https://doi.org/10.1007/s00165-020-00521-4
BCS © 2020
Formal Aspects of Computing (2021) 33: 3–25

Formal Aspects
of Computing

Optimal and robust controller synthesis[1] using energy timed automata with uncertainty

Giovanni Bacci[1], Patricia Bouyer[2], Uli Fahrenberg[3], Kim G. Larsen[1], Nicolas Markey[4], Pierre-Alain Reynier[5]

[1]Department of Computer Science, Aalborg University, Selma Lagerlöfs Vej 300, 9220 Aalborg Øst, Denmark,
[2]LSV, CNRS & ENS Cachan, Université Paris-Saclay, Cachan, France
[3]École Polytechnique, Palaiseau, France
[4]IRISA, CNRS & INRIA, Univ. Rennes, Rennes, France
[5]CNRS, LIS, Aix Marseille Univ., Université de Toulon, 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.

Keywords: Energy timed automata; Controller synthesis; Quantifier elimination

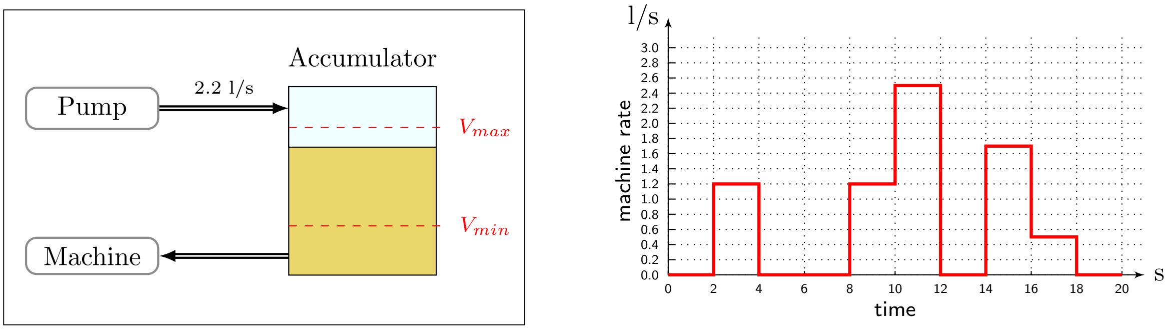
1. Introduction

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. [BGH+16, vBHLO17, PHM14], 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 [BFL+08] extend timed automata [AD94] with a continuous *energy* variable that evolves with varying rates and discrete updates during the behaviour of the model. Addressing an open problem from [BFL+08], 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.

[1] Work supported by ERC projects Lasso and EQualIS and by French ANR project TickTac.

Correspondence to: Giovanni Bacci, e-mail: giovbacci@cs.aau.dk



(a) System Components

(b) Cycle of the Machine

Fig. 1. Overview of the HYDAC system

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 are based on reductions to quantifier elimination (QE) for linear real arithmetic, for which we combine Mathematica [Wol] and Mjollnir [Mon10] 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 [Qua]. It consists of 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 [CJL$^+$09] as a timed game using the tool Uppaal- Tiga [CDF$^+$05, BCD$^+$07] for synthesis. Discretization of oil-level (and time) was used to make synthesis feasible. Besides limiting the opportunity of optimality, the discretization also necessitated posterior verification using PHAVER [Fre08] to rule out possible resulting incorrectness. Also, identification of safety and minimal oil levels were done by manual and laborious search. In [MFÅL15] the timed game models of [CJL$^+$09] (rephrased as Timed Discrete Event Systems) are reused, but BDDs are applied for compact representation of the discrete oil-levels and time-points encountered during synthesis. [JST11] provides a framework for learning optimal switching strategies by a combination of off-the-shelf numerical optimization and generalization by learning. The HYDAC case is one of the considered cases. The method offers no absolute guarantees of hard constraints on energy-level, but rather attempts to enforce these through the use of high penalties. [ZZKL12] focuses exclusively on the HYDAC case using a direct encoding of the safety- and optimality-constraints as QE problems. This gives—like in our case—absolute guarantees. However, we are additionally offering a complete and decidable framework based on energy timed automata, which extends to several other systems. Moreover, the controllers we obtain perform significantly better than those of [CJL$^+$09] and [ZZKL12] (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) [BFTM00] and hybrid systems [ACHH93]. In particular, the energy-constrained infinite-run problem is related to the so called *stability problem* for PWAs. Blondel and Tsitsiklis [BT99] 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 [BBKT01].

The current paper is an extended and improved version of [BBF$^+$18], containing detailed proofs of the above mentioned results. Furthermore, we elaborate on the HYDAC case synthesizing strategies for a more accurate (non-flat) model of the oil pump system.

2. Energy timed automata

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 $v \colon X \to \mathbb{R}_{\geq 0}$ 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 $v[R \to 0]$ 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 2.1 An *energy timed automaton* (*ETA* for short; a.k.a. *priced* or *weighted timed automaton* [ALP01, BFH$^+$01]) 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, $I \colon S \to C(X)$ assigns invariants to states, $r \colon S \to \mathbb{Q}$ 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; the first and last components of those 5-tuples are states of the automaton, while g_i represents the clock constraint to be satisfied for the transition to be available, u_i is the amount of energy gained or consumed along the transition, and z_i indicates which clocks are reset when the transition takes place. The runs of an (energy) timed automaton $\mathcal{A}$ are the sequences of configurations visited by the automaton when alternatively taking a transition of the automaton, and letting time elapse [AD94]. Formally, given a sequence $\tau = (t_i)_{0 \leq i < n}$ with $t_i = (s_i, g_i, u_i, z_i, s_{i+1})$, 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 exists a sequence of delays $(d_i)_{0 \leq i < n}$ 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 $v_{2j+2} = v_{2j+1}[z_j \to 0]$;
- 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 2.1 Figure 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 the energy constraint $[0;\ 5]$.

Definition 2.2 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*), 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.

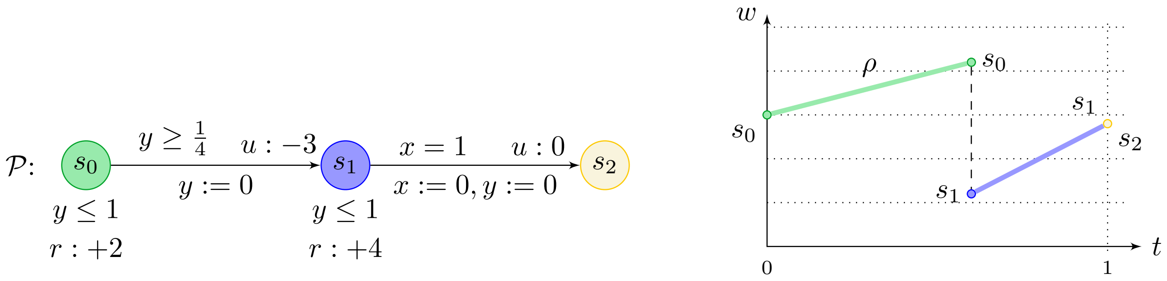


Fig. 2. An energy timed path $\mathcal{P}$, and a run ρ of $\mathcal{P}$ with initial energy level 3.

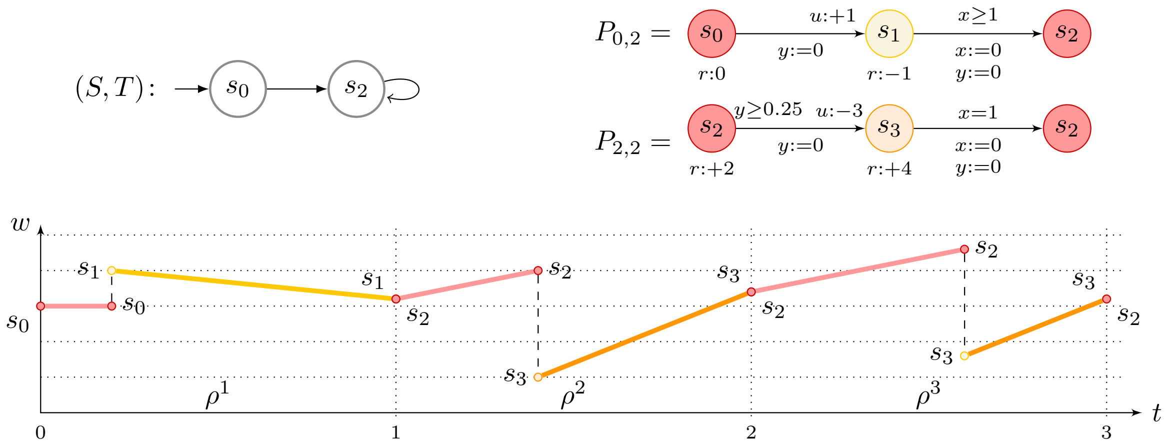


Fig. 3. A SETA $\mathcal{A} = (S, T, P)$ with implicit global invariant $y \leq 1$; omitted discrete updates are assumed to be zero. The map P associates with each $(s_i, s_j) \in T$ the ETP $P_{i,j}$. The infinite sequence $\rho^1 \cdot (\rho^2 \cdot \rho^3)^\omega$ is an infinite execution of $\mathcal{A}$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

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 [CJ98, BIL06]). 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 $\rho^i = (\ell^i_j, v^i_j, w^i_j)_{0 \leq j \leq 2n_i}$, it holds:

- ℓ^i_0 and $\ell^i_{2n_i}$ are macro-states of $\mathcal{A}$, and ρ^i is a run of the ETP $P(\ell^i_0, \ell^i_{2n_i})$;
- $\ell^{i+1}_0 = \ell^i_{2n_i}$ and $w^{i+1}_0 = w^i_{2n_i}$.

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.

Remark 2.1 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 2.2 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 $[1;\ 4.6]$).

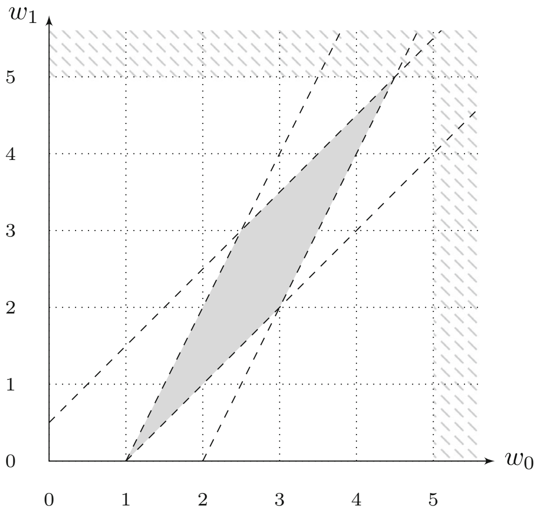


Fig. 4. Energy relation for $\mathcal{P}$ with $E = [0;\ 5]$.

