

Optimal and Robust Controller Synthesis*

Using Energy Timed Automata with Uncertainty

Giovanni Bacci¹, Patricia Bouyer², Uli Fahrenberg³, Kim G. Larsen¹,
Nicolas Markey⁴, and Pierre-Alain Reynier⁵

¹ Department of Computer Science, Aalborg University, Denmark

² LSV, CNRS & ENS Cachan, Université Paris-Saclay, Cachan, France

³ École Polytechnique, Palaiseau, France

⁴ Univ. Rennes, IRISA, CNRS & INRIA, Rennes, France

⁵ Aix Marseille Univ., Université de Toulon, CNRS, LIS, Marseille, France

Abstract. In this paper, we propose a novel framework for the synthesis of robust and optimal energy-aware controllers. The framework is based on energy timed automata, allowing for easy expression of timing constraints and variable energy rates. We prove decidability of the energy-constrained infinite-run problem in settings with both certainty and uncertainty of the energy rates. We also consider the optimization problem of identifying the minimal upper bound that will permit existence of energy-constrained infinite runs. Our algorithms are based on quantifier elimination for linear real arithmetic. Using Mathematica and Mjollnir, we illustrate our framework through a real industrial example of a hydraulic oil pump. Compared with previous approaches our method is completely automated and provides improved results.

1 Introduction

Design of controllers for embedded systems is a difficult engineering task. Controllers must ensure a variety of safety properties as well as optimality with respect to given performance properties. Also, for several systems, e.g. [8,27,25], the properties involve non-functional aspects such as time and energy.

We provide a novel framework for automatic synthesis of safe and optimal controllers for resource-aware systems based on *energy timed automata*. Synthesis of controllers is obtained by solving time- and energy-constrained infinite run problems. Energy timed automata [12] extend timed automata [2] with a continuous *energy* variable that evolves with varying rates and discrete updates during the behaviour of the model. Closing an open problem from [12], we prove decidability of the infinite run problem in settings, where rates and updates may be both positive and negative and possibly subject to uncertainty. Additionally, the accumulated energy may be subject to lower and upper bounds reflecting constraints on capacity. Also we consider the optimization problems of identifying minimal upper bounds that will permit the existence of infinite energy-constrained runs. Our decision and optimization algorithms for the energy-constrained infinite run problems

* Work supported by ERC projects Lasso and EQualIS.

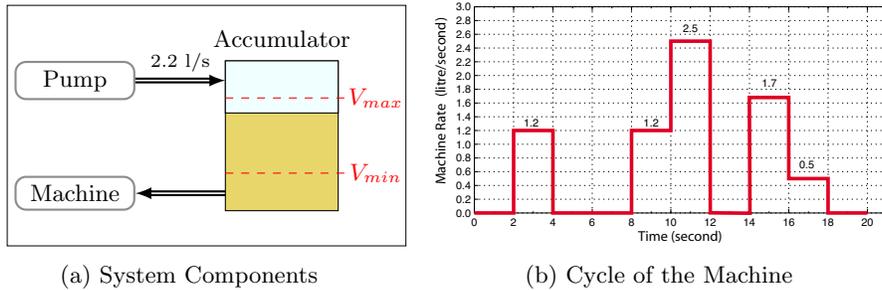


Fig. 1: Overview of the HYDAC system

are based on reductions to quantifier elimination (QE) for linear real arithmetic, for which we combine Mathematica [28] and Mjollnir [24] into a tool chain.

To demonstrate the applicability of our framework, we revisit an industrial case study provided by the HYDAC company in the context of the European project Quasimodo [26]. It consists in an on/off control system (see Fig. 1a) composed of (i) a machine that consumes oil according to a cyclic pattern of 20 s (see Fig. 1b), (ii) an accumulator containing oil and a fixed amount of gas in order to put the oil under pressure, and (iii) a controllable pump which can pump oil into the accumulator with rate 2.2 l/s. The control objective for switching the pump on and off is twofold: first the level of oil in the accumulator (and so the gas pressure) shall be maintained within a safe interval; second, the controller should try to minimize the (maximum and average) level of oil such that the pressure in the system is kept minimal. We show how to model this system, with varying constraints on pump operation, as energy timed automata. Thus our tool chain may automatically synthesize guaranteed safe and optimal control strategies.

The HYDAC case was first considered in [16] as a timed game using the tool UPPAAL-TIGA [15,5] for synthesis. Discretization of oil-level (and time) was used to make synthesis feasible. Besides limiting the opportunity of optimality, the discretization also necessitated posterior verification using PHAVER [20] to rule out possible resulting incorrectness. Also, identification of safety and minimal oil levels were done by manual and laborious search. In [23] the timed game models of [16] (rephrased as Timed Discrete Event Systems) are reused, but BDDs are applied for compact representation of the discrete oil-levels and time-points encountered during synthesis. [21] provides a framework for learning optimal switching strategies by a combination of off-the-shelf numerical optimization and generalization by learning. The HYDAC case is one of the considered cases. The method offers no absolute guarantees of hard constraints on energy-level, but rather attempts to enforce these through the use of high penalties. [29] focuses exclusively on the HYDAC case using a direct encoding of the safety- and optimality-constraints as QE problems. This gives—like in our case—absolute guarantees. However, we are additionally offering a complete and decidable framework based on energy timed automata, which extends to several other

systems. Moreover, the controllers we obtain perform significantly better than those of [16] and [29] (respectively up to 22% and 16% better) and are obtained automatically by our tool chain combining Mjollnir and Mathematica. This combination permits quantifier elimination and formula simplification to be done in a compositional manner, resulting in performance surpassing each tool individually. We believe that this shows that our framework has a level of maturity that meets the complexity of several relevant industrial control problems.

Our work is related to controllability of (constrained) piecewise affine (PWA) [7] and hybrid systems [1]. In particular, the energy-constrained infinite-run problem is related to the so called *stability problem* for PWAs. Blondel and Tsitsiklis [10] have shown that verifying stability of autonomous piecewise-linear (PWL) systems is NP-hard, even in the simple case of two-component subsystems; several global properties (e.g. global convergence, asymptotic stability and mortality) of PWA systems have been shown undecidable in [9].

2 Energy Timed Automata

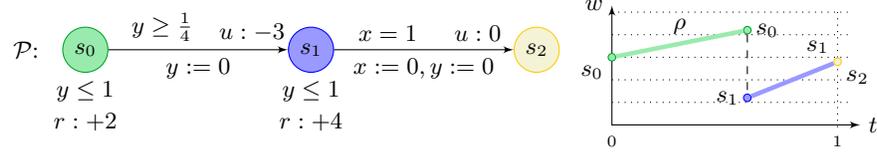
Definitions. Given a finite set X of clocks, the set of *closed clock constraints* over X , denoted $C(X)$, is the set of formulas built using $g ::= x \sim n \mid g \wedge g$, where x ranges over X , \sim ranges over $\{\leq, \geq\}$ and n ranges over $\mathbb{Q}_{\geq 0}$. That a clock valuation $v: X \rightarrow \mathbb{R}_{\geq 0}$ satisfies a clock constraint g , denoted $v \models g$, is defined in the natural way. For a clock valuation v , a real $t \in \mathbb{R}_{\geq 0}$, and a subset $R \subseteq X$, we write $v + t$ for the valuation mapping each clock $x \in X$ to $v(x) + t$, and $v[R \rightarrow 0]$ for the valuation mapping clocks in R to zero and clocks not in R to their value in v . Finally we write $\mathbf{0}_X$ (or simply $\mathbf{0}$) for the clock valuation assigning 0 to every $x \in X$.

For $E \subseteq \mathbb{R}$, we let $\mathcal{I}(E)$ be the set of closed intervals of \mathbb{R} with bounds in $E \cap \mathbb{Q}$. Notice that any interval in $\mathcal{I}(E)$ is bounded, for any $E \subseteq \mathbb{R}$.

Definition 1. An energy timed automaton (ETA for short; a.k.a. priced or weighted timed automaton [3,6]) is a tuple $\mathcal{A} = (S, S_0, X, I, r, T)$ where S is a finite set of states, $S_0 \subseteq S$ is the set of initial states, X is a finite set of clocks, $I: S \rightarrow C(X)$ assigns invariants to states, $r: S \rightarrow \mathbb{Q}$ assigns rates to states, and $T \subseteq S \times C(X) \times \mathbb{Q} \times 2^X \times S$ is a finite set of transitions.

