf{t} = (t_1, t_2, t_3, t_4)$ can be obtained by applying QE to

$$C_{10} \widehat{=} C_2 \wedge C_7 \wedge C_9 \ \wedge \mathcal{S}\,.$$

The result given by Mjollnir, when converted to DNF, is a disjunction of 92 components:

$$\mathcal{D}(L, U, v_0, t_1, t_2, t_3, t_4) \mathrel{\widehat{=}} \bigvee_{i=1}^{92} D_i$$

(denoted by $\mathcal{D}$ for short), with each D_i representing a nonempty closed convex polyhedron.[7]

5 A "Hybrid" Approach for Optimization

5.1 Encoding of the Optimization Objective

By Definition 1, the optimal average accumulated oil volume in $\mathrm{R_o}$ can be redefined as

$$\bullet\ \mathrm{R'_o}: \qquad \min_{[L,U]} \max_{v_0 \in [L,U]} \min_{\mathbf{t}} \frac{1}{20} \int_{t=0}^{20} v(t) \mathrm{d}t \ . \tag{8}$$

The intuitive meaning of $(\mathrm{R'_o})$ is:

– for each admissible $[L, U]$ satisfying C_9 and each $v_0 \in [L, U]$, minimize the average accumulated oil volume in one cycle, i.e. $\frac{1}{20} \int_{t=0}^{20} v(t) \mathrm{d}t$, over all admissible local controllers $\mathbf{t}$;
– fix $[L, U]$ and select the *worst* local minimum by traversing all $v_0 \in [L, U]$;
– then the global minimum is obtained at the interval whose worst local minimum is *minimal*.

Definition 2 (Local Optimal Controller). *Let* $\mathcal{D}_{\mathbf{t}} \mathrel{\widehat{=}} \{\mathbf{t} \mid (L, U, v_0, \mathbf{t}) \in \mathcal{D}\}$ *for fixed* L, U, v_0*. Then we call*

$$\min_{\mathbf{t} \in \mathcal{D}_{\mathbf{t}}} \frac{1}{20} \int_{t=0}^{20} v(t) \mathrm{d}t$$

the local optimal average accumulated oil volume corresponding to L, U, v_0*, and the optimizer* $\mathbf{t}$ *is called the local optimal controller.*

Let $g(v_0, t_1, t_2, t_3, t_4) \mathrel{\widehat{=}} \frac{1}{20} \int_{t=0}^{20} v(t) \mathrm{d}t$, denoted by g for short. It can be computed from C_1, C_3, C_4 without considering fluctuations of consumption rate that

$$g = \frac{20v_0 + 1.1(t_1^2 - t_2^2 + t_3^2 - t_4^2 - 40t_1 + 40t_2 - 40t_3 + 40t_4) - 132.2}{20} \ .$$

Then by Proposition 1, $(\mathrm{R'_o})$ can be encoded into

$$\exists L, U. \Big(C_9 \wedge \forall v_0. \big(C_7 \longrightarrow \exists t_1 t_2 t_3 t_4. (\mathcal{D} \wedge g \le z) \big) \Big) \ , \tag{9}$$

which is equivalent to $z \ge z^*$ or $z > z^*$, where z^* equals the value of (8).

[7] The fact that each D_i is a nonempty closed set can be checked using QE.

5.2 Techniques for Performing QE

The above deduced (9) is a nonlinear formula with hundreds of atomic formulas and two alternations of quantifiers, for which QE tools such as Redlog [7] or QEPCAD [3] fail. Therefore we have developed specialized heuristics to decompose the QE problem into manageable parts.

Eliminating the Inner Quantifiers. We first eliminate the innermost quantified variables $\exists t_1 t_2 t_3 t_4$ by employing the theory of quadratic programming.

Note that D_i in $\mathcal{D}$ is a closed convex polyhedron for all i and g is a quadratic polynomial function, so minimization of g on D_i is a *quadratic programming* problem. Then the *Karush-Kuhn-Tucker* (KKT) [14] condition

$$\theta_{\text{kkt}} \mathrel{\widehat{=}} \exists \boldsymbol{\mu}.\, \mathcal{L}(g, D_i)\,, \tag{10}$$

where $\mathcal{L}(g, D_i)$ is a linear formula constructed from g and D_i, and $\boldsymbol{\mu}$ is a vector of new variables, gives a *necessary* condition for a local minimum of g on D_i.