In this paper, we consider the following *energy-constrained infinite-run problem* [BFL$^+$08]: 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 [Mar11]. In this paper, we prove:

Theorem 2.1 The energy-constrained infinite-run problem is decidable for flat SETA.

Theorem 2.2 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.

The rest of this section is devoted to the proof of the above two decidability results. In Sects. 2.1, 2.2, and 2.3 we first introduce some technical tools that we will use in Sect. 2.4 for developing the algorithms witnessing decidability of our problems.

2.1. 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* $\mathcal{R}^E_{\mathcal{P}} \subseteq E \times E$ 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:

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

where Φ_{timing} encodes all the timing constraints that the sequence $(d_i)_{0 \leq i < n}$ has to fulfill, while Φ_{energy} encodes the energy constraints. More precisely:

- timing constraints are obtained by computing the clock valuations in each state of the execution, and expressing that those values must satisfy the corresponding invariants and guards. The value of a clock in a state is the sum of the delays d_j since the last reset of that clock along the ETP.
- energy constraints are obtained by expressing the value of the energy level in each state as the sum of the initial energy level, the energy $r(s_i) \cdot d_i$ gained or consumed in each intermediary state, and the updates u_i of the transitions that have been traversed. All those values are constrained to lie in E.

Fourier-Motzkin elimination is a classical technique for removing existentially-quantified formulas from conjunctions of linear expressions such as the one defining $\mathcal{R}^R_{\mathcal{P}}$. Basically, there exists a value for variable x satisfying a conjunction $\mathcal{E}(x, Y)$ of linear constraints defined as

$$\bigwedge_{1\leq j\leq m} x \leq \phi_j(Y) \wedge \bigwedge_{1\leq k\leq n} x \geq \psi_k(Y)$$

if, and only if, for any $1 \leq j \leq m$ and any $1 \leq k \leq n$, it holds $\psi_k(Y) \leq \phi_j(Y)$. In other terms,

$$\{Y \mid \exists x.\ \mathcal{E}(x, Y)\} = \{Y \mid \forall j \in [1;\ m].\ \forall k \in [1;\ n].\ \psi_k(Y) \leq \phi_j(Y)\}.$$

By noticing that $\exists x.\ (\mathcal{E}(x, Y) \vee \mathcal{E}'(x, Y))$ is equivalent to $\exists x.\ (\mathcal{E}(x, Y) \vee \exists x.\ \mathcal{E}'(x, Y))$, we can eliminate existentially-quantified variables for any boolean combination of linear constraints. By duality, universally-quantified variables can be eliminated with the same procedure.

It follows that $\mathcal{R}^E_{\mathcal{P}}$ is a closed, convex subset of $E \times E$, and can be described as a conjunction of a finite set of linear constraints over w_0 and w_1 (with non-strict inequalities).

Example 2.3 We illustrate this computation on the ETP of Fig. 2. For energy constraint [0; 5], the energy relation can be written (after removing redundant constraints) as

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

Applying quantifier elimination, the above simplifies to

$$\mathcal{R}^E_{\mathcal{P}}(w_0, w_1) \iff (w_1 + 2 \leq 2w_0 \leq w_1 + 4) \wedge (w_1 - 0.5 \leq w_0 \leq w_1 + 1).$$

The corresponding polyhedron is depicted in Fig. 4.

2.2. 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 $\mathcal{R}^E_{\mathcal{P}}$, we associate an *energy function* (also denoted with $\mathcal{R}^E_{\mathcal{P}}$, or simply $\mathcal{R}$, as long as no ambiguity may arise), defined for any closed subinterval $I \in \mathcal{I}(E)$ as

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

Symmetrically, we let

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

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 monotonic: for any two intervals I and J in $\mathcal{I}(E)$ such that $I \subseteq J$, it holds that $\mathcal{R}(I) \subseteq \mathcal{R}(J)$ and $\mathcal{R}^{-1}(I) \subseteq \mathcal{R}^{-1}(J)$.

The energy functions $\mathcal{R}$ and $\mathcal{R}^{-1}$ also satisfy the following continuity properties:

Lemma 2.1 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 $\mathcal{R}^{-1}(\bigcap_{j\in\mathbb{N}} I_j) = \bigcap_{j\in\mathbb{N}} \mathcal{R}^{-1}(I_j)$.

Proof For any $i \in \mathbb{N}$, we have $I_i \supseteq \bigcap_{j\in\mathbb{N}} I_j$. By monotonicity of $\mathcal{R}^{-1}$, we get $\mathcal{R}^{-1}(I_i) \supseteq \mathcal{R}^{-1}(\bigcap_{j\in\mathbb{N}} I_j)$. It follows that $\bigcap_{i\in\mathbb{N}} \mathcal{R}^{-1}(I_i) \supseteq \mathcal{R}^{-1}(\bigcap_{j\in\mathbb{N}} I_j)$.

Now, let $w_0 \in \bigcap_{j\in\mathbb{N}} \mathcal{R}^{-1}(I_j)$. Then for all $i \in \mathbb{N}$, there exists w_1^i such that $\mathcal{R}(w_0, w_1^i)$. It follows that for any $i \in \mathbb{N}$, $\mathcal{R}(\{w_0\}) \cap I_i$ is a non-empty interval of $\mathcal{I}(E)$. Applying Cantor's Intersection Theorem, we get that $\bigcap_{i\in\mathbb{N}} \mathcal{R}(\{w_0\}) \cap I_i$ is a non-empty interval of $\mathcal{I}(E)$. This intersection can be rewritten as $\mathcal{R}(\{w_0\}) \cap \bigcap_{i\in\mathbb{N}} I_i$; hence there exists $w_1 \in \bigcap_{i\in\mathbb{N}} I_i$ such that $\mathcal{R}(w_0, w_1)$, which proves that $w_0 \in \mathcal{R}^{-1}(\bigcap_{i\in\mathbb{N}} I_i)$. □

2.3. 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 $\mathcal{R}^E_{\mathcal{P}_k} \circ \cdots \circ \mathcal{R}^E_{\mathcal{P}_1}$; 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 $(\mathcal{R}^E_{\mathcal{P}})^k$ for the composition of k copies of the same relations $\mathcal{R}^E_{\mathcal{P}}$.

Now, using Lemma 2.1, we get 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 $\mathcal{R}^{-1}$ in the sense that $S \subseteq \mathcal{R}^{-1}(S)$).

In the end, if $\mathcal{R}$ is the energy relation of a cycle $\mathcal{C}$ in the 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.

Now if $\mathcal{R}$ is the energy relation for a cycle $\mathcal{C}$, described as the conjunction $\phi_{\mathcal{C}}$ of a finite set of linear constraints, we can characterize those intervals $[a, b] \subseteq E$ that constitute a post-fixpoint for $\mathcal{R}^{-1}$ by the following first-order formula:

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

Applying quantifier eliminination (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 maximizing $b - a$.

Example 2.4 We again consider the 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: as explained in Example 2.3, the energy relation for this cycle can be written as

$$\mathcal{R}^E_{\mathcal{C}}(w_0, w_1) \iff (w_1 + 2 \leq 2w_0 \leq w_1 + 4) \wedge (w_1 - 0.5 \leq w_0 \leq w_1 + 1).$$

Our first-order formula for the fixpoint then reads as follows:

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

Applying quantifier elimination, we end up with

$$2 \leq a \leq b \leq 4.$$

This characterizes all post-fixpoints; the greatest fixpoint then obviously is [2; 4].

Now, the energy relation for the path $\mathcal{P}$ from s_1 to s_2 is

$$\begin{aligned}\mathcal{R}^E_{\mathcal{P}}(w_0, w_1) \iff \exists\, d_0, d_1.\ \ & 0 \leq d_0 \leq 1 \wedge 0 \leq d_1 \leq 1 \wedge d_0 + d_1 \geq 1 \wedge \\ & 0 \leq w_0 \leq 5 \wedge 0 \leq w_0 + 1 \leq 5 \wedge \\ & w_1 = w_1 + 1 - d_1 \wedge 0 \leq w_1 \leq 5\end{aligned}$$

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 the following constraint:

$$\exists\, w_1.\ (0 \leq w_0 \leq 4 \wedge w_0 \leq w_1 \leq w_0 + 1) \wedge (2 \leq w_1 \leq 4),$$

which, after quantifier elimination, reduces to $1 \leq w_0 \leq 4$.

2.4. Algorithm for flat segmented energy timed automata

Following Example 2.4, 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 2.3 Let $\mathcal{R}$ be the energy relation of an ETP $\mathcal{P}$ with energy constraint E, and let I be a closed sub-interval of E. Then either $I \cap \nu\mathcal{R}^{-1} \neq \emptyset$ or $\mathcal{R}^n(I) = \emptyset$ for some n.

Proof We assume that $I \cap \nu\mathcal{R}^{-1} = \emptyset$, and prove that $\mathcal{R}^n(I) = \emptyset$ for some n. We have:

$$\begin{aligned} I \cap \nu\mathcal{R}^{-1} &= I \cap \bigcap_{n\in\mathbb{N}} \left(\mathcal{R}^{-1}\right)^n(E) \\ &= I \cap \bigcap_{n\in\mathbb{N}} \left(\mathcal{R}^{n}\right)^{-1}(E) && (\text{by } \mathcal{R}^{-1} \circ \mathcal{R}^{-1} = (\mathcal{R} \circ \mathcal{R})^{-1}) \\ &= \bigcap_{n\in\mathbb{N}} \left(I \cap \left(\mathcal{R}^{n}\right)^{-1}(E)\right). \end{aligned}$$

Note that $\left(I \cap (\mathcal{R}^n)^{-1}(E)\right)_{n\in\mathbb{N}}$ is a decreasing sequence because $\left((\mathcal{R}^{-1})^n(E)\right)_{n\in\mathbb{N}}$ is. From our assumption that $I \cap \nu\mathcal{R}^{-1} = \emptyset$, we get that $\bigcap_{n\in\mathbb{N}} \left(I \cap (\mathcal{R}^n)^{-1}(E)\right) = \emptyset$. By Cantor's intersection theorem, it follows that $I \cap \left(\mathcal{R}^n\right)^{-1}(E) = \emptyset$ for some $n \in \mathbb{N}$.