An energy timed path (ETP, a.k.a. linear energy timed automaton) is an energy timed automaton for which S can be written as $\{s_i \mid 0 \leq i \leq n\}$ in such a way that $S_0 = \{s_0\}$, and $T = \{(s_i, g_i, u_i, z_i, s_{i+1}) \mid 0 \leq i < n\}$. We additionally require that all clocks are reset on the last transition, i.e., $z_{n-1} = X$.

Let $\mathcal{A} = (S, S_0, X, I, r, T)$ be an ETA. A *configuration* of \mathcal{A} is a triple $(\ell, v, w) \in S \times (\mathbb{R}_{\geq 0})^X \times \mathbb{R}$, where v is a clock valuation, and w is the energy level. Let $\tau = (t_i)_{0 \leq i < n}$ be a finite sequence of transitions, with $t_i = (s_i, g_i, u_i, z_i, s_{i+1})$ for every i . A finite *run* in \mathcal{A} on τ is a sequence of configurations $\rho = (\ell_j, v_j, w_j)_{0 \leq j < 2n}$ such that there exists a sequence of delays $(d_i)_{0 \leq i < n}$ for which the following requirements hold:

Fig. 2: An energy timed path \mathcal{P} , and a run ρ of \mathcal{P} with initial energy level 3.

- for all $0 \leq j < n$, $\ell_{2j} = \ell_{2j+1} = s_j$, and $\ell_{2n} = s_n$;
- for all $0 \leq j < n$, $v_{2j+1} = v_{2j} + d_j$ and $v_{2j+2} = v_{2j+1}[z_j \rightarrow 0]$;
- for all $0 \leq j < n$, $v_{2j} \models I(s_j)$ and $v_{2j+1} \models I(s_j) \wedge g_j$;
- for all $0 \leq j < n$, $w_{2j+1} = w_{2j} + d_j \cdot r(s_j)$ and $w_{2j+2} = w_{2j+1} + u_j$.

We will by extension speak of runs read on ETPs (those runs will then end with clock valuation $\mathbf{0}$). The notion of infinite run is defined similarly. Given $E \in \mathcal{I}(\mathbb{Q})$, such a run is said to satisfy energy constraint E if $w_j \in E$ for all j .

Example 1. Fig. 2 displays an example of an ETP \mathcal{P} and one of its runs ρ . Since no time will be spent in s_2 , we did not indicate the invariant and rate of that state. The sequence ρ is a run of \mathcal{P} . Spending 0.6 time units in s_0 , the value of clock x reaches 0.6, and the energy level grows to $3 + 0.6 \times 2 = 4.2$; it equals $4.2 - 3 = 1.2$ when entering s_1 . Then ρ satisfies energy constraint $[0; 5]$. \triangleleft

Definition 2. A segmented energy timed automaton (SETA for short) is a tuple $\mathcal{A} = (S, T, P)$ where (S, T) is a finite graph (whose states and transitions are called macro-states and macro-transitions), S_0 is a set of initial macro-states, and P associates with each macro-transition $t = (s, s')$ of \mathcal{A} an ETP with initial state s and final state s' . We require that for any two different transitions t and t' of \mathcal{A} , the state spaces of $P(t)$ and $P(t')$ are disjoint and contain no macro-states, except (for both conditions) for their first and last states.

A SETA is flat if the underlying graph (S, T) is (i.e., for any $s \in S$, there is at most one non-empty path in the graph (S, T) from s to itself [17,14]). It is called depth-1 whenever the graph (S, T) is tree-like, with only loops at leaves.

A (finite or infinite) execution of a SETA is a (finite or infinite) sequence of runs $\rho = (\rho^i)_i$ such that for all i , writing $\rho^i = (\ell_j^i, v_j^i, w_j^i)_{0 \leq j \leq 2n_i}$, it holds:

- ℓ_0^i and $\ell_{2n_i}^i$ are macro-states of \mathcal{A} , and ρ^i is a run of the ETP $P(\ell_0^i, \ell_{2n_i}^i)$;
- $\ell_0^{i+1} = \ell_{2n_i}^i$ and $w_0^{i+1} = w_{2n_i}^i$.

Hence a run in a SETA should be seen as the concatenation of paths ρ^i between macro-states. Notice also that each ρ^i starts and ends with all clock values zero, since all clocks are reset at the end of each ETP, when a main state is entered. Finally, given an interval $E \in \mathcal{I}(\mathbb{Q})$, an execution $(\rho^i)_i$ satisfies energy constraint E whenever all individual runs ρ^i do.

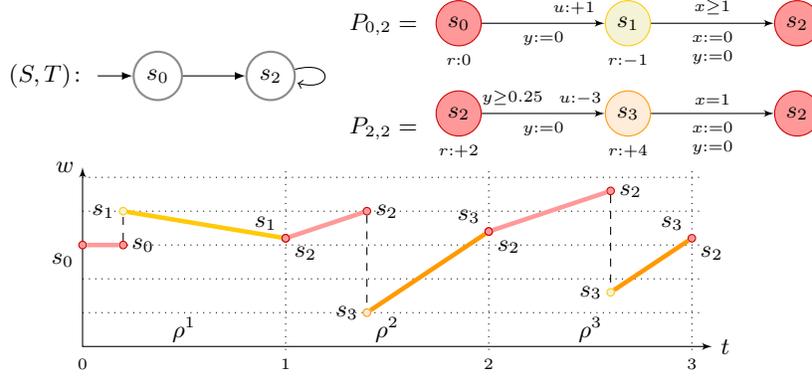


Fig. 3: A SETA $\mathcal{A} = (S, T, P)$ with implicit global invariant $y \leq 1$; omitted discrete updates are assumed to be zero. The map P associates with each $(s_i, s_j) \in T$ the ETP $P_{i,j}$. The infinite sequence $\rho^1 \cdot (\rho^2 \cdot \rho^3)^\omega$ is an infinite execution of \mathcal{A} with initial energy level 3 satisfying the energy constraint $E = [0; 5]$.

Remark 1. In contrast with ETAs, the class of SETAs is not closed under parallel composition. Intuitively, the ETA resulting from the parallel composition of two SETAs may not be “segmented” into a graph of energy timed-paths because the requirement that all clocks are reset on the last transition may not be satisfied. Furthermore, parallel composition does not preserve flatness because it may introduce nested loops.

Example 2. Figure 3 displays a SETA \mathcal{A} with two macro-states s_0 and s_2 , and two macro-transitions. The macro-self-loop on s_2 is associated with the energy timed path of Fig. 2. The execution $\rho = \rho^1 \cdot (\rho^2 \cdot \rho^3)^\omega$ is an ultimately-periodic execution of \mathcal{A} . This infinite execution satisfies the energy constraint $E = [0; 5]$ (as well as the (tight) energy constraint $[1; 4.6]$). \triangleleft

In this paper, we consider the following *energy-constrained infinite-run problem* [12]: given an energy timed automaton \mathcal{A} and a designated state s_0 , an energy constraint $E \in \mathcal{I}(\mathbb{Q})$ and an initial energy level $w_0 \in E$, does there exist an infinite execution in \mathcal{A} starting from $(s_0, \mathbf{0}, w_0)$ that satisfies E ?

In the general case, the energy-constrained infinite-run problem is undecidable, even when considering ETA with only two clocks [22]. In this paper, we prove:

Theorem 3. *The energy-constrained infinite-run problem is decidable for flat SETA.*

Theorem 4. *Given a fixed lower bound L , the existence of an upper bound U , such that there is a solution to the energy-constrained infinite-run problem for energy constraint $E = [L; U]$, is decidable for flat SETA. If such a U exists, then for depth-1 flat SETA, we can compute the least one.*

We only sketch a proof of the former result, and refer to [4] for the full proof.