By applying the KKT condition to each D_i and eliminating all $\boldsymbol{\mu}$, we can get a *necessary* condition $\mathcal{D}'$, a disjunction of 580 parts, for the minimum of g on $\mathcal{D}$:

$$\mathcal{D}' = \bigvee_{j=1}^{580} B_j\,.$$

Furthermore, each B_j has the nice property that for any L, U, v_0, a *unique* $\mathbf{t}_j$ is determined by B_j.[8] For instance, one of the B_j reads:

$$\begin{array}{l} t_4 = 20 \wedge 16t_2 + 10L - 349 = 0 \wedge \\ t_2 - t_3 + 2 = 0 \wedge 22t_1 - 16t_2 - 10v_0 + 107 = 0 \ \wedge \cdots \end{array} . \tag{11}$$

Since $\mathcal{D}'$ keeps the minimal value point of g on $\mathcal{D}$, the formula obtained by replacing $\mathcal{D}$ by $\mathcal{D}'$ in (9)

$$\exists L, U.\Big(C_9 \wedge \forall v_0.\big(C_7 \longrightarrow \exists t_1 t_2 t_3 t_4.(\mathcal{D}' \wedge g \le z)\big)\Big) \tag{12}$$

is equivalent to (9). Then according to formulas like (11), $\exists t_1 t_2 t_3 t_4$ in (12) can be eliminated by the distribution of $\exists$ among disjunctions, followed by instantiations of $\mathbf{t}_j$ in each disjunct. Thus (12) can be converted to

$$\exists L, U.\Big(C_9 \wedge \forall v_0.\big(C_7 \longrightarrow \bigvee_{j=1}^{580} (A_j \wedge g_j \le z)\big)\Big)\ , \tag{13}$$

where A_j is a constraint on L, U, v_0, and g_j is the instantiation of g using $\mathbf{t}_j$ given by formulas like (11).

---

[8] This has been verified by QE.

Eliminating the Outer Quantifiers. We eliminate the outermost quantifiers $\exists L, U$ in (13) by discretization, as discussed in Section 2.

According to C_9, the interval $[5.1, 24.9]$ is discretized with a granularity of 0.1 (the same granularity adopted in [4]), which gives a set of 199 elements. Then assignments to L, U from this set satisfying C_9 are used to instantiate (13). There are totally 15400 such pairs of L, U, e.g. $(5.1, 7.5)$, $(5.1, 7.6)$ etc, and as many instantiations in the form of

$$\forall v_0.\big(C_7 \longrightarrow \bigvee_{j=1}^{580} (A_j \wedge g_j \leq z)\big) \ , \tag{14}$$

each of which gives an optimal value corresponding to $[L, U]$. In practice, we start from $L = 5.1, U = 7.5$, and search for the minimal optimal value through all the 15400 cases with L or U incremented by 0.1 every iteration.

Eliminating the Middle Quantifier. We finally eliminate the only quantifier left in (14) by a divide-and-conquer strategy. First, we can show that

Lemma 2. $\bigvee_{j=1}^{580} A_j$ *is equivalent to* C_7 *in (14).*

By this lemma if all A_j are pairwise disjoint then (14) is equivalent to

$$\bigwedge_{j=1}^{580} \forall v_0.\big(v_0 \in A_j \longrightarrow (A_j \wedge g_j \leq z)\big) \, . \tag{15}$$

Since each conjunct in (15) is a small formula with only two variables v_0, z and one universal quantifier, it can be dealt with quite efficiently.

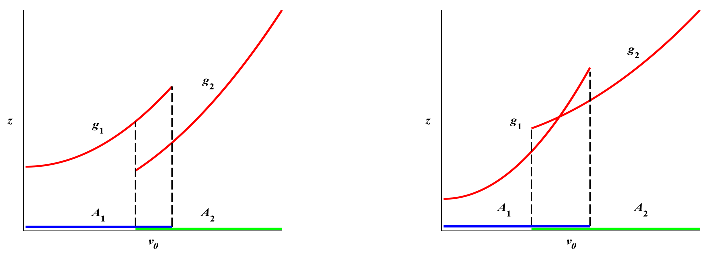

Fig. 3. Region partition.