Now, assume $\mathcal{R}^n(I) \neq \emptyset$, and pick $w_1 \in \mathcal{R}^n(I)$. Then for some $w_0 \in I$, we have $\mathcal{R}^n(w_0, w_1)$, so that also $w_0 \in (\mathcal{R}^n)^{-1}(E)$, so that $I \cap \left(\mathcal{R}^n\right)^{-1}(E) \neq \emptyset$. Hence $\mathcal{R}^n(I)$ must be empty. □

We will show that the energy-constrained infinite run problem is decidable for flat SETAs. For a SETA $\mathcal{A}$, an infinite run exists if and only if the underlying graph of $\mathcal{A}$ has a path $\mathcal{P} = (m_0, m_1)(m_1, m_2)\cdots(m_{k-1}, m_k)$ and cycle $\mathcal{C} = (m_k, m_{k+1})(m_{k+1}, m_{k+2})\cdots(m_{k+n-1}, m_{k+n})$ where $m_k = m_{k+n}$, such that

$$\nu\mathcal{R}_{\mathcal{C}}^{-1} \cap \mathcal{R}_{\mathcal{P}}(I_0) \neq \emptyset \tag{2}$$

where $\mathcal{R}_{\mathcal{C}} = \mathcal{R}_{P(m_{k+n-1}, m_{k+n})} \circ \cdots \circ \mathcal{R}_{P(m_k, m_{k+1})}$ and $\mathcal{R}_{\mathcal{P}} = \mathcal{R}_{P(m_k, m_{k-1})} \circ \cdots \circ \mathcal{R}_{P(m_1, m_0)}$. Intuitively, the interval $I = \mathcal{R}_{\mathcal{P}}(I_0)$ represents the energy interval I_0 propagated forward along the path $\mathcal{P}$ until reaching m_k. Moreover, if the SETA is flat, we have that the cycle $\mathcal{C}$ can be unambiguously represented by its initial state m_k—recall that flatness entails that any state belongs to at most one cycle.

The decision procedure traverses the underlying graph of $\mathcal{A}$, forward propagating an initial energy interval $I_0 \subseteq E$ for all reachable macro states m then looking for a simple cycle $\mathcal{C}$ starting from m such that $\nu\mathcal{R}_{\mathcal{C}}^{-1} \cap I \neq \emptyset$, where $I \subseteq E$ is the energy interval forward-propagated until reaching m.

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-state s that need to be further explored. The list W contains tasks of the form (m, I, flag) where the first component $m \in S$ is the current macro-state reached following some path $\mathcal{P}$ in $\mathcal{A}$, the second component $I \in \mathcal{I}(E)$ is the current energy interval, i.e., $I = \mathcal{R}_{\mathcal{P}}(I_0)$, and the third component $\mathit{flag} \in \{c, \bar{c}\}$ is a flag indicating whether the algorithm should consider m as the last element of the prefix path $\mathcal{P}$ and explore the cycle it belongs to ($\mathit{flag} = c$), or if it should proceed by exiting that cycle ($\mathit{flag} = \bar{c}$) further extending the prefix path.

The algorithm initialises the waiting list with the initial task (cf. line 1). The main **while** loop processes each task in the waiting list, as long as the list is not empty. It picks a task (m, I, flag) from W (line 3). If $\mathit{flag} = \bar{c}$, the exploration will continue from macro-state s m' adjacent to m by forward propagating the current energy interval I following the timed path $P(m, m')$ (cf. lines 6–7). Note that the choice of the arcs (m, m') ensures that m' does not belong to the same cycle as m, thus skipping the (unique) cycle containing m.

Otherwise, if $\mathit{flag} = c$, the exploration attempts to follow the simple cycle that contains m. If m does not belong to any cycle, the current task will be simply put back in the waiting list with the opposite flag (cf. line 23). In case m belongs to the simple cycle $\mathcal{C} = (m_1, m_2)\cdots(m_k, m_{k+1})$, the energy relation $\mathcal{R}_{\mathcal{C}}^{E}$ is used to check if, for the current energy interval, there exists an infinite run along the cycle $\mathcal{C}$. If such is not the case, the cycle will be iterated only finitely many times (cf. lines 17–21). This is done by inserting in W the current task with the flag set to $\bar{c}$—corresponding to zero executions of the cycle—then for each execution i of $\mathcal{C}$, the cycle is unfolded up to its j-th transition and the task $(m_{j+1}, \mathcal{R}_{\mathcal{P}_j}^{E}((\mathcal{R}_{\mathcal{C}}^{E})^i(I)), \bar{c})$ is added to the waiting list—corresponding to i executions of $\mathcal{C}$ followed by a tail $(m_1, m_2)\cdots(m_j, m_{j+1})$. Theorem 2.3 ensures termination of the **while** loop of lines 17–21.

```
Input: A flat SETA $\mathcal{A} = (S, T, P)$; initial state $m_0 \in S$; energy interval $I_0$
 1. $W \leftarrow \{(m_0, I_0, c)\}$                                          ◁ initialise the waiting list
 2. while $W \neq \emptyset$ do
 3.   pick $(m, I, flag) \in W$                                      ◁ pick an element from the waiting list
 4.   $W \leftarrow W \setminus (m, I, flag)$                         ◁ remove the element from the waiting list
 5.   if $flag = \bar{c}$ then                    ◁ 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.       $W \leftarrow W \cup \{(m', \mathcal{R}^E_{P(m,m')}(I), c)\}$          ◁ add this new task to the waiting list
 8.   else                                         ◁ 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 $\mathcal{R}_\mathcal{C} = \mathcal{R}_{P(m_k,m_{k+1})} \circ \cdots \circ \mathcal{R}_{P(m_1,m_2)}$          ◁ energy relation of the cycle
12.       if $I \cap \nu\mathcal{R}_\mathcal{C}^{-1} \neq \emptyset$ then          ◁ check if there is an infinite run via the cycle $\mathcal{C}$
13.         return tt
14.       else                                   ◁ the cycle can be executed only finitely many times
15.         $W \leftarrow W \cup \{(m, I, \bar{c})\}$                              ◁ add a new task to the waiting list
16.         $i \leftarrow 0$                                      ◁ initialise the number of cycle executions
17.         while $\mathcal{R}^i_\mathcal{C}(I) \neq \emptyset$ do                          ◁ while $i$-th energy relation is satified
18.           for $1 \leq j < k$ do
19.             let $\mathcal{R}_{\mathcal{P}_j} = \mathcal{R}_{P(m_j,m_{j+1})} \circ \cdots \circ \mathcal{R}_{P(m_1,m_2)}$                 ◁ unfold $\mathcal{C}$ up to $m_{j+1}$
20.             $W \leftarrow W \cup \{(m_{j+1}, \mathcal{R}_{\mathcal{P}_j}(\mathcal{R}^i_\mathcal{C}(I)), \bar{c})\}$            ◁ add a task to the waiting list
21.           $i \leftarrow i + 1$                              ◁ increment the number of cycle executions
22.     else                                                          ◁ $m$ doesn't belong to a cycle
23.       $W \leftarrow W \cup \{(m, I, \bar{c})\}$                                ◁ add a new task to the waiting list
24. return ff                                                          ◁ no infinite run could be found
```

Algorithm 1: Infinite Run

As for the correctness of the procedure, one can note that each task (m, I, flag) is used to represent a path $\mathcal{P}$ from m_0 to m having $I = \mathcal{R}_\mathcal{P}(I_0)$. In particular, when $\mathit{flag} = c$ the task indicates that the prefix $\mathcal{P}$ should be tested against condition (2) w.r.t. a possible cycle starting from m; whereas, when $\mathit{flag} = c$ the task indicates that the prefix $\mathcal{P}$ still has to be extended by at least one more step. Note that tasks having $\mathit{flag} = c$ are inserted in W only if a cycle having m was not already tested against condition (2) (cf. line 4 and 7); while tasks with $\mathit{flag} = \bar{c}$ are inserted in W only after having tested against condition (2) a cycle passing through such state (cf. lines 15, 20, and 23). The **for** loop at lines 6–7 ensures that all reachable states are eventually inserted in the waiting list; whereas the **while** loop at lines 17–21 ensures that all extensions of the current prefix obtained by appending to it a finite unfolding of a cycle are added to the waiting lists. Therefore, all possible prefixes are eventually tested against condition (2). It is worth noting that the flatness assumption for the SETA $\mathcal{A}$ ensures that all cycles are tested, because for each prefix $\mathcal{P}$ ending in m there exists at most one cycle having m. This proves our first result:

Theorem 2.1 The energy-constrained infinite-run problem is decidable for flat SETA.

Notice that the technique does not trivially extend to SETAs with nested cycles, because they may have infinitely many different cycles.

Example 2.5 Consider the SETA $\mathcal{A} = (S, T, P)$ depicted in Fig. 5 and the energy constraint $E = [0;\ 6]$. We describe a step-by-step execution of Algorithm 1 starting with $s_0 \in S$ and initial energy interval $I_0 = [0;\ 0]$.