Binary energy relations. Let $\mathcal{P} = (\{s_i \mid 0 \leq i \leq n\}, \{s_0\}, X, I, r, T)$ be an ETP from s_0 to s_n . Let $E \subseteq \mathcal{I}(\mathbb{Q})$ be an energy constraint. The *binary energy relation* $\mathcal{R}_{\mathcal{P}}^E \subseteq E \times E$ for \mathcal{P} under energy constraint E relates all pairs (w_0, w_1) for which there is a finite run of \mathcal{P} from $(s_0, \mathbf{0}, w_0)$ to $(s_n, \mathbf{0}, w_1)$ satisfying energy constraint E . This relation is characterized by the following first-order formula:

$$\mathcal{R}_{\mathcal{P}}^E(w_0, w_1) \iff \exists (d_i)_{0 \leq i < n}. \Phi_{\text{timing}} \wedge \Phi_{\text{energy}} \wedge w_1 = w_0 + \sum_{k=0}^{n-1} (d_k \cdot r(s_k) + u_k)$$

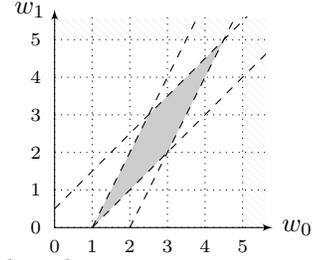
where Φ_{timing} encodes all the timing constraints that the sequence $(d_i)_{0 \leq i < n}$ has to fulfill (derived from guards and invariants, by expressing the values of the clocks in terms of $(d_i)_{0 \leq i < n}$), while Φ_{energy} encodes the energy constraints (in each state, the accumulated energy must be in E).

It is easily shown that $\mathcal{R}_{\mathcal{P}}^E$ is a closed, convex subset of $E \times E$ (remember that we consider closed clock constraints); thus it can be described as a conjunction of a finite set of linear constraints over w_0 and w_1 (with non-strict inequalities), using quantifier elimination of variables $(d_i)_{0 \leq i < n}$.

Example 3. We illustrate this computation on the ETP of Fig. 2. For energy constraint $[0; 5]$, the energy relation (after removing redundant constraints) reads as

$$\begin{aligned} \mathcal{R}_{\mathcal{P}}^E(w_0, w_1) \iff \exists d_0, d_1. & d_0 \in [0.25; 1] \wedge d_1 \in [0; 1] \wedge d_0 + d_1 = 1 \wedge \\ & w_0 \in [0; 5] \wedge w_0 + 2d_0 \in [0; 5] \wedge w_0 + 2d_0 - 3 \in [0; 5] \wedge \\ & w_1 = w_0 + 2d_0 + 4d_1 - 3 \wedge w_1 \in [0; 5]. \end{aligned}$$

This simplifies to $(w_1 + 2 \leq 2w_0 \leq w_1 + 4) \wedge (w_1 - 0.5 \leq w_0 \leq w_1 + 1)$. The corresponding polyhedron is depicted above. \triangleleft



Energy functions. We now focus on properties of energy relations. First notice that for any interval $E \in \mathcal{I}(\mathbb{Q})$, the partially-ordered set $(\mathcal{I}(E), \supseteq)$ is ω -complete, meaning that for any chain $(I_j)_{j \in \mathbb{N}}$, with $I_j \supseteq I_{j+1}$ for all j , the limit $\bigcap_{j \in \mathbb{N}} I_j$ also belongs to $\mathcal{I}(E)$. By Cantor's Intersection Theorem, if additionally each interval I_j is non-empty, then so is the limit $\bigcap_{j \in \mathbb{N}} I_j$.

With an energy relation $\mathcal{R}_{\mathcal{P}}^E$, we associate an *energy function* (also denoted with $\mathcal{R}_{\mathcal{P}}^E$, or simply \mathcal{R} , as long as no ambiguity may arise), defined for any closed sub-interval $I \in \mathcal{I}(E)$ as $\mathcal{R}(I) = \{w_1 \in E \mid \exists w_0 \in I. \mathcal{R}(w_0, w_1)\}$. Symmetrically:

$$\mathcal{R}^{-1}(I) = \{w_0 \in E \mid \exists w_1 \in I. \mathcal{R}(w_0, w_1)\}.$$

Observe that $\mathcal{R}(I)$ and $\mathcal{R}^{-1}(I)$ also belong to $\mathcal{I}(E)$ (because the relation \mathcal{R} is closed and convex). Moreover, \mathcal{R} and \mathcal{R}^{-1} are non-decreasing: for any two intervals I and J in $\mathcal{I}(E)$ such that $I \subseteq J$, it holds $\mathcal{R}(I) \subseteq \mathcal{R}(J)$ and $\mathcal{R}^{-1}(I) \subseteq \mathcal{R}^{-1}(J)$. Energy function \mathcal{R}^{-1} also satisfies the following continuity property:

Lemma 5. *Let $(I_j)_{j \in \mathbb{N}}$ be a chain of intervals of $\mathcal{I}(E)$, such that $I_j \supseteq I_{j+1}$ for all $j \in \mathbb{N}$. Then $\mathcal{R}^{-1}(\bigcap_{j \in \mathbb{N}} I_j) = \bigcap_{j \in \mathbb{N}} \mathcal{R}^{-1}(I_j)$.*

Composition and fixpoints of energy functions. Consider a finite sequence of paths $(\mathcal{P}_i)_{1 \leq i \leq k}$. Clearly, the energy relation for this sequence can be obtained as the composition of the individual energy relations $\mathcal{R}_{\mathcal{P}_k}^E \circ \dots \circ \mathcal{R}_{\mathcal{P}_1}^E$; the resulting energy relation still is a closed convex subset of $E \times E$ that can be described as the conjunction of finitely many linear constraints over w_0 and w_1 . As a special case, we write $(\mathcal{R}_{\mathcal{P}}^E)^k$ for the composition of k copies of the same relations $\mathcal{R}_{\mathcal{P}}^E$.

Now, using Lemma 5, we easily prove that the greatest fixpoint $\nu\mathcal{R}^{-1}$ of \mathcal{R}^{-1} in the complete lattice $(\mathcal{I}(E), \supseteq)$ exists and equals:

$$\nu\mathcal{R}^{-1} = \bigcap_{i \in \mathbb{N}} (\mathcal{R}^{-1})^i(E).$$

Moreover $\nu\mathcal{R}^{-1}$ is a closed (possibly empty) interval. Note that $\nu\mathcal{R}^{-1}$ is the maximum subset $S_{\mathcal{R}}$ of E such that, starting with any $w_0 \in S_{\mathcal{R}}$, it is possible to iterate \mathcal{R} infinitely many times (that is, for any $w_0 \in S_{\mathcal{R}}$, there exists $w_1 \in S_{\mathcal{R}}$ such that $\mathcal{R}(w_0, w_1)$ —any such set S is a post-fixpoint of \mathcal{R}^{-1} , i.e. $S \subseteq \mathcal{R}^{-1}(S)$).

If \mathcal{R} is the energy relation of a cycle \mathcal{C} in the flat SETA, then $\nu\mathcal{R}^{-1}$ precisely describes the set of initial energy levels allowing infinite runs through \mathcal{C} satisfying the energy constraint E . If \mathcal{R} is described as the conjunction $\phi_{\mathcal{C}}$ of a finite set of linear constraints, then we can characterize those intervals $[a, b] \subseteq E$ that constitute a post-fixpoint for \mathcal{R}^{-1} by the following first-order formula:

$$a \leq b \wedge a \in E \wedge b \in E \wedge \forall w_0 \in [a; b]. \exists w_1 \in [a; b]. \phi_{\mathcal{C}}(w_0, w_1). \quad (1)$$

Applying quantifier elimination (to w_0 and w_1), the above formula may be transformed into a direct constraint on a and b , characterizing all post-fixpoints of \mathcal{R}^{-1} . We get a characterization of $\nu\mathcal{R}^{-1}$ by computing the values of a and b that satisfy these constraint and maximize $b - a$.