If the set of A_j's are not pairwise disjoint, then we have to partition them into disjoint regions and assign a new cost function g'_k to each region. The idea for performing such partition is simple, which is illustrated by Fig. 3.

Suppose two sets, say A_1, A_2, are chosen arbitrarily from the set of A_j's. If $A_1 \cap A_2 = \emptyset$, then we do nothing. Otherwise check wether $g_1 \leq g_2$ (or $g_2 \leq g_1$)

on $A_1 \cap A_2$: if so, assign the smaller one, i.e. $g_1 \leq z$ (or $g_2 \leq z$) to $A_1 \cap A_2$; otherwise we simply assign $(g_1 \leq z) \vee (g_2 \leq z)$ to $A_1 \cap A_2$.

If at the same time of partitioning regions we also make a record of the local control strategy in each region, i.e. $\mathbf{t}_j$, then in the end we can get exactly the family of local optimal controllers corresponding to each v_0.

5.3 Results of QE

Various tools are available for doing QE. In our implementation, the SMT-based tool Mjollnir [20, 19] is chosen for QE on linear formulas, while REDLOG [7] implementing *virtual substitution* [18] is chosen for formulas with nonlinear terms. The computer algebra system REDUCE [12], of which REDLOG is an integral part, allows us to perform some programming tasks, e.g. region partition. Table 1 shows the performance of our approach. All experiments are done on a desktop running Linux with a 2.66 GHz CPU and 3 GB memory.

Table 1. Timing of different QE tasks.

formula	C_8	C_{10}	θ_{kkt} (all 92)	all the rest
tool	Mjollnir	Mjollnir	Mjollnir	Redlog/Reduce
time	8m8s	4m13s	31s	$<$1s

Remark In Table 1, timing is in minutes (m) and seconds (s); in the last column, the time taken to get the first optimal value[9] is less than 1 second, whereas all 15400 iterations will cost more than 10 hours (using a single computing process).

The final results are as follows:

– The interval that produces the optimal value is $[5.1, 7.5]$.
– The local optimal controller for $v_0 \in [5.1, 7.5]$ is

$$t_1 = \frac{10v_0 - 25}{13} \wedge t_2 = \frac{10v_0 + 1}{13} \wedge t_3 = \frac{10v_0 + 153}{22} \wedge t_4 = \frac{157}{11} \ , \qquad (16)$$

which is illustrated by Fig. 4. If $v_0 = 6.5$, then by (16) the pump should be switched on at $t_1 = 40/13$, off at $t_2 = 66/13$, then on at $t_3 = 109/11$, and finally off at $t_4 = 157/11$ (dashed line in Fig. 4).
– The optimal average accumulated oil volume $\frac{215273}{28600} = 7.53$ is obtained (dashed line in Fig. 5), improving by 5% the optimal value 7.95 in [4], which is already a 40% improvement of the controllers from the HYDAC company. The local optimal average accumulated oil volume for $v_0 \in [5.1, 7.5]$ under controller (16), i.e. $V_{aav}(v_0) = \frac{1300v_0^2+20420v_0+634817}{114400}$, is illustrated by Fig. 5.

[9] For the model with 2 activations, this optimal value is only obtained at the 1st iteration, using interval [5.1,7.5].

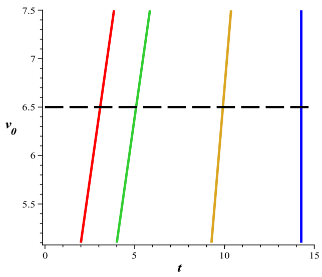

Fig. 4. Optimal controller.

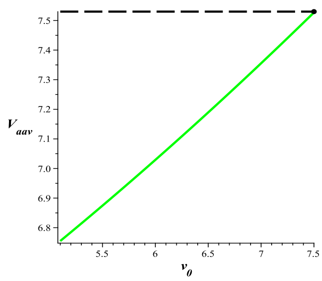

Fig. 5. Local optimal value for $v_0 \in [5.1, 7.5]$.

5.4 Improvement by Increasing Activation Times

In the controller shown by Fig. 4, we noticed that when v_0 is small and the pump is started on for the second time, it stays on for a period longer than 4 seconds. Based on this observation, we conjectured that if the pump is allowed to be activated three times in one cycle, then each time it could stay on for a shorter period, and the time it is activated for the third time can be postponed. As a result, the accumulated oil volume in one cycle may become less.