The waiting list is initialised as $W_0 = \{(s_0, I_0, c)\}$. After the first execution of the main **while** loop, $W_1 = \{(s_0, I_0, \bar{c})\}$ because s_0 does not belong to any simple cycle of (S, T). In the second iteration, we pick the task $(s_0, I_0, \bar{c})$ and we update the waiting list as $W_2 = \{(s_1, [4;\ 4], c), (s_2, [0;\ 1], c)\}$. In the third iteration, we pick the task $(s_2, [0;\ 1], c)$ from W_2. Since s_2 belongs to the self-cycle $\mathcal{C} = (s_2, s_2)$, we compute $[0;\ 1] \cap \nu\mathcal{R}_\mathcal{C}^{-1} = [0;\ 1] \cap [\frac{5}{3};\ 6] = \emptyset$. Thus, we proceed by computing $\mathcal{R}^0([0;\ 1]) = [0;\ 1]$, $\mathcal{R}^1([0;\ 1]) = [0;\ 0]$ and $\mathcal{R}^2([0;\ 1]) = \emptyset$, and update the waiting list as $W_3 = \big(W_2 \setminus (s_2, [0;\ 1], c)\big) \cup \{(s_2, [0;\ 1], \bar{c}), (s_2, [0;\ 0], \bar{c})\}$. In the fourth and fifth iterations, we pick the tasks $(s_2, [0;\ 1], \bar{c})$ and $(s_2, [0;\ 0], \bar{c})$, respectively. Since s_2 cannot escape from the self-cycle, we will not insert any tasks in the waiting list, thus having $W_5 = \{(s_1, [4;\ 4], c)\}$. During the sixth iteration, we pick the task $(s_1, [4;\ 4], c)$. Since s_1 belongs to the self-cycle $\mathcal{C}' = (s_1, s_1)$, we compute $[4;\ 4] \cap \nu\mathcal{R}_{\mathcal{C}'}^{-1} = [4;\ 4] \cap \emptyset = \emptyset$.

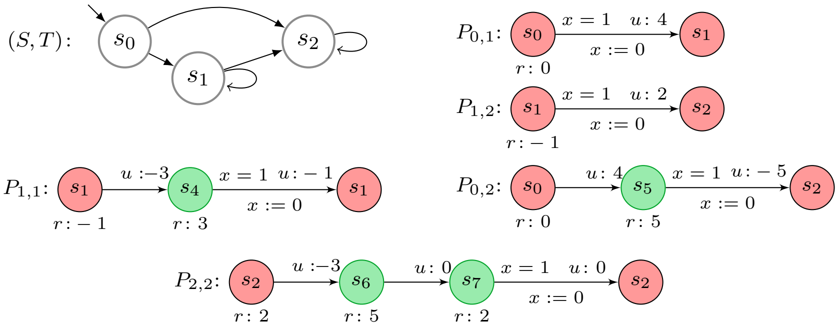


Fig. 5. An example of SETA $\mathcal{A} = (S, T, P)$ with implicit global variant $x \leq 1$. The map P associates with each $(s_i, s_j) \in T$ the ETP $P_{i,j}$.

Thus we proceed by computing $\mathcal{R}^0([4;\ 4]) = [4;\ 4]$, $\mathcal{R}^1([4;\ 4]) = [0;\ 3]$, $\mathcal{R}^2([4;\ 4]) = [2;\ 2]$, and $\mathcal{R}^3([4;\ 4]) = \emptyset$ and obtaining $W_6 = \big(W_5 \setminus (s_1, [4;\ 4], c)\big) \cup \{(s_1, [4;\ 4], \bar{c}), (s_1, [0;\ 3], \bar{c}), (s_1, [2;\ 2], \bar{c})\}$. In the seventh iteration, we pick the task $(s_1, [4;\ 4], \bar{c})$. The only transition that escapes from the self-cycle of s_1 is (s_1, s_2), thus we get $W_7 = \big(W_6 \setminus (s_1, [4;\ 4], \bar{c})\big) \cup \{(s_2, [5;\ 5], c)\}$. Finally, we pick the task $(s_2, [5;\ 5], c)$ and since $[5;\ 5] \cap \nu\mathcal{R}^{-1}_{\mathcal{C}''} = [5;\ 5] \cap [\frac{5}{3};\ 6] \neq \emptyset$ where $\mathcal{C}'' = (s_2, s_2)$, we stop the computation and return $\mathtt{tt}$.

We are now ready to prove our second main result of this section.

Theorem 2.2 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.

Proof Let $\mathcal{A}$ be a flat SETA and $L \in \mathbb{Q}$ be the fixed lower bound.

Let $\mathcal{C}$ be a simple cycle of $\mathcal{A}$ (which may formally be the concatenation of several energy timed paths but w.l.o.g. we can assume it is a single energy timed path). We analyze when this cycle can be iterated, and for which upper bound U. Adding U as a parameter, we can refine the approach of Sect. 2, and safely define the ternary energy relation $\mathcal{R}_{\mathcal{C}}(w_0, w_1, U)$ as $\mathcal{R}^{[L;\ U]}_{\mathcal{C}}(w_0, w_1)$. It is a convex subset of $\mathbb{R}^3$, described as a conjunction of a finite set of linear constraints over w_0, w_1 and U (with non-strict inequalities and rational coefficients). We can then define the predicate $\mathcal{R}^{\infty}_{\mathcal{C}}(a, b, U)$ as:

$$\mathcal{R}^{\infty}_{\mathcal{C}}(a, b, U) \iff L \leq a \leq b \leq U \wedge \forall\, w_0 \in [a;\ b],\ \exists\, w_1 \in [a;\ b].\ \mathcal{R}_{\mathcal{C}}(w_0, w_1, U)$$

characterizing the intervals $[a;\ b]$ and upper-bounds U such that $\mathcal{C}$ can be iterated infinitely many times from any initial value in $[a;\ b]$ with energy constraint $[L;\ U]$. This relation is again a closed convex subset of $\mathbb{R}^3$, described as a conjunction of a finite set of linear constraints over a, b and U (with non-strict inequalities and rational coefficients).

For a fixed $U \in \mathbb{Q}$, this predicate coincides with the greatest fixpoint $\nu(\mathcal{R}^{[L;\ U]}_{\mathcal{C}})^{-1}$ that was discussed on page 9. Hence $\mathcal{R}^{\infty}_{\mathcal{C}}(a, b, U)$ holds if, and only if, for every $w_0 \in [a;\ b]$, there is an infinite run starting at $(s_0, \mathbf{0}, w_0)$ (where s_0 is the first state of $\mathcal{C}$) satisfying the energy constraint $[L;\ U]$. Furthermore, the set $\{a \in \mathbb{R} \mid \exists\, b, U.\ \mathcal{R}^{\infty}_{\mathcal{C}}(a, b, U)\}$ is a closed subset of $\mathbb{R}$, defined as a conjunction of linear constraints with rational coefficients, and bounded below by L; thus there is a least value $a^{\mathcal{C}}_{\min} \in \mathbb{Q}$ such that the set $\{(b, U) \mid \mathcal{R}^{\infty}_{\mathcal{C}}(a^{\mathcal{C}}_{\min}, b, U)\}$ is non-empty. For this value $a^{\mathcal{C}}_{\min}$:

Lemma 2.2 The following properties hold

- For any energy level $w < a^{\mathcal{C}}_{\min}$, and for any U, there are no infinite runs from $(s_0, \mathbf{0}, w)$ cycling around $\mathcal{C}$ and satisfying energy constraint $[L;\ U]$;
- For every $w \geq a^{\mathcal{C}}_{\min}$, there exist U and an infinite run from $(s_0, \mathbf{0}, w)$ cycling around $\mathcal{C}$ and satisfying energy constraint $[L;\ U]$.

Proof The first part of the lemma is a direct consequence of the analysis of the fixed point $\nu(\mathcal{R}^{[L;\ U]}_{\mathcal{C}})^{-1}$ made in Sect. 2.1.

For the second property, we first realize that there is $(b, U) \in \mathbb{Q}^2$ such that $\mathcal{R}^{\infty}_{\mathcal{C}}(a^{\mathcal{C}}_{\min}, b, U)$ (because relation $\mathcal{R}^{\infty}_{\mathcal{C}}(a, b, U)$ is a finite conjunction of linear contraints with rational coefficients). This means in particular that there is an infinite run from $(s_0, \mathbf{0}, a^{\mathcal{C}}_{\min})$ cycling around $\mathcal{C}$ and satisfying the energy constraint $[L;\ U]$. By mimicking the same delays from $(s_0, \mathbf{0}, a^{\mathcal{C}}_{\min})$, we create an infinite run along which the energy levels are simply shifted up by $w - a_{\min}$. This way, we have built an infinite run from $(s_0, \mathbf{0}, w)$ satisfying the energy constraint $[L;\ U + w - a^{\mathcal{C}}_{\min}]$. □

Coming back to our automaton $\mathcal{A}$: if there is a solution to the energy-constrained infinite-run problem in $\mathcal{A}$ for some upper bound U, the witness infinite run must end up cycling in one of the cycles of $\mathcal{A}$. Let $\mathcal{C}$ be a cycle. We know from the lemma above that, to be able to generate a witness infinite run cycling around $\mathcal{C}$, one needs to be able to reach the start of that cycle with at least energy level $a^{\mathcal{C}}_{\min}$. Note that if we find a finite run reaching the start of cycle $\mathcal{C}$ with energy level $w \geq a^{\mathcal{C}}_{\min}$ and satisfying the energy constraint $[L;\ +\infty)$ (only a lower-bound constraint) along the way, then for some U' this finite path satisfies the energy constaint $[L;\ U']$; the concatenation of that finite run with a witness infinite run cycling along $\mathcal{C}$ while satisfying some $[L;\ U]$-energy constraint gives a witness infinite run for the existence of an upper bound (with upper bound $\max(U;\ U')$).

We therefore study finite runs leading to the start of cycle $\mathcal{C}$, with only the lower bound L on the energy level. Recall that this problem is in general not easy to solve [BLM14], and only single-clock automata can be handled in general [BFLM10]. However in the special setting of flat SETA, we are able to decide the existence of a well-adapted finite run reaching the start of cycle $\mathcal{C}$. Let $\mathcal{P}$ be an energy timed path. Following a similar approach to the approach developed on Sect. 2.1, one can define a predicate $\mathcal{S}_{\mathcal{P}}(w_0, w_1)$ that is true whenever there is a run satisfying the energy constraint $[L;\ +\infty)$, starting with energy level w_0 and ending with energy level w_1. From that predicate, one can derive the predicates $\mathcal{S}^{\uparrow}_{\mathcal{P}}(w_0)$ (resp. $\mathcal{S}^{=}_{\mathcal{P}}(w_0)$, $\mathcal{S}^{\times}_{\mathcal{P}}(w_0)$) such that:

- $\mathcal{S}^{\uparrow}_{\mathcal{P}}(w_0) \iff \exists\, w_1 > w_0$ s.t. $\mathcal{S}_{\mathcal{P}}(w_0, w_1)$;
- $\mathcal{S}^{=}_{\mathcal{P}}(w_0) \iff \mathcal{S}_{\mathcal{P}}(w_0, w_0)$ and $\neg\mathcal{S}^{\uparrow}_{\mathcal{P}}(w_0)$;
- $\mathcal{S}^{\times}_{\mathcal{P}}(w_0) \iff \forall\, w_1 \geq w_0,\ \neg\mathcal{S}_{\mathcal{P}}(w_0, w_1)$.