Example 4. We again consider the flat SETA of Fig. 3, and consider the energy constraint $E = [0; 5]$. We first focus on the cycle \mathcal{C} on the macro-state s_2 : using the energy relation computed in Example 3, our first-order formula for the fixpoint then reads as follows:

$$0 \leq a \leq b \leq 5 \wedge \forall w_0 \in [a; b]. \exists w_1 \in [a; b]. \\ ((w_1 + 2 \leq 2w_0 \leq w_1 + 4) \wedge (w_1 - 0.5 \leq w_0 \leq w_1 + 1)).$$

Applying quantifier elimination, we end up with $2 \leq a \leq b \leq 4$. The maximal fixpoint then is $[2; 4]$. Similarly, for the path \mathcal{P} from s_0 to s_2 :

$$\mathcal{R}_{\mathcal{P}}^E(w_0, w_1) \iff \exists d_0, d_1. 0 \leq d_0 \leq 1 \wedge 0 \leq d_1 \leq 1 \wedge d_0 + d_1 \geq 1 \wedge \\ 0 \leq w_0 \leq 5 \wedge 0 \leq w_0 + 1 \leq 5 \wedge w_1 = w_0 + 1 - d_1 \wedge 0 \leq w_1 \leq 5$$

which reduces to $0 \leq w_0 \leq 4 \wedge w_0 \leq w_1 \leq w_0 + 1$. Finally, the initial energy levels w_0 for which there is an infinite-run in the whole SETA are characterized by $\exists w_1. (0 \leq w_0 \leq 4 \wedge w_0 \leq w_1 \leq w_0 + 1) \wedge (2 \leq w_1 \leq 4)$, which reduces to $1 \leq w_0 \leq 4$. \triangleleft

| | |
|--|---|
| Input: A flat SETA $\mathcal{A} = (S, T, P)$; initial state $m_0 \in S$; energy interval I_0 | |
| 1. $W \leftarrow \{(m_0, I_0, c)\}$ | \triangleleft initialize the waiting list |
| 2. while $W \neq \emptyset$ do | |
| 3. pick $(m, I, flag) \in W$ | \triangleleft pick an element from the waiting list |
| 4. $W \leftarrow W \setminus (m, I, flag)$ | \triangleleft remove the element from the waiting list |
| 5. if $flag = \bar{c}$ then | \triangleleft the node m shall be explored without following a cycle |
| 6. for each $(m, m') \in T$ that is not part of a simple cycle of (S, T) do | |
| 7. $W \leftarrow W \cup \{(m', \mathcal{R}_{P(m, m')}^E(I), c)\}$ | \triangleleft add this new task to the waiting list |
| 8. else | \triangleleft the node m shall be explored by following a cycle |
| 9. if m belongs to a cycle of (S, T) then | |
| 10. let $\mathcal{C} = (m_1, m_2) \cdots (m_k, m_{k+1})$ be the simple cycle s.t. $m = m_1 = m_{k+1}$ | |
| 11. let $\mathcal{R}_{\mathcal{C}} = \mathcal{R}_{P(m_k, m_{k+1})} \circ \cdots \circ \mathcal{R}_{P(m_1, m_2)}$ | \triangleleft energy relation of the cycle |
| 12. if $I \cap \nu \mathcal{R}_{\mathcal{C}}^{-1} \neq \emptyset$ then | \triangleleft check if there is an infinite run via the cycle \mathcal{C} |
| 13. return tt | |
| 14. else | \triangleleft the cycle can be executed only finitely many times |
| 15. $W \leftarrow W \cup \{(m, I, \bar{c})\}$ | \triangleleft add a new task to the waiting list |
| 16. $i \leftarrow 0$ | \triangleleft initialize the number of cycle executions |
| 17. while $\mathcal{R}_{\mathcal{C}}^i(I) \neq \emptyset$ do | \triangleleft while i -th energy relation is satisfied |
| 18. for $1 \leq j < k$ do | |
| 19. let $\mathcal{R}_{\mathcal{P}_j} = \mathcal{R}_{P(m_j, m_{j+1})} \circ \cdots \circ \mathcal{R}_{P(m_1, m_2)}$ | \triangleleft unfold \mathcal{C} up to m_{j+1} |
| 20. $W \leftarrow W \cup \{(m_{j+1}, \mathcal{R}_{\mathcal{P}_j}^i(I), \bar{c})\}$ | \triangleleft add a task to the waiting list |
| 21. $i \leftarrow i + 1$ | \triangleleft increment the number of cycle executions |
| 22. else | \triangleleft m doesn't belong to a cycle |
| 23. $W \leftarrow W \cup \{(m, I, \bar{c})\}$ | \triangleleft add a new task to the waiting list |
| 24. return ff | \triangleleft no infinite run could be found |

Algorithm 1: Existence of energy-constrained infinite runs in flat SETA

Algorithm for flat segmented energy timed automata. Following Example 4, we now prove that we can solve the energy-constrained infinite-run problem for any flat SETA. The next theorem is crucial for our algorithm:

Theorem 6. *Let \mathcal{R} be the energy relation of an ETP \mathcal{P} with energy constraint E , and let $I \in \mathcal{I}(E)$. Then either $I \cap \nu \mathcal{R}^{-1} \neq \emptyset$ or $\mathcal{R}^n(I) = \emptyset$ for some n .*

It follows that the energy-constrained infinite-run problem is decidable for flat SETAs. The decision procedure traverses the underlying graph of \mathcal{A} , forward propagating an initial energy interval $I_0 \subseteq E$ looking for a simple cycle \mathcal{C} such that $\nu \mathcal{R}_{\mathcal{C}}^{-1} \cap I \neq \emptyset$, where $I \subseteq E$ is the energy interval forward-propagated until reaching the cycle. Algorithm 1 gives a detailed description of the decision procedure. It traverses the underlying graph (S, T) of the flat SETA \mathcal{A} , using a waiting list W to keep track of the macro-states that need to be further explored. The list W contains tasks of the form $(m, I, flag)$ where $m \in S$ is the current macro-state, $I \in \mathcal{I}(E)$ is the current energy interval, and $flag \in \{c, \bar{c}\}$ is a flag indicating if m shall be explored by following a cycle it belongs to ($flag = c$), or proceed by exiting that cycle ($flag = \bar{c}$). Theorem 6 ensures termination of the **while** loop of lines 17-21, whereas flatness ensures the correctness of Algorithm 1.

It is worth noting that the flatness assumption for the SETA \mathcal{A} implies that the graph (S, T) has finitely many cycles (each macro-state belongs to at most one simple cycle of (S, T) , therefore the number of cycles is bounded by the number of macro-states). As a consequence, Algorithm 1 performs in the worst case an exhaustive search of all cycles in \mathcal{A} . The technique does not trivially extend to SETAs with nested cycles, because they may have infinitely many cycles.

3 Energy Timed Automata with Uncertainties

The assumptions of perfect knowledge of energy-rates and energy-updates are often unrealistic, as is the case in the HYDAC oil-pump control problem (see Section 4). Rather, the knowledge of energy-rates and energy-updates comes with a certain imprecision, and the existence of energy-constrained infinite runs must take these into account in order to be robust. In this section, we revisit the energy-constrained infinite-run problem in the setting of imprecisions, by viewing it as a two-player game problem.

Adding uncertainty to ETA. An *energy timed automaton with uncertainty* (ETAu for short) is a tuple $\mathcal{A} = (S, S_0, X, I, r, T, \epsilon, \Delta)$, where (S, S_0, X, I, r, T) is an energy timed automaton, with $\epsilon: S \rightarrow \mathbb{Q}_{>0}$ assigning imprecisions to rates of states and $\Delta: T \rightarrow \mathbb{Q}_{>0}$ assigning imprecisions to updates of transitions. This notion of uncertainty extends to *energy timed path with uncertainty* (ETPu) and to *segmented energy timed automaton with uncertainty* (SETAu).

Let $\mathcal{A} = (S, S_0, X, I, r, T, \epsilon, \Delta)$ be an ETAu, and let $\tau = (t_i)_{0 \leq i < n}$ be a finite sequence of transitions, with $t_i = (s_i, g_i, u_i, z_i, s_{i+1})$ for every i . A finite *run* in \mathcal{A} on τ is a sequence of configurations $\rho = (\ell_j, v_j, w_j)_{0 \leq j \leq 2n}$ such that there exist a sequence of delays $d = (d_i)_{0 \leq i < n}$ for which the following requirements hold:

- for all $0 \leq j < n$, $\ell_{2j} = \ell_{2j+1} = s_j$, and $\ell_{2n} = s_n$;
- for all $0 \leq j < n$, $v_{2j+1} = v_{2j} + d_j$ and $v_{2j+2} = v_{2j+1}[z_j \rightarrow 0]$;
- for all $0 \leq j < n$, $v_{2j} \models I(s_j)$ and $v_{2j+1} \models I(s_j) \wedge g_j$;
- for all $0 \leq j < n$, it holds that $w_{2j+1} = w_{2j} + d_j \cdot \alpha_j$ and $w_{2j+2} = w_{2j+1} + \beta_j$, where $\alpha_j \in [r(s_j) - \epsilon(s_j), r(s_j) + \epsilon(s_j)]$ and $\beta_j \in [u_j - \Delta(t_j), u_j + \Delta(t_j)]$.