To verify the above conjecture, some modifications must be made on the previous model. Firstly, C_2 and C_3 should be replaced by their counterparts with 3 activations respectively; secondly, in C_5 and C_6 the tolerance of noises should be increased to 0.3, because due to the increase of times to operate the pump, the maximal uncertainty caused by imprecision in the measurement of volume and time is now $13.2\delta + \epsilon < 0.3$; thirdly, the objective function g should be recomputed.

For this model, using interval $[5.2, 8.1]$, the optimal average accumulated oil volume $6613/900 = 7.35$ is obtained, which is a 7.5% improvement over the optimum 7.95 in [4]. The explicit form of local optimal controllers like (16) can be found in [30].

Furthermore, the following theorem indicates that the theoretically optimal controller can be obtained using the local control strategy with 3 activations.

Theorem 1. *For each admissible* $[L, U]$, *each* $v_0 \in [L, U]$, *and any local control strategy* s_4 *with at least 4 activations subject to* R_{lu}, R_i *and* R_{ls}, *there exists a local control strategy* s_3 *subject to* R_{lu}, R_i *and* R_{ls} *with 3 activations such that* $\frac{1}{20}\int_{t=0}^{20} v_{s_3}(t)\mathrm{d}t < \frac{1}{20}\int_{t=0}^{20} v_{s_4}(t)\mathrm{d}t$, *where* $v_{s_3}(t)$ *(resp.* $v_{s_4}(t)$*) is the oil volume in the accumulator at* t *with* s_3 *(resp.* s_4*).*

6 Concluding Remarks

We propose a "hybrid" approach for synthesizing optimal controllers of hybrid systems subject to safety requirements by first reducing the problem to QE and then combining symbolic computation and numerical computation for scalability.

We illustrate our approach using a real industrial case of an oil pump provided by the HYDAC company.

Compared to the related work, e.g. [4], our approach has the following advantages. 1) Using first-order real arithmetic formulas to model the system, safety requirements as well as optimality objectives uniformly and succinctly, synthesis, verification and optimization are integrated into one elegant framework. The synthesized controllers are guaranteed to be correct. 2) By combining symbolic and numerical computation, we can obtain both high precision and efficiency: for the oil pump example, our approach can synthesize a better optimal controller (up to 7.5% improvement of [4]) in a reasonable amount of time (see Table 1). By Theorem 1, the synthesized controller is even theoretically optimal.

The issues of evaluation and implementation of our controllers are being considered. To make our approach more general with symbolic and numerical components, and apply it to more examples in practice will be our future work.

Acknowledgements. Special thanks go to Mr. Quan Zhao for his kind help in writing an interface between different QE tools, and to Dr. David Monniaux for his instructions on the use of the tool Mjollnir.

References

1. 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 (Jul 2000)
2. Bemporad, A., Morari, M., Dua, V., Pistikopoulos, E.N.: The explicit linear quadratic regulator for constrained systems. Automatica 38(1), 3–20 (2002)
3. Brown, C.W.: QEPCAD B: A program for computing with semi-algebraic sets using CADs. SIGSAM Bulletin 37, 97–108 (2003)
4. Cassez, F., Jessen, J.J., Larsen, K.G., Raskin, J.F., Reynier, P.A.: Automatic synthesis of robust and optimal controllers — an industrial case study. In: Majumdar, R., Tabuada, P. (eds.) HSCC'09. pp. 90–104. Springer, Berlin, Heidelberg (2009)
5. Chatterjee, K., de Alfaro, L., Majumdar, R., Raman, V.: Algorithms for game metrics (full version). Logical Methods in Computer Science 6(3) (2010), `http://arxiv.org/abs/0809.4326`
6. Davenport, J.H., Heintz, J.: Real quantifier elimination is doubly exponential. J. Symb. Comput. 5(1-2), 29–35 (1988)
7. Dolzmann, A., Seidl, A., Sturm, T.: Redlog User Manual (Nov 2006), `http://redlog.dolzmann.de/downloads/`, edition 3.1, for redlog Version 3.06 (reduce 3.8)
8. Dolzmann, A., Sturm, T., Weispfenning, V.: Real quantifier elimination in practice. In: Algorithmic Algebra and Number Theory. pp. 221–247. Springer (1998)
9. Fotiou, I.A., Rostalski, P., Parrilo, P.A., Morari, M.: Parametric optimization and optimal control using algebraic geometry methods. International Journal of Control 79(11), 1340–1358 (2006)
10. Frehse, G.: PHAVer: algorithmic verification of hybrid systems past HyTech. Int. J. Softw. Tools Technol. Transf. 10(3), 263–279 (May 2008)
11. Gulwani, S., Tiwari, A.: Constraint-based approach for analysis of hybrid systems. In: Gupta, A., Malik, S. (eds.) CAV'08. LNCS, vol. 5123, pp. 190–203. Springer, Berlin, Heidelberg (2008)