In the first two cases, and only in these cases, the path can be iterated while satisfying the energy constraint $[L;\ +\infty)$. In the first case, by iterating the path, one can increase the energy level up to an arbitrarily high value. In the second case, only energy level w_0 can be reached. These properties are straightforward (since there is no upper bound), and are therefore omitted.

Let $\mathcal{A}$ be a SETA with initial energy level w_0. We perform the following (partial) labelling λ of the graph in a forward manner:

- we label the initial macro-state m_0 with $\lambda(m_0) = \top$ if there is a path $\mathcal{P}$ from m_0 to itself, where $\mathcal{S}^{\uparrow}_{\mathcal{P}}(e_0)$ holds; otherwise we set $\lambda(m_0) = w_0$.
- let m be a macro-state which does not belong to a cycle, and such that all its predecessors have been already labelled with λ. Write $(m_i)_{1 \leq i \leq p}$ for a non-empty list of its predecessors, with redundancies if there are multiple transitions between macro-states. For each $1 \leq i \leq p$, write $\mathcal{P}_i$ for the ETP labelling the edge (m_i, m). If there is some i such that $\lambda(m_i) = \top$, then set $\lambda(m) = \top$. Otherwise, define w'_i for the largest energy level such that $\mathcal{S}_{\mathcal{P}_i}(w_i, w'_i)$ holds (w'_i can be equal to $+\infty$ whenever w'_i can be made arbitrarily large). If there is a cycle $\mathcal{C}$ starting at m_i such that $\mathcal{S}^{\uparrow}_{\mathcal{C}}(w'_i)$, then set $\lambda(m) = \top$. If $w'_i = +\infty$ for some i, then set $\lambda(m) = \top$, otherwise set $\lambda(m) = \max_{1 \leq i \leq p} w'_i$.

The following lemma concludes the decidability proof for the existence of an upper bound.

Lemma 2.3 There is a solution to the upper-bound existence problem if, and only if, there is a cycle $\mathcal{C}$ starting at some macro-state m in $\mathcal{A}$ such that $a^{\mathcal{C}}_{\min}$ is well-defined, and such that $\lambda(m) = \top$ or $\lambda(m) \geq a^{\mathcal{C}}_{\min}$.

Proof We can prove the following invariant to the labelling algorithm:

- $\lambda(m) = \top$ if, and only if, for every $\alpha \in \mathbb{R}$ there is $w \geq \alpha$ such that energy level w can be achieved when reaching m;
- $\lambda(m) = \alpha$ if, and only if, α is the maximal energy level that can be reached at m. □

It remains to discuss the synthesis of the least upper bound for which there is a solution to the upper bound synthesis problem. In this case, we will restrict to depth-1 flat SETA, that is the graph underlying the SETA is a tree, with self-loops at leaves.[1] We assume we have found a bound U such that $\mathcal{A}$ satisfies the infinite path problem with energy constraint $[L;\ U]$.

Since $\mathcal{A}$ is depth-1, it can be decomposed as a union of timed paths followed by a cycle. Let $\mathcal{P}$ be such a path, followed by cycle $\mathcal{C}$. We assume w.l.o.g. that there is an infinite run satisfying the energy constraint $[L;\ U]$ following $\mathcal{P}$ and cycling along $\mathcal{C}$. We define the predicate $\mathcal{R}_{\mathcal{P}.\mathcal{C}^\omega}(U')$ by

$$U' \leq U \ \wedge\ \exists\, L \leq a \leq w_1 \leq b \leq U' \text{ s.t. } \mathcal{R}_{\mathcal{P}}(w_0, w_1, U') \text{ and } \mathcal{R}^{\infty}_{\mathcal{C}}(a, b, U')$$

Then $\mathcal{R}_{\mathcal{P}.\mathcal{C}^\omega}(U')$ holds if, and only if, $U' \leq U$ is a correct upper bound for a witness along $\mathcal{P} \cdot \mathcal{C}^\omega$. We can simplify the predicate $\mathcal{R}_{\mathcal{P}.\mathcal{C}^\omega}(U')$, and obtain the least upper bound as the smallest U' such that $\mathcal{R}_{\mathcal{P}.\mathcal{C}^\omega}(U')$ holds for some $\mathcal{P}$ and $\mathcal{C}$ in $\mathcal{A}$. □

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 Sect. 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.

3.1. Adding uncertainty to ETA

Definition 3.1 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 $\epsilon \colon S \to \mathbb{Q}_{>0}$ assigning imprecisions to rates of states and $\Delta \colon T \to \mathbb{Q}_{>0}$ assigning imprecisions to updates of transitions.

In the obvious manner, this notion of uncertainty extends to *energy timed paths with uncertainty* (ETPu) as well as to *segmented energy timed automata 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 $v_{2j+2} = v_{2j+1}[z_j \to 0]$;
- 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 $\beta_j \in [u_j - \Delta(t_j), u_j + \Delta(t_j)]$.

Notice that uncertainty only affects the measure of energy, not the measure of time. 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 in the case of uncertainty, any sequence d of delays may have several possible outcomes (and corresponding energy levels) along a given transition sequence τ due to the uncertainty in rates and updates. In particular, we say that τ together with d with 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 sequence d of delays for $\mathcal{P}$ satisfying a given energy constraint E from initial level w_0, we denote by $\mathcal{E}^E_{\mathcal{P},d}(w_0)$ the set of possible final energy levels. It may be seen that $\mathcal{E}^E_{\mathcal{P},d}(w_0)$ is a closed subset of E.

Example 3.1 Figure 6 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 $\mathcal{P}$ and the delay sequence $d = (0.6, 0.4)$ with initial energy level $w_0 = 3$ satisfy the energy constraint $E = [0;\ 5]$. The set of final energy levels in $\mathcal{E}^E_{\mathcal{P},d}(w_0)$ is then $[2.5;\ 3.1]$.

[1] The general case of flat SETA might be solvable, but we do not have a complete proof of that general case yet.

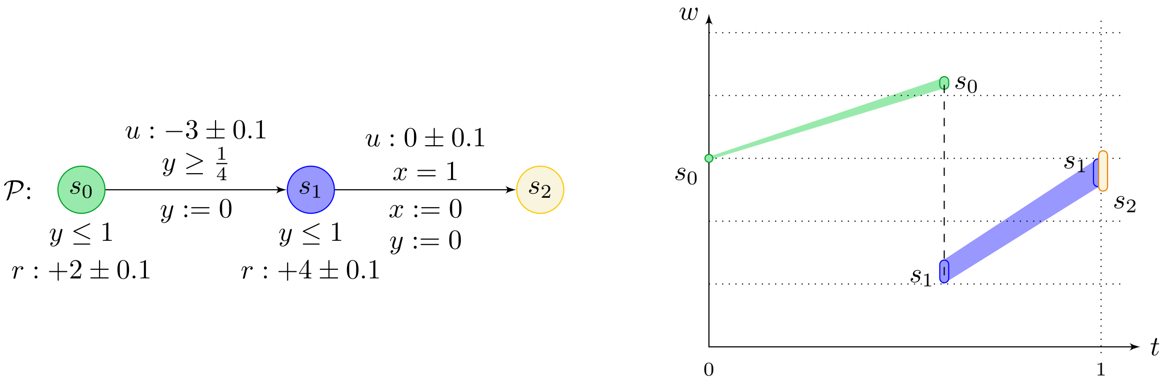


Fig. 6. An energy timed path $\mathcal{P}$ with uncertainty, and a representation of the runs corresponding to the delay sequence (0.6, 0.4) with initial energy level 3.

Now let $\mathcal{A} = (S, T, P)$ be an SETAu and let E be an energy constraint. A (memoryless[2]) *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 $d \in \mathbb{R}^n_{\geq 0}$ 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 $\rho^i = (\ell^i_j, x^i_j, w^i_j)_{0 \leq j \leq 2n_i}$, is an outcome of σ if the following conditions hold:

- s^i_0 and $s^i_{2n_i}$ are macro-states of $\mathcal{A}$, and ρ^i is a possible outcome of $P(s^i_0, s^i_{2n_i})$ for d where $\sigma(s^i_0, w^i_0) = \big((s^i_0, s^i_{2n_i}), d\big)$;
- $s^{i+1}_0 = s^i_{2n_i}$ and $w^{i+1}_0 = w^i_{2n_i}$.

Now we may formulate the infinite-run problem in the setting of uncertainty:

Definition 3.2 Let $\mathcal{A}$ be a SETAu, $E \in \mathcal{I}(\mathbb{Q})$ be an energy constraint, and (s_0, w_0) an initial macro-configuration (s_0 macro-state of $\mathcal{A}$ and $w_0 \in E$ energy level). The *energy-constrained infinite-run problem* is as follows: does there exist a strategy σ for $\mathcal{A}$ such that all runs $(\rho^i)_i$ that are outcomes of σ starting from configuration (s_0, w_0) satisfy E?

3.2. 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 $\mathcal{U}^E_{\mathcal{P}} \subseteq E \times E \times E$ relates all triples (w_0, a, b) for which there is a sequence of delays whose outcomes from $(s_0, \mathbf{0}, w_0)$ all satisfy E and end 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:

$$\begin{aligned}\mathcal{U}^E_{\mathcal{P}}(w_0, a, b) \iff{} & \exists (d_i)_{0 \leq i < n}. \forall (\alpha_i \in [r(s_i) - \epsilon(s_i);\ r(s_i) + \epsilon(s_i)])_{0 \leq i < n}. \\ & \forall (\beta_i \in [u_j - \Delta(t_j);\ u_j + \Delta(t_j)])_{0 \leq i < n}. \ \Phi_{\text{timing}} \wedge \Phi^u_{\text{energy}} \wedge a \leq w_0 + \sum_{k=0}^{n-1} (d_k \cdot \alpha_k + \beta_k) \leq b\end{aligned}$$

where Φ_{timing} encodes all the timing constraints that the sequence $(d_i)_{0 \leq i < n}$ has to fulfill and is identical to that used in the case of full precision. Also Φ^u_{energy} encodes the energy constraints relative to E. Formula Φ^u_{energy} is similar to Φ_{energy} from Sect. 2, but refers to α_i and β_i rather than to the nominal rates $r(s_j)$ and updates u_i.