We say that ρ is a possible outcome of d along τ , and that w_{2n} is a possible final energy level for d along τ , given initial energy level w_0 . Note that due to uncertainty, a given delay sequence d may have several possible outcomes (and corresponding energy levels) along a given transition sequence τ . In particular, we say that τ together with d and initial energy level w_0 satisfy an energy constraint $E \in \mathcal{I}(\mathbb{Q})$ if any possible outcome run ρ for t and d starting with w_0 satisfies E . All these notions are formally extended to ETPu.

Given an ETPu \mathcal{P} , and a delay sequence d for \mathcal{P} satisfying a given energy constraint E from initial level w_0 , we denote by $\mathcal{E}_{\mathcal{P},d}^E(w_0)$ the set of possible final energy levels. It may be seen that $\mathcal{E}_{\mathcal{P},d}^E(w_0)$ is a closed subset of E .

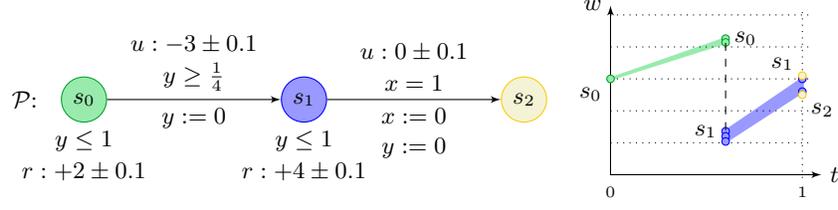


Fig. 4: An energy timed path \mathcal{P} with uncertainty, and a representation of the runs corresponding to the delay sequence $(0.6, 0.4)$ with initial energy level 3.

Example 5. Figure 4 is the energy timed path \mathcal{P} of Fig. 2 extended with uncertainties of ± 0.1 on all rates and updates. The runs associated with path \mathcal{P} , delay sequence $d = (0.6, 0.4)$ and initial energy level $w_0 = 3$ satisfy the energy constraint $E = [0; 5]$. The set $\mathcal{E}_{\mathcal{P}, d}^E(w_0)$ then is $[2.5; 3.1]$. \triangleleft

Now let $\mathcal{A} = (S, T, P)$ be an SETAu and let E be an energy constraint. A (memoryless⁶) strategy σ returns for any macro-configuration (s, w) ($s \in S$ and $w \in E$) a pair (t, d) , where $t = (s, s')$ is a successor edge in T and $d \in \mathbb{R}_{\geq 0}^n$ is a delay sequence for the corresponding energy timed path, i.e. $n = |P(t)|$. A (finite or infinite) execution of $(\rho^i)_i$ writing $\rho^i = (\ell_j^i, x_j^i, w_j^i)_{0 \leq j \leq 2n_i}$, is an outcome of σ if the following conditions hold:

- s_0^i and $s_{2n_i}^i$ are macro-states of \mathcal{A} , and ρ^i is a possible outcome of $P(s_0^i, s_{2n_i}^i)$ for d where $\sigma(s_0^i, w_0^i) = ((s_0^i, s_{2n_i}^i), d)$;
- $s_0^{i+1} = s_{2n_i}^i$ and $w_0^{i+1} = w_{2n_i}^i$.

Now we may formulate the infinite-run problem in the setting of uncertainty: for a SETAu \mathcal{A} , an energy constraint $E \in \mathcal{I}(\mathbb{Q})$, and a macro-state s_0 and an initial energy level w_0 the *energy-constrained infinite-run problem* is to decide the existence of a strategy σ for \mathcal{A} such that all runs $(\rho^i)_i$ that are outcome of σ starting from configuration (s_0, w_0) satisfy E ?

Ternary energy relations. Let $\mathcal{P} = (\{s_i \mid 0 \leq i \leq n\}, \{s_0\}, X, I, r, T, \epsilon, \Delta)$ be an ETPu and let $E \in \mathcal{I}(\mathbb{Q})$ be an energy constraint. The ternary energy relation $\mathcal{U}_{\mathcal{P}}^E \subseteq E \times E \times E$ relates all triples (w_0, a, b) for which there is a strategy σ such that any outcome of ρ from $(s_0, \mathbf{0}, w_0)$ satisfies E and ends in a configuration $(s_n, \mathbf{0}, w_1)$ where $w_1 \in [a; b]$. This relation can be characterized by the following first-order formula:

$$\mathcal{U}_{\mathcal{P}}^E(w_0, a, b) \iff \exists (d_i)_{0 \leq i < n}. \Phi_{\text{timing}} \wedge \Phi_{\text{energy}} \wedge w_0 + \sum_{k=0}^{n-1} (r(s_k) \cdot d_k + u_k) + \sum_{k=0}^{n-1} ([-\epsilon(s_k); \epsilon(s_k)] \cdot d_k + [-\Delta(t_k); \Delta(t_k)]) \subseteq [a; b]$$

⁶ For the infinite-run problem, it can be shown that memoryless strategies suffice.

where Φ_{energy}^i encodes the energy constraints as the inclusion of the interval of reachable energy levels in the energy constraint (in the same way as we do on the second line of the formula). Interval inclusion can then be expressed as constraints on the bounds of the intervals. It is clear that \mathcal{U}_P^E is a closed, convex subset of $E \times E \times E$ and can be described as a finite conjunction of linear constraints over w_0, a and b using quantifier elimination.

Example 6. We illustrate the above translation on the ETPu of Fig. 4. For energy constraint $[0; 5]$, the energy relation can be written as:

$$\begin{aligned} \mathcal{U}_P^E(w_0, a, b) \iff & \exists d_0, d_1. d_0 \in [0.25; 1] \wedge d_1 \in [0; 1] \wedge d_0 + d_1 = 1 \wedge w_0 \in [0; 5] \wedge \\ & w_0 + [1.9; 2.1] \cdot d_0 \subseteq [0; 5] \wedge \\ & w_0 + [1.9; 2.1] \cdot d_0 + [-3.1; -2.9] \subseteq [0; 5] \wedge \\ & w_0 + [1.9; 2.1] \cdot d_0 + [-3.1; -2.9] + [3.9; 4.1] \cdot d_1 \subseteq [0; 5] \wedge \\ & w_0 + [1.9; 2.1] \cdot d_0 + [-3.1; -2.9] + [3.9; 4.1] \cdot d_1 + [-0.1; 0.1] \subseteq [a; b] \subseteq [0; 5] \end{aligned}$$

Applying quantifier elimination, we end up with:

$$\begin{aligned} \mathcal{U}_P^E(w_0, a, b) \iff & 0 \leq a \leq b \leq 5 \wedge b \geq a + 0.6 \wedge a - 0.2 \leq w_0 \leq b + 0.7 \wedge \\ & (4.87 + 1.9 \cdot a)/3.9 \leq w_0 \leq (7.27 + 2.1 \cdot b)/4.1 \end{aligned}$$

We can use this relation in order to compute the set of initial energy levels from which there is a strategy to end up in $[2.5; 3.1]$ (which was the set of possible final energy levels in the example of Fig. 4). We get $w_0 \in [37/15; 689/205]$, which is (under-)approximately $w_0 \in [2.467; 3.360]$. \triangleleft

Algorithm for SETAu. Let $\mathcal{A} = (S, T, P)$ be a SETAu and let $E \in \mathcal{I}(\mathbb{Q})$ be an energy constraint. Let $\mathcal{W} \subseteq S \times E$ be the maximal set of configurations satisfying the following:

$$\begin{aligned} (s, w) \in \mathcal{W} \Rightarrow & \exists t = (s, s') \in T. \exists a, b \in E. \\ & \mathcal{U}_{P(t)}^E(w, a, b) \wedge \forall w' \in [a; b]. (s', w') \in \mathcal{W} \end{aligned} \quad (2)$$

Now \mathcal{W} is easily shown to characterize the set of configurations (s, w) that satisfy the energy-constrained infinite-run problem. Unfortunately this characterization does not readily provide an algorithm. We thus make the following restriction and show that it leads to decidability of the energy-constrained infinite-run problem:

(R) in any of the ETPu $P(t)$ of \mathcal{A} , on at least one of its transitions, some clock x is compared with a positive lower bound. Thus, there is an (overall minimal) positive time-duration D to complete any $P(t)$ of \mathcal{A} .