12. Hearn, A.C.: Reduce User's Manual (Feb 2004), `http://reduce-algebra.com/docs/reduce.pdf`, version 3.8
13. Henzinger, T., Ho, P.H., Wong-Toi, H.: HyTech: A model checker for hybrid systems. In: Grumberg, O. (ed.) CAV'97. LNCS, vol. 1254, pp. 460–463. Springer, Berlin, Heidelberg (1997)
14. Jensen, P.A., Bard, J.F.: Operations Research Models and Methods. John Wiley & Sons (Oct 2002)
15. Jha, S., Seshia, S.A., Tiwari, A.: Synthesis of optimal switching logic for hybrid systems. In: EMSOFT'11. pp. 107–116. ACM, New York, NY, USA (2011)
16. Kanno, M., Yokoyama, K., Anai, H., Hara, S.: Symbolic optimization of algebraic functions. In: ISSAC'08. pp. 147–154. ACM, New York, NY, USA (2008)
17. Liu, J., Zhan, N., Zhao, H.: Computing semi-algebraic invariants for polynomial dynamical systems. In: EMSOFT'11. pp. 97–106. ACM, New York, NY, USA (2011)
18. Loos, R., Weispfenning, V.: Applying linear quantifier elimination. The Computer Journal 36(5), 450–462 (May 1993)
19. Monniaux, D.: Mjollnir-2009-07-10, `http://www-verimag.imag.fr/~monniaux/mjollnir.html`
20. Monniaux, D.: A quantifier elimination algorithm for linear real arithmetic. In: Cervesato, I., Veith, H., Voronkov, A. (eds.) LPAR'08. pp. 243–257. Springer, Berlin, Heidelberg (2008)
21. Platzer, A., Clarke, E.M.: Computing differential invariants of hybrid systems as fixedpoints. In: Gupta, A., Malik, S. (eds.) CAV'08. LNCS, vol. 5123, pp. 176–189. Springer, Berlin, Heidelberg (2008)
22. Platzer, A., Clarke, E.M.: Formal verification of curved flight collision avoidance maneuvers: A case study. In: Cavalcanti, A., Dams, D. (eds.) FM'09. LNCS, vol. 5850, pp. 547–562. Springer, Berlin, Heidelberg (2009)
23. Sankaranarayanan, S., Sipma, H.B., Manna, Z.: Constructing invariants for hybrid systems. In: Alur, R., Pappas, G. (eds.) HSCC'04. LNCS, vol. 2993, pp. 539–554. Springer, Berlin, Heidelberg (2004)
24. Sturm, T., Tiwari, A.: Verification and synthesis using real quantifier elimination. In: ISSAC'11. pp. 329–336. ACM, New York, NY, USA (2011)
25. Taly, A., Gulwani, S., Tiwari, A.: Synthesizing switching logic using constraint solving. In: Jones, N., Müller-Olm, M. (eds.) VMCAI'09. LNCS, vol. 5403, pp. 305–319. Springer, Berlin, Heidelberg (2009)
26. Taly, A., Tiwari, A.: Switching logic synthesis for reachability. In: EMSOFT'10. pp. 19–28. ACM, New York, NY, USA (2010)
27. Tarski, A.: A Decision Method for Elementary Algebra and Geometry. University of California Press, Berkeley (May 1951)
28. 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 (Jul 2000)
29. Weispfenning, V.: Parametric linear and quadratic optimization by elimination. Tech. rep., Fakultät für Mathematik und Informatik, Universität Passau (1994)
30. 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. CoRR abs/1203.6025 (2012), `http://arxiv.org/abs/1203.6025`