The expression above has two drawbacks: it mixes existential and universal quantifiers (which may severely impact efficiency), and the arithmetic expression is quadratic (for which no efficient tools provide quantifier elimination). A better way to characterize the ternary relation is by expressing inclusion of the set of reachable energy levels in the energy constraint:

[2] For the infinite-run problem we consider it may be shown that memoryless strategies suffice.

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

where Φ^i_{energy} 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. This way, we get linear arithmetic expressions and no quantifier alternations. Applying Fourier-Motzkin elimination, $\mathcal{U}^E_{\mathcal{P}}$ 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.

Example 3.2 We illustrate the above translation on the ETPu of Fig. 6. For energy constraint [0; 5], the energy relation can be written as:

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

Applying quantifier elimination, we end up with:

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

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. 6). We get $w_0 \in [7.4/3;\ 13.78/4.1]$, which is (under-)approximately $w_0 \in [2.467;\ 3.360]$.

3.3. 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:

$$(s, w) \in \mathcal{W} \Rightarrow \exists\, t = (s, s') \in T.\ \exists\, a, b \in E.\, \mathcal{U}^E_{P(t)}(w, a, b) \wedge \forall\, w' \in [a;\ b].\ (s', w') \in \mathcal{W} \qquad (3)$$

This expresses that from any $(s, \mathbf{0}, w)$ with $(s, w) \in \mathcal{W}$, there is a macro-transition (s, s') that can be taken, ending up in configurations $(s', \mathbf{0}, w')$ with $(s', w') \in \mathcal{W}$. Thus $\mathcal{W}$ characterizes the set of configurations (s, w) that satisfy the energy-constained 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 3.1 Let $\mathcal{A}$ be an SETAu satisfying **(R)**, $E \in \mathcal{I}(\mathbb{Q})$ an energy constraint, and (s_0, w_0) an initial macro-configuration. Then it is decidable whether the energy-constrained infinite-run problem is satisfied.

Proof Under hypothesis **(R)**, there is a minimum level of imprecision for any transition $t = (s, s')$: whenever $\mathcal{U}^E_{P(t)}(w, a, b)$ then $\mid b - a \mid \geq D\cdot\Delta_{\min}$, where $\Delta_{\min}$ is the minimal imprecision within all ETPu $P(t)$ of $\mathcal{A}$. Thus if $(s, w) \in \mathcal{W}$ "due to" some transition $t = (s, s')$, then for some interval $[a, b]$ with $\mid b - a \mid \geq D\cdot\Delta_{\min}$ all configurations (s', w') with $w' \in [a, b]$ must be in $\mathcal{W}$. Now let $N = \left\lceil \frac{|E|}{D\cdot\Delta_{\min}} \right\rceil$. It follows that the subset of E given by $\mathcal{W}_s = \{w \mid (s, w) \in \mathcal{W}\}$ may be divided into at most N disjoint intervals $[a_{s,j}, b_{s,j}]$ $(1 \leq j \leq N)$, each of size at least $D\cdot\Delta_{\min}$. We may therefore characterize the set of configurations (s_0, w_0) satisfying the energy-constained infinite-run problem as being those for which there exist values $(a_{s,j}, b_{s,j})_{s\in S, 1\leq j\leq N}$ such that

$$w_0 \in \bigcup_{1\le j\le N} [a_{s_0,j};\ b_{s_0,j}] \wedge \bigwedge_{s\in S} \bigwedge_{1\le j\le N} \Big([a_{s,j};\ b_{s,j}] \subseteq E \wedge \\ \forall\, w \in [a_{s,j};\ b_{s,j}].\ \bigvee_{(s,s')\in T} \Big(\exists\, a,\, b \in E.\ \mathcal{U}^{E}_{P(s,s')}(w,\, a,\, b) \wedge \bigvee_{1\le k\le N} ([a;\ b] \subseteq [a_{s',k};\ b_{s',k}])\Big)\Big) \quad (4)$$

By quantifier elimination, the above may be rewritten as a boolean combination of linear constraints over the variables $a_{s,j}$, $b_{s,j}$, and determining the satisfiability of the formula is decidable. □

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.

Example 3.3 We pursue on Example 3.2. If ETPu $\mathcal{P}$ is iterated (as on the loop on state m_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 Eq. (4), as follows:

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

We end up with

$$2.435 \le a \wedge b \le 3.635 \wedge b \ge a + 0.6.$$

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 2.4 for the same cycle without uncertainty).

3.4. Synthesis of optimal upper bound

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

Theorem 3.2 Let $\mathcal{A} = (S,\, T,\, P)$ be a depth-1 flat SETAu. 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 one exists.

Proof First, for a cycle ETPu $\mathcal{C}$ and a lower energy bound L, we may define a quaternary relation $\mathcal{X}^{L}_{\mathcal{C}}$ on E such that $\mathcal{X}^{L}_{\mathcal{C}}(w,\, a,\, b,\, U)$ holds if, and only if, $\mathcal{U}^{[L;\ U]}_{\mathcal{C}}(w,\, a,\, b)$. Clearly $\mathcal{X}^{L}_{\mathcal{C}}$ can be described as a first-order formula over linear arithmetic, and, by quantifier elimination, as a boolean combination of linear constraints over w, a, b and U.

Now, since $\mathcal{A}$ is a depth-1 flat SETAu, we can assume w.l.o.g. that $\mathcal{A}$ consists in a path followed by a cycle that one tries to iterate. This is no loss of generality since a depth-1 flat SETAu can be seen as a finite union of such simple automata. Hence we assume $\mathcal{A} = (S,\, T,\, P)$ has two macro states s and s', and two macro-transitions $(s,\, s')$ and $(s',\, s')$. We let $\mathcal{P}$ be the path $P(s,\, s')$ and $\mathcal{C}$ be $P(s',\, s')$. Since we consider only one cycle, we can capture $\mathcal{W}_{s'}$ with a single interval $[a_{s'};\ b_{s'}]$. For any given U, following the idea of Eq. (4), the set of configurations $(s_0,\, w_0)$ satisfying the energy-constrained infinite-run problem is the set for which there exist $a_{s'}$ and $b_{s'}$ such that

$$w_0 \in [L;\ U] \wedge \exists\, a,\, b.\ \mathcal{X}^{L}_{\mathcal{P}}(w_0,\, a,\, b,\, U) \wedge [a;\ b] \subseteq [a_{s'};\ b_{s'}] \subseteq [L;\ U] \wedge \\ \forall\, w \in [a_{s'};\ b_{s'}].\ \exists\, a',\, b' \ge L'.\ \mathcal{X}^{L}_{\mathcal{C}}(w,\, a',\, b',\, U) \wedge [a';\ b'] \subseteq [a_{s'};\ b_{s'}]$$

By quantifier elimination, the above may be rewritten as a boolean combination of linear constraints over the variables $a_{s'}$, $b_{s'}$ and U, denoted by $\varphi(a_{s'},\, b_{s'},\, U)$. Determining the satisfiability of the formula $\varphi(a_{s'},\, b_{s'},\, U)$ is decidable. In addition, eliminating the quantifiers in the formula $\exists\, a_{s'},\, b_{s'}.\ \varphi(a_{s'},\, b_{s'},\, U)$ yields a boolean combination of linear constraints over the single variable U. For the fact that such a formula has only one variable, it needs to represent the interval of values for U which admit an energy-constrained infinite run. Clearly, the lower bound of such interval is the minimal value of U. □

4. Case study

In this section we present an industrial case study that was provided by the HYDAC company in the context of a European research project Quasimodo [Qua]. The case study consists in an on-off control system where the system to be controlled, depicted in Fig 1a, is composed of (i) a machine that consumes oil, (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 in the accumulator. When the system is operating, the machine consumes oil under pressure out of the accumulator. The level of the oil, and so the pressure within the accumulator, can be controlled by pumping additional oil in the accumulator (thereby increasing the gas pressure). The control objective is twofold: first the level of oil into the accumulator (and so the gas pressure) shall be maintained within a safe interval; second, at the end of each operating cycle, the accumulator shall be in a state that ensures the controllability of the following cycle. Besides these safety requirements, the controller should also try to minimize the oil level in the tank, so as to not damage the system.

4.1. Modelling the oil pump system

In this section we describe the characteristics of each component of the HYDAC case. Then we model the system 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 Fig. 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 $E = [V_{\min}; \ V_{\max}]$[3] (R2) at the end of each machine cycle, the level of oil in the accumulator must ensure the controllability of the following cycle.

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 minimise $V_{\max} - V_{\min}$, 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 in Fig. 7. In order to fully model the behaviour 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 Sect. 5.

Instead, we consider a simplified 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. 7, a copy of the ETP depicted on Fig. 8. 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 equiped with a self-loop labelled with the ETP above.

In order to take into account 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. 8 only after the first, third, fifth, seventh and ninth states of the ETP of Fig. 7.

[3] The HYDAC company has fixed $V_{\min} = 4.9$ l and $V_{\max} = 25.1$ l.

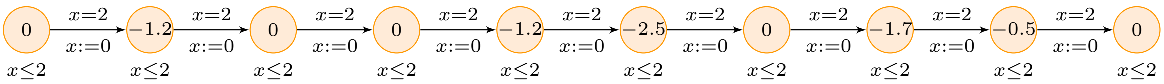


Fig. 7. The ETP representing the oil consumption of the machine.

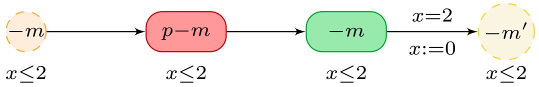


Fig. 8. An ETP for modelling the pump

Table 1. Characteristics of the synthesised strategies, compared with the strategies proposed in [CJL+09, ZZKL12]. (∗) Safety interval as specified by the HYDAC company.

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 [CJL+09]	[4.9; 25.1](∗)	[5.1; 9.4]	8.2
G2M1 [CJL+09]	[4.9; 25.1](∗)	[5.1; 8.3]	7.95
[ZZKL12]	[4.9; 25.1](∗)	[5.2; 8.1]	7.35

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.