Theorem 7. *The energy-constrained infinite-run problem is decidable for SETAu satisfying (R).*

It is worth noticing that we do **not** assume flatness of the model for proving the above theorem. Instead, the minimal-delay assumption **(R)** has to be made: it entails that any stable set is made of intervals whose size is bounded below, which provides an upper bound on the number of such intervals. We can then rewrite the right-hand-side expression of (2) as:

$$\bigwedge_{s \in S} \bigwedge_{1 \leq j \leq N} [a_{s,j}; b_{s,j}] \subseteq E \wedge w_0 \in \bigvee_{1 \leq j \leq N} [a_{s_0,j}; b_{s_0,j}] \wedge \forall w \in [a_{s,j}; b_{s,j}].$$

$$\bigvee_{(s,s') \in T} [\exists a, b \in E. \mathcal{U}_{\mathcal{P}(s,s')}^E(w, a, b) \wedge \bigvee_{1 \leq k \leq N} ([a; b] \subseteq [a_{s',k}; b_{s',k}])] \quad (3)$$

Example 7. We pursue on Example 6. If ETPu \mathcal{P} is iterated (as on the loop on state s_2 of Fig. 3, but now with uncertainty), the set \mathcal{W} (there is a single macro-state) can be captured with a single interval $[a, b]$. We characterize the set of energy levels from which the path \mathcal{P} can be iterated infinitely often while satisfying the energy constraint $E = [0; 5]$ using equation (3), as follows:

$$0 \leq a \leq b \leq 5 \wedge \forall w_0 \in [a; b]. \mathcal{U}_{\mathcal{P}}^E(w_0, a, b).$$

We end up with $2.435 \leq a \wedge b \leq 3.635 \wedge b \geq a + 0.6$, so that the largest interval is $[2.435; 3.635]$ (which can be compared to the maximal fixpoint $[2; 4]$ that we obtained in Example 4 for the same cycle without uncertainty). \triangleleft

As in the setting without uncertainties, we can also synthesize an (optimal) upper-bound for the energy constraint:

Theorem 8. *Let $\mathcal{A} = (S, T, P)$ be a depth-1 flat SETAu satisfying **(R)**. Let $L \in \mathbb{Q}$ be an energy lower bound, and let (s_0, w_0) be an initial macro-configuration. Then the existence of an upper energy bound U , such that the energy-constrained infinite-run problem is satisfied for the energy constraint $[L; U]$ is decidable. Furthermore, one can compute the least upper bound, if there is one.*

4 Case Study

Modelling the Oil Pump System. In this section we describe the characteristics of each component of the HYDAC case, which we then model as a SETA.

The Machine. The oil consumption of the machine is cyclic. One cycle of consumptions, as given by HYDAC, consists of 10 periods of consumption, each having a duration of two seconds, as depicted in Figure 1b. Each period is described by a rate of consumption m_r (expressed in litres per second). The consumption rate is subject to noise: if the mean consumption for a period is c l/s (with $c \geq 0$) its actual value lies within $[\max(0, c - \epsilon); c + \epsilon]$, where ϵ is fixed to 0.1 l/s.

The Pump. The pump is either **On** or **Off**, and we assume it is initially **Off** at the beginning of a cycle. While it is **On**, it pumps oil into the accumulator with a rate $p_r = 2.2$ l/s. The pump is also subject to timing constraints, which prevent switching it on and off too often.

The Accumulator. The volume of oil within the accumulator will be modelled by means of an energy variable v . Its evolution is given by the differential inclusion $dv/dt - u \cdot p_r \in -[m_r + \epsilon; m_r - \epsilon]$ (or $-[m_r + \epsilon; 0]$ if $m_r - \epsilon < 0$), where $u \in \{0, 1\}$ is the state of the pump.

The controller must operate the pump (switch it on and off) to ensure the following requirements: (R1) the level of oil in the accumulator must always stay within the safety bounds $E = [V_{\min}; V_{\max}] = [4.9; 25.1]$ l (R2) the average level of oil in the accumulator is kept as low as possible.

By modelling the oil pump system as a SETA \mathcal{H} , the above control problem can be reduced to finding a deterministic schedule that results in a safe infinite run in \mathcal{H} . Furthermore, we are also interested in determining the minimal safety interval E , i.e., finding interval bounds that minimize $V_{\max} - V_{\min}$, while ensuring the existence of a valid controller for \mathcal{H} .

As a first step in the definition of \mathcal{H} , we build an ETP representing the behaviour of the machine, depicted on Fig. 5. In order to fully model the behaviour

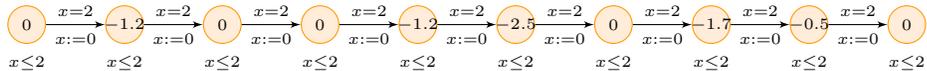


Fig. 5: The ETP representing the oil consumption of the machine.

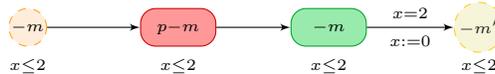


Fig. 6: An ETP for modelling the pump

of our oil-pump system, one would require the parallel composition of this ETP with another ETP representing the pump. The resulting ETA would not be a flat SETA, and is too large to be handled by our algorithm with uncertainty. Since it still provides interesting results, we develop this (incomplete) approach in the long version of this article [4].

Instead, we consider an alternative model of the pump, which only allows to switch it on and off once during each 2-second slot. This is modelled by inserting, between any two states of the model of Fig. 5, a copy of the ETP depicted on Fig. 6. In that ETP, the state with rate $p - m$ models the situation when the pump is on. Keeping the pump off for the whole slot can be achieved by spending delay zero in that state. We name $\mathcal{H}_1 = (M, T, P_1)$ the SETA made of a single macro-state equipped with a self-loop labelled with the ETP above.

In order to represent the timing constraints of the pump switches, we also consider a second SETA model $\mathcal{H}_2 = (M, T, P_2)$ where the pump can be operated only during every other time slot. This amounts to inserting the ETP of Fig. 6 only after the first, third, fifth, seventh and ninth states of the ETP of Fig. 5.

We also consider extensions of both models with uncertainty $\epsilon = 0.1$ l/s (changing any negative rate $-m$ into rate interval $[-m - \epsilon; -m + \epsilon]$, but changing rate 0 into $[-\epsilon; 0]$). We write $\mathcal{H}_1(\epsilon)$ and $\mathcal{H}_2(\epsilon)$ for the corresponding models.

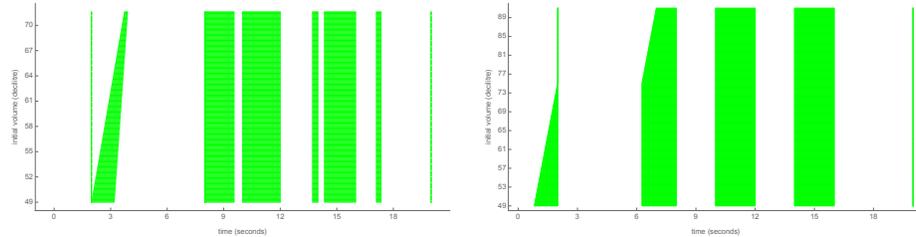


Fig. 7: Local strategies for $\mathcal{H}_1(\epsilon)$ (left) and $\mathcal{H}_2(\epsilon)$ (right) for a single cycle of the HYDAC system.

Synthesizing controllers. For each model, we synthesize minimal upper bounds U (within the interval $[V_{\min}; V_{\max}]$) that admit a solution to the energy-constrained infinite-run problem for the energy constraint $E = [V_{\min}; U]$. Then, we compute the greatest stable interval $[a; b] \subseteq [L; U]$ of the cycle witnessing the existence of an E -constrained infinite-run. This is done by following the methods described in Sections 2 and 3 where quantifier elimination is performed using Mjollnir [24].

Finally for each model we synthesise *optimal* strategies that, given an initial volume $w_0 \in [a, b]$ of the accumulator, return a sequence of pump activation times t_i^{on} and t_i^{off} to be performed during the cycle. This is performed in two steps: first we encode the set of safe *permissive strategies* as a quantifier-free first-order linear formula having as free variables w_0 , and the times t_i^{on} and t_i^{off} . The formula is obtained by relating w_0 , and the times t_i^{on} and t_i^{off} with the intervals $[L; U]$ and $[a; b]$ and delays d_i as prescribed by the energy relations presented in Sections 2 and 3. We use Mjollnir [24] to eliminate the existential quantifiers on the delays d_i . Then, given an energy value w_0 we determine an optimal safe strategy for it (i.e., some timing values when the pump is turned on and off) as the solution of the optimization problem that minimizes the average oil volume in the tank during one consumption cycle subject to the *permissive strategies* constraints. To this end, we use the function `FindMinimum` of Mathematica [28] to minimize the non-linear cost function expressing the average oil volume subject to the linear constraints obtained above. Fig. 7 shows the resulting strategies: there, each horizontal line at a given initial oil level indicates the delays (green intervals) where the pump will be running.

Table 1 summarizes the results obtained for our models. It gives the optimal volume constraints, the greatest stable intervals, and the values of the worst-case (over all initial oil levels in $[a; b]$) mean volume. It is worth noting that the models without uncertainty outperform the respective version with uncertainty. Moreover, the worst-case mean volume obtained both for $\mathcal{H}_1(\epsilon)$ and $\mathcal{H}_2(\epsilon)$ are significantly better than the optimal strategies synthesized both in [16] and [29].

The reason for this may be that (i) our models relax the latency requirement for the pump, (ii) the strategies of [16] are obtained using a discretization of the

| Controller | $[L; U]$ | $[a; b]$ | Mean vol. (l) |
|---------------------------|----------------------------|-------------|---------------|
| \mathcal{H}_1 | [4.9; 5.84] | [4.9; 5.84] | 5.43 |
| $\mathcal{H}_1(\epsilon)$ | [4.9; 7.16] | [5.1; 7.16] | 6.15 |
| \mathcal{H}_2 | [4.9; 7.9] | [4.9; 7.9] | 6.12 |
| $\mathcal{H}_2(\epsilon)$ | [4.9; 9.1] | [5.1; 9.1] | 7.24 |
| G1M1 [16] | [4.9; 25.1] ^(*) | [5.1; 9.4] | 8.2 |
| G2M1 [16] | [4.9; 25.1] ^(*) | [5.1; 8.3] | 7.95 |
| [29] | [4.9; 25.1] ^(*) | [5.2; 8.1] | 7.35 |

(*) Safety interval given by the HYDAC company.

Table 1: Characteristics of the synthesized strategies, compared with the strategies proposed in [16,29].

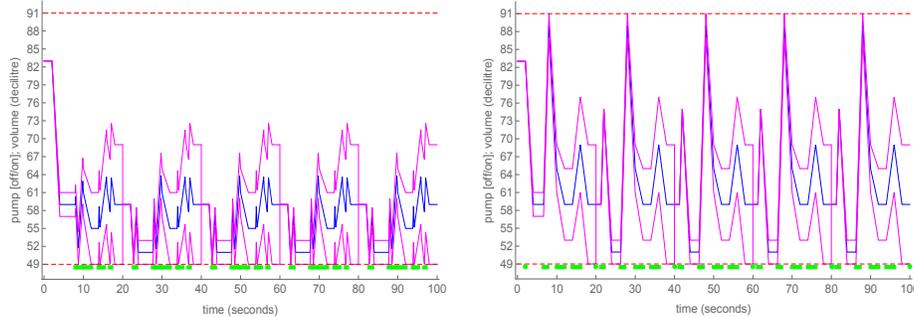


Fig. 8: Simulations of 5 consecutive machine cycles for $\mathcal{H}_1(\epsilon)$ and $\mathcal{H}_2(\epsilon)$.

dynamics within the system, and (iii) the strategies of [16] and [29] were allowed to activate the pump respectively two and three times during each cycle.

We proceed by comparing the performances of our strategies in terms of accumulated oil volume. Fig. 8 shows the result of simulating our strategies for a duration of 100 s. The plots illustrate in blue (resp. red) the dynamics of the mean (resp. min/max) oil level in the accumulator as well as the state of the pump. The initial volume used for the simulations is 8.3 l, as done in [16] for evaluating respectively the Bang-Bang controller, the Smart Controller developed by HYDAC, and the controllers G1M1 and G2M1 synthesized with UPPAAL-TIGA.

Table 2 presents, for each of the strategies, the resulting accumulated volume of oil, and the corresponding mean volume. There is a clear evidence that the strategies for \mathcal{H}_1 and \mathcal{H}_2 outperform all the other strategies. Clearly, this is due to the fact that they assume full precision in the rates, and allow for more switches of the pump. However, these results shall be read as what one could achieve by investing in more precise equipment. The results also confirm that both our strategies outperform those presented in [16]. In particular the strategy for $\mathcal{H}_1(\epsilon)$ provides an improvement of 55%, 46%, 20%, and 19% respectively

| Controller | Acc. vol. (l) | Mean vol. (l) | Controller | Acc. vol. (l) | Mean vol. (l) |
|---------------------------|---------------|---------------|------------|---------------|---------------|
| \mathcal{H}_1 | 1081.77 | 5.41 | Bang-Bang | 2689 | 13.45 |
| \mathcal{H}_2 | 1158.90 | 5.79 | HYDAC | 2232 | 11.60 |
| $\mathcal{H}_1(\epsilon)$ | 1200.21 | 6.00 | G1M1 | 1518 | 7.59 |
| $\mathcal{H}_2(\epsilon)$ | 1323.42 | 6.62 | G2M1 | 1489 | 7.44 |

Table 2: Performance based on simulations of 200 s starting with 8.3 l.

for the Bang-Bang controller, the Smart Controller of HYDAC, and the two strategies synthesized with UPPAAL-TIGA.

Tool chain⁷. Our results have been obtained using Mathematica and Mjollnir. Specifically, Mathematica was used to construct the formulas modelling the post-fixpoint of the energy functions, calling Mjollnir for performing quantifier elimination on them. The combination of both tools allowed us to solve one of our formulas with 27 variables in a compositional manner in ca. 20 ms, while Mjollnir alone would take more than 20 minutes. Mjollnir was preferred to Mathematica’s built-in support for quantifier elimination because the latter does not scale.

5 Conclusion

We developed a novel framework allowing for the synthesis of safe and optimal controllers, based on energy timed automata. Our approach consists in a translation to first-order linear arithmetic expressions representing our control problem, and solving these using quantifier elimination and simplification. We demonstrated the applicability and performance of our approach by revisiting the HYDAC case study and improving its best-known solutions.

Future works include extending our results to non-flat and non-segmented energy timed automata. However, existing results [22] indicate that we are close to the boundary of decidability. Another interesting continuation of this work would be to add UPPAAL STRATEGO [18,19] to our tool chain. This would allow to optimize the permissive strategies that we compute with quantifier elimination in the setting of probabilistic uncertainty, thus obtaining controllers that are optimal with respect to expected accumulated oil volume.

References

1. R. Alur, C. Courcoubetis, T. A. Henzinger, and P.-H. Ho. Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems. In R. L. Grossman, A. Nerode, A. P. Ravn, and H. Rischel, editors, *Hybrid Systems*, pages 209–229, Berlin, Heidelberg, 1993. Springer Berlin Heidelberg.

⁷ More details on our scripts are available at <http://people.cs.aau.dk/~giovbacci/tools.html>, together with the models we used for our examples and case study.