4.2. Synthesizing controllers

For each model, we synthesise minimal upper bounds U (within the interval $[V_{\min};\ V_{\max}]$) that admit a solution to the energy-constrained infinite-run problem for 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 Sects. 2 and 3 where quantifier elimination is performed using Mjollnir [Mon10].

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 Sects. 2 and 3. We use Mjollnir [Mon10] 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 [Wol] to minimize the non-linear cost function expressing the average oil volume subject to the linear constraints obtained above. Figure 9 shows the resulting strategies: there, each horizontal section of the graph represents a strategy for an entire pump cycle when the system enters the cycle at a given initial oil level (measured in decilitres). The green intervals indicate where the pump, according to the strategy, will be running.

The first part of Table 1 summarises 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 synthesised both in [CJL+09] and [ZZKL12].

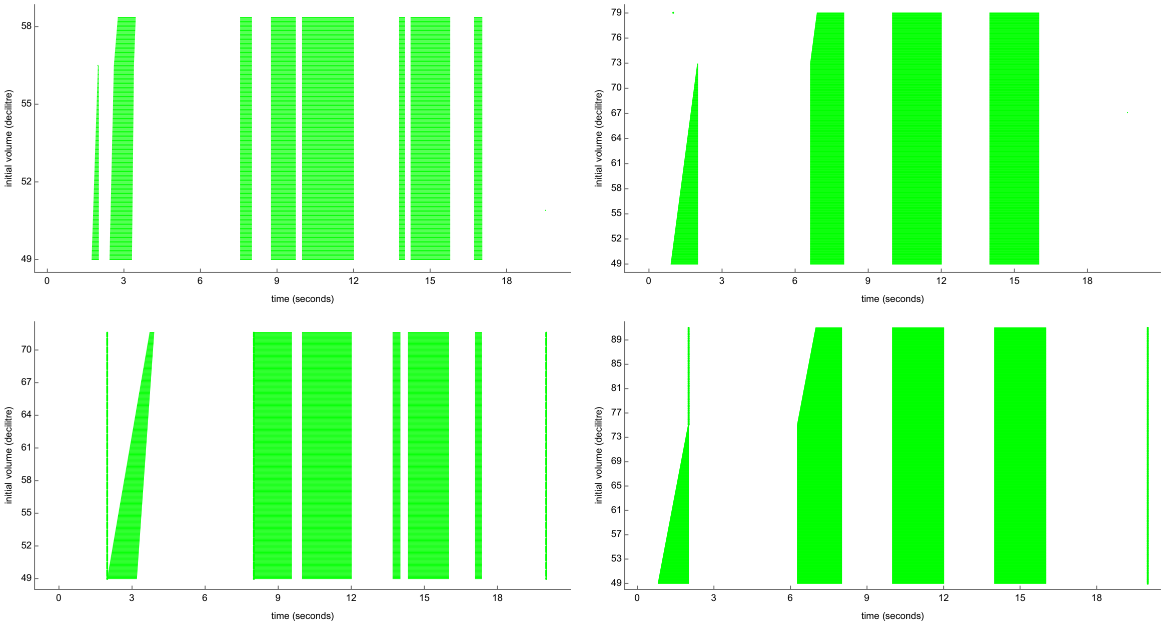


Fig. 9. Local strategies for a single cycle of the HYDAC system. (top-left) $\mathcal{H}_1$; (top-right) $\mathcal{H}_2$; (bottom-left) $\mathcal{H}_1(\epsilon)$; (bottom-right) $\mathcal{H}_2(\epsilon)$ ($\epsilon = 0.1\ l/s$).

The reason for this may be that (i) our models relax the latency requirement for the pump, (ii) the strategies of [CJL+09] are obtained using a discretisation of the dynamics within the system, and (iii) the strategies of [CJL+09] and [ZZKL12] 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. Figure 10 shows the result of simulating our strategies for a duration of 200 s, i.e., 10 consecutive machine cycles. 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—a green interval indicates that in that period the pump is on. The initial volume used for evaluating the strategies is 8.3 l, as done in [CJL+09] for evaluating respectively the Bang-Bang controller, the Smart Controller developed by HYDAC, and the controllers G1M1 and G2M1 synthesised with UPPAAL- TIGA.[4]

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 [CJL+09]. In particular the strategy for $\mathcal{H}_1(\epsilon)$ provides an improvement of 55%, 46%, 20%, and 19% respectively for the Bang-Bang controller, the Smart Controller of HYDAC, and the two strategies synthesised with UPPAAL- TIGA.

One can see in the plots in Fig. 10 all the strategies start by keeping the pump off until the oil volume reaches a level within a stable interval (*cf.* Table 1). From there on, by following the strategies described in Fig. 9, the oil level varies following a repetitive pattern.

[4] We refer the reader to [CJL+09] for a more detailed description of the controllers.

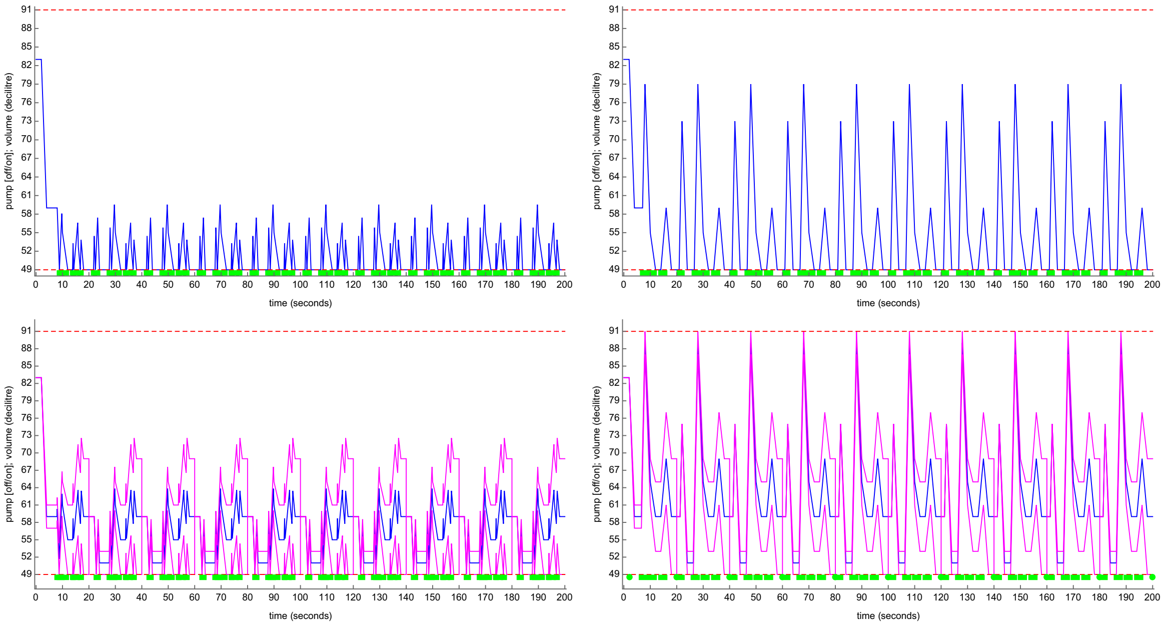


Fig. 10. Simulations of 10 consecutive machine cycles, started at initial oil level 83 decilitres, performed resp. with the strategies for (top-left) $\mathcal{H}_1$; (top-right) $\mathcal{H}_2$; (bottom-left) $\mathcal{H}_1(\epsilon)$; and (bottom-right) $\mathcal{H}_2(\epsilon)$.

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

Controller	Acc. vol. (l)	Mean vol. (l)
$\mathcal{H}_1$	1081.77	5.41
$\mathcal{H}_2$	1158.9	5.79
$\mathcal{H}_1(\epsilon)$	1200.21	6.00
$\mathcal{H}_2(\epsilon)$	1323.42	6.62
Bang-Bang	2689	13.45
HYDAC	2232	11.6
G1M1	1518	7.59
G2M1	1489	7.44

4.3. Tool chain

Our results have been obtained using Mathematica [Wol] and Mjollnir [Mon10]. Specifically, Mathematica was used to construct the formulas modelling the post-fixpoints of the energy functions, calling Mjollnir for performing quantifier elimination on them. The computation of the optimal upper bounds, and greatest stable intervals were then handled with Mathematica, as well as the computation of the optimal schedules and the respective simulations. It is worth mentioning that Mathematica provides the built-in function `Resolve` for preforming quantifier elimination, but Mjollnir was preferred to it both for its performances and its concise output. 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.

The Mathematica source code as well as the set up of our experiments are available at http://people.cs.aau.dk/giovbacci/tools.html.

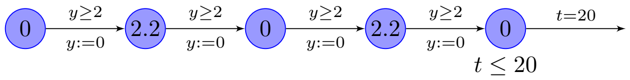


Fig. 11. An ETP modelling the pump

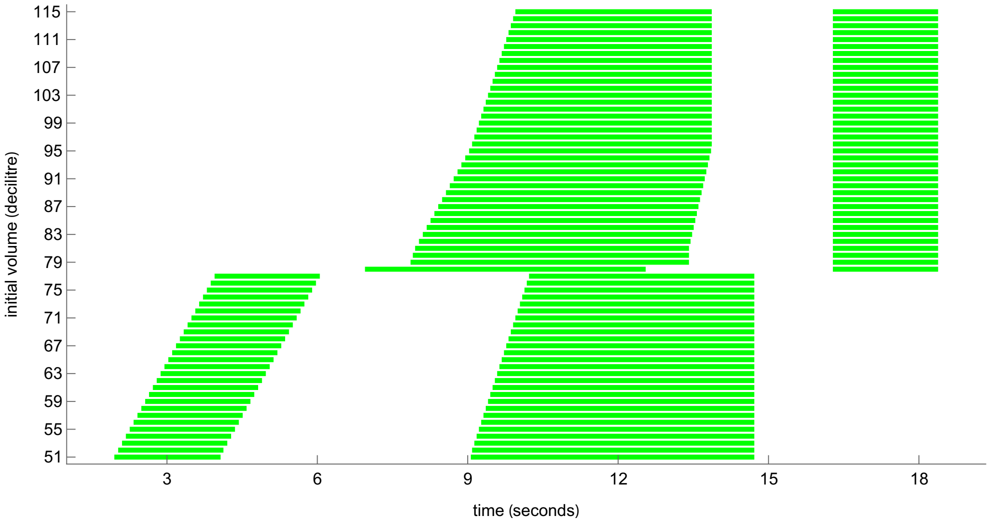


Fig. 12. Strategies for the m-stable interval [5.1; 8.9] l (for $U = 11.5$ l)

5. Non-flat model of the HYDAC case

We briefly present a more precise model of the HYDAC example, closer to what appeared in [CJL$^+$09], using a non-flat SETA. The model is built by considering two flat ETPs running in parallel: one ETP models the consumption cycle of the machine (with fixed delays; see Fig. 7), and the second one models the state of the pump over a complete cycle of the machine, allowing for instance at most 4 switches during one cycle (see Fig. 11). This almost exactly corresponds to the model considered in [CJL$^+$09].