2. R. Alur and D. L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126(2):183–235, Apr. 1994.
3. R. Alur, S. La Torre, and G. J. Pappas. Optimal paths in weighted timed automata. In M. D. Di Benedetto and A. L. Sangiovani-Vincentelli, editors, *Proceedings of the 4th International Workshop on Hybrid Systems: Computation and Control (HSCC'01)*, volume 2034 of *Lecture Notes in Computer Science*, pages 49–62. Springer-Verlag, Mar. 2001.
4. G. Bacci, P. Bouyer, U. Fahrenberg, K. G. Larsen, N. Markey, and P.-A. Reynier. Optimal and Robust Controller Synthesis: Using Energy Timed Automata with Uncertainty, 2018. arXiv:1805.00847 [cs.FL].
5. G. Behrmann, A. Cougnard, A. David, E. Fleury, K. G. Larsen, and D. Lime. UPPAAL-Tiga: Time for playing games! In W. Damm and H. Hermanns, editors, *Computer Aided Verification, 19th International Conference, CAV 2007, Berlin, Germany, July 3-7, 2007, Proceedings*, volume 4590 of *Lecture Notes in Computer Science*, pages 121–125. Springer, 2007.
6. G. Behrmann, A. Fehnker, T. Hune, K. G. Larsen, P. Pettersson, J. Romijn, and F. Vaandrager. Minimum-cost reachability for priced timed automata. In M. D. Di Benedetto and A. L. Sangiovani-Vincentelli, editors, *Proceedings of the 4th International Workshop on Hybrid Systems: Computation and Control (HSCC'01)*, volume 2034 of *Lecture Notes in Computer Science*, pages 147–161. Springer-Verlag, Mar. 2001.
7. A. Bemporad, G. Ferrari-Trecate, and M. Morari. Observability and controllability of piecewise affine and hybrid systems. *IEEE Transactions on Automatic Control*, 45(10):1864–1876, 2000.
8. M. Bisgaard, D. Gerhardt, H. Hermanns, J. Krcál, G. Nies, and M. Stenger. Battery-aware scheduling in low orbit: The GomX-3 case. In J. S. Fitzgerald, C. L. Heitmeyer, S. Gnesi, and A. Philippou, editors, *FM 2016: Formal Methods - 21st International Symposium, Limassol, Cyprus, November 9-11, 2016, Proceedings*, volume 9995 of *Lecture Notes in Computer Science*, pages 559–576, 2016.
9. V. D. Blondel, O. Bournez, P. Koiran, and J. N. Tsitsiklis. The stability of saturated linear dynamical systems is undecidable. *Journal of Computer and System Sciences*, 62(3):442–462, 2001.
10. V. D. Blondel and J. N. Tsitsiklis. Complexity of stability and controllability of elementary hybrid systems. *Automatica*, 35(3):479–489, 1999.
11. P. Bouyer, U. Fahrenberg, K. G. Larsen, and N. Markey. Timed automata with observers under energy constraints. In K. H. Johansson and W. Yi, editors, *Proceedings of the 13th International Workshop on Hybrid Systems: Computation and Control (HSCC'10)*, pages 61–70. ACM Press, Apr. 2010.
12. P. Bouyer, U. Fahrenberg, K. G. Larsen, N. Markey, and J. Srba. Infinite runs in weighted timed automata with energy constraints. In F. Cassez and C. Jard, editors, *Proceedings of the 6th International Conferences on Formal Modelling and Analysis of Timed Systems (FORMATS'08)*, volume 5215 of *Lecture Notes in Computer Science*, pages 33–47. Springer-Verlag, Sept. 2008.
13. P. Bouyer, K. G. Larsen, and N. Markey. Lower-bound constrained runs in weighted timed automata. *Performance Evaluation*, 73:91–109, Mar. 2014.
14. M. Bozga, R. Iosif, and Y. Lakhnech. Flat parametric counter automata. In M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener, editors, *Proceedings of the 33rd International Colloquium on Automata, Languages and Programming (ICALP'06) – Part II*, volume 4052 of *Lecture Notes in Computer Science*, pages 577–588. Springer-Verlag, July 2006.

- 18 G. Bacci, P. Bouyer, U. Fahrenberg, K.G. Larsen, N. Markey, P.A. Reynier
15. F. Cassez, A. David, E. Fleury, K. G. Larsen, and D. Lime. Efficient on-the-fly algorithms for the analysis of timed games. In M. Abadi and L. de Alfaro, editors, *CONCUR 2005 - Concurrency Theory, 16th International Conference, CONCUR 2005, San Francisco, CA, USA, August 23-26, 2005, Proceedings*, volume 3653 of *Lecture Notes in Computer Science*, pages 66–80. Springer, 2005.
 16. F. Cassez, J. J. Jensen, K. G. Larsen, J.-F. Raskin, and P.-A. Reynier. Automatic synthesis of robust and optimal controllers – an industrial case study. In R. Majumdar and P. Tabuada, editors, *Proceedings of the 12th International Workshop on Hybrid Systems: Computation and Control (HSCC'09)*, volume 5469 of *Lecture Notes in Computer Science*, pages 90–104. Springer-Verlag, Apr. 2009.
 17. H. Comon and Y. Jurski. Multiple counters automata, safety analysis, and Presburger arithmetic. In A. J. Hu and M. Y. Vardi, editors, *Proceedings of the 10th International Conference on Computer Aided Verification (CAV'98)*, volume 1427 of *Lecture Notes in Computer Science*, pages 268–279. Springer-Verlag, June-July 1998.
 18. A. David, P. G. Jensen, K. G. Larsen, A. Legay, D. Lime, M. G. Sørensen, and J. H. Taankvist. On time with minimal expected cost! In F. Cassez and J. Raskin, editors, *Automated Technology for Verification and Analysis - 12th International Symposium, ATVA 2014, Sydney, NSW, Australia, November 3-7, 2014, Proceedings*, volume 8837 of *Lecture Notes in Computer Science*, pages 129–145. Springer, 2014.
 19. A. David, P. G. Jensen, K. G. Larsen, M. Mikucionis, and J. H. Taankvist. Up-paal Stratego. In C. Baier and C. Tinelli, editors, *Tools and Algorithms for the Construction and Analysis of Systems - 21st International Conference, TACAS 2015, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2015, London, UK, April 11-18, 2015. Proceedings*, volume 9035 of *Lecture Notes in Computer Science*, pages 206–211. Springer, 2015.
 20. G. Frehse. Phaver: algorithmic verification of hybrid systems past hytech. *STTT*, 10(3):263–279, 2008.
 21. S. Jha, S. A. Seshia, and A. Tiwari. Synthesis of optimal switching logic for hybrid systems. In S. Chakraborty, A. Jerraya, S. K. Baruah, and S. Fischmeister, editors, *Proceedings of the 11th International Conference on Embedded Software, EMSOFT 2011, part of the Seventh Embedded Systems Week, ESWeek 2011, Taipei, Taiwan, October 9-14, 2011*, pages 107–116. ACM, 2011.
 22. N. Markey. *Verification of Embedded Systems – Algorithms and Complexity*. Mémoire d’habilitation, École Normale Supérieure de Cachan, France, Apr. 2011.
 23. S. Miremadi, Z. Fei, K. Åkesson, and B. Lennartson. Symbolic supervisory control of timed discrete event systems. *IEEE Trans. Contr. Sys. Techn.*, 23(2):584–597, 2015.
 24. D. Monniaux. Quantifier elimination by lazy model enumeration. In T. Touili, B. Cook, and P. B. Jackson, editors, *Computer Aided Verification, 22nd International Conference, CAV 2010, Edinburgh, UK, July 15-19, 2010. Proceedings*, volume 6174 of *Lecture Notes in Computer Science*, pages 585–599. Springer, 2010.
 25. A. Phan, M. R. Hansen, and J. Madsen. EHRA: Specification and analysis of energy-harvesting wireless sensor networks. In S. Iida, J. Meseguer, and K. Ogata, editors, *Specification, Algebra, and Software - Essays Dedicated to Kokichi Futatsugi*, volume 8373 of *Lecture Notes in Computer Science*, pages 520–540. Springer, 2014.
 26. Quasimodo. Quantitative system properties in model-driven design of embedded systems. <http://www.quasimodo.aau.dk/>.
 27. G. von Bochmann, M. Hilscher, S. Linker, and E. Olderog. Synthesizing and verifying controllers for multi-lane traffic maneuvers. *Formal Asp. Comput.*, 29(4):583–600, 2017.

28. Wolfram Research, Inc. Mathematica, Version 11.2. Champaign, IL, 2017.
29. H. Zhao, N. Zhan, D. Kapur, and K. G. Larsen. A “hybrid” approach for synthesizing optimal controllers of hybrid systems: A case study of the oil pump industrial example. In D. Giannakopoulou and D. Méry, editors, *FM 2012: Formal Methods - 18th International Symposium, Paris, France, August 27-31, 2012. Proceedings*, volume 7436 of *Lecture Notes in Computer Science*, pages 471–485. Springer, 2012.