The resulting model is an ETA, which can actually be turned into a non-flat SETA. Hence it only fits in our framework with uncertainty. However, for fixed L and U, it is still possible to write down the energy relation, with or without uncertainty: it results in a (large) list of cases, because of interleaving.

Following [CJL$^+$09], we then compute m-stable intervals, i.e., intervals $[a;\ b]$ of oil levels for which there is a schedule to end up with final oil level in $[a + m;\ b - m]$. In the absence of uncertainties, fixing $L = 4.9$ l and $m = 0.4$ l, we could then prove that there are m-stable intervals as soon as $U \geq 8.1$ l.

With uncertainties, we obtain an m-stable interval [5.1; 8.9] l as soon as $U \geq 11.5$ l. This again significantly improves on [CJL$^+$09] (which considered discrete time). Notice we did not apply our algorithm based on Formula (4) here (hence we may have missed better solutions): the formula would be very large, and would involve $(U - L)/0.2$ intervals $[a_{s,j};\ b_{s,j}]$ for each state of the automaton; this is much more than what our approach can currently handle.

For the m-stable interval [5.1; 8.9] l, we computed the constraints characterising all safe strategies. Figure 12 displays our strategies (notice the similarities with Fig. 5 of [CJL$^+$09]). We were not able to select the optimal strategy for the mean volume because expressing the mean volume results in a piecewise-quadratic function. Instead we selected the strategy that fills in the tank as late as possible (which intuitively tends to reduce the mean volume over one cycle). Figure 13 shows a simulation performed over 10 cycles (which correspond to 200 s) starting from initial volume 8.3 l. As before, the plot illustrates 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—time intervals where the pump is on are indicated in green.

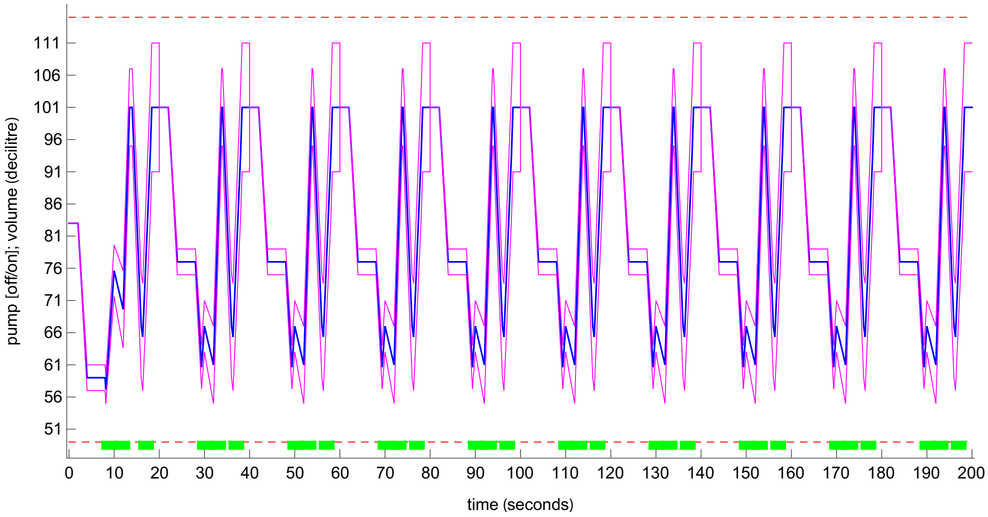


Fig. 13. Simulation of 10 cycles

For this experiment we obtain a total accumulated volume of 1728.85 l, having mean accumulated volume 8.64 l within the uncertainty interval [8.14, 9.14] l. In contrast with Fig. 10, in the simulation depicted in Fig. 13 the oil level never touched the lower bound of the m-stable interval. This may indicate that the proposed strategy may not be optimal.

Remark 5.1 We recall that energy-constrained infinite-run problem is in general undecidable. Hence the strategies proposed in [CJL+09] and [ZZKL12] for the HYDAC case are based on heuristics or models which used permissive energy intervals where the scheduling could be (semi-automatically) proven to be found.

In this paper we showed that for the subclass of flat SETA, the energy-constrained infinite-run problem is, in fact, decidable (Theorem 2.1) and that optimal energy bounds can be computed (Theorem 2.2). The class of flat SETA is quite expressive, though it has some limitations. These limitations emerged also in our case study.

6. 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 work includes extending our results to non-flat and non-segmented energy timed automata. Existing results [Mar11] indicate that we are close to the boundary of decidability, but we believe that by extending the work on energy relations (Sect. 2.1) along the lines of [CFL19], it should be possible to further expand the horizon of our decidability results towards the boundary set in [Mar11].

Another interesting continuation of this work would be to add Uppaal Stratego [DJL+14, DJL+15] 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.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

References

[ACHH93] Alur R, Courcoubetis C, Henzinger TA, Ho P-H (1993) Hybrid automata: an algorithmic approach to the specification and verification of hybrid systems. In: Grossman RL, Nerode A, Ravn AP, Rischel H (eds) Hybrid systems. Springer, Berlin, pp 209–229

[AD94] Alur R, Dill DL (1994) A theory of timed automata. Theor Comput Sci 126(2):183–235

[ALP01] Alur R, La Torre S, Pappas GJ (2001) Optimal paths in weighted timed automata. In: Di Benedetto MD, Sangiovani-Vincentelli AL (eds) Proceedings of the 4th international workshop on hybrid systems: computation and control (HSCC'01), volume 2034 of lecture notes in computer science. Springer, Berlin, pp 49–62

[BBF$^+$18] Bacci G, Bouyer P, Fahrenberg U, Larsen KG, Markey N, Reynier P-A (2018) Optimal and robust controller synthesis—using energy timed automata with uncertainty. In: Havelund K, Peleska J, Roscoe B, de Vink EP (eds) Formal methods—22nd international symposium, FM 2018, volume 10951 of lecture notes in computer science. Springer, Berlin, pp 203–221

[BBKT01] Blondel VD, Bournez O, Koiran P, Tsitsiklis JN (2001) The stability of saturated linear dynamical systems is undecidable. J Comput Syst Sci 62(3):442–462

[BCD$^+$07] Behrmann G, Cougnard A, David A, Fleury E, Larsen KG, Lime D (2007) UPPAAL-Tiga: time for playing games! In: Damm W, Hermanns H (eds) Computer aided verification, 19th international conference, CAV 2007, Berlin, Germany, July 3–7, 2007, proceedings, volume 4590 of lecture notes in computer science. Springer, Berlin, pp 121–125

[BFH$^+$01] Behrmann G, Fehnker A, Hune T, Larsen KG, Pettersson P, Romijn J, Vaandrager F (2001) Minimum-cost reachability for priced timed automata. In: Di Benedetto MD, Sangiovani-Vincentelli AL (eds) Proceedings of the 4th international workshop on hybrid systems: computation and control (HSCC'01), volume 2034 of lecture notes in computer science. Springer, Berlin, pp 147–161

[BFL$^+$08] Bouyer P, Fahrenberg U, Larsen KG, Markey N, Srba J (2008) Infinite runs in weighted timed automata with energy constraints. In: Cassez F, Jard C (eds) Proceedings of the 6th international conferences on formal modelling and analysis of timed systems (FORMATS'08), volume 5215 of lecture notes in computer science. Springer, Berlin, pp 33–47

[BFLM10] Bouyer P, Fahrenberg U, Larsen KG, Markey N (2010) Timed automata with observers under energy constraints. In: Johansson KH, Yi W (eds) Proceedings of the 13th international workshop on hybrid systems: computation and control (HSCC'10). ACM Press, pp 61–70

[BFTM00] Bemporad A, Ferrari-Trecate G, Morari M (2000) Observability and controllability of piecewise affine and hybrid systems. IEEE Trans Autom Control 45(10):1864–1876

[BGH$^+$16] Bisgaard M, Gerhardt D, Hermanns H, Krcál J, Nies G, Stenger M (2016) Battery-aware scheduling in low orbit: the GomX-3 case. In: Fitzgerald JS, Heitmeyer CL, Gnesi S, Philippou A (eds) FM 2016: Formal methods—21st international symposium, Limassol, Cyprus, November 9–11, 2016, proceedings, volume 9995 of lecture notes in computer science, pp 559–576

[BIL06] Bozga M, Iosif R, Lakhnech Y (2006) Flat parametric counter automata. In: Bugliesi M, Preneel B, Sassone V, Wegener I (eds) Proceedings of the 33rd international colloquium on automata, languages and programming (ICALP'06))— part II, volume 4052 of lecture notes in computer science. Springer, Berlin, pp 577–588

[BLM14] Bouyer P, Larsen KG, Markey N (2014) Lower-bound constrained runs in weighted timed automata. Perform Eval 73:91–109, March

[BT99] Blondel VD, Tsitsiklis JN (1999) Complexity of stability and controllability of elementary hybrid systems. Automatica 35(3):479–489

[CDF$^+$05] Cassez F, David A, Fleury E, Larsen KG, Lime D (2005) Efficient on-the-fly algorithms for the analysis of timed games. In: Abadi M, de Alfaro L (eds) 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. Springer, pp 66–80

[CFL19] Cachera D, Fahrenberg U, Legay A (2019) An ω-algebra for real-time energy problems. Log Methods Comput Sci 15(2)

[CJ98] Comon H, Jurski Y (1998) Multiple counters automata, safety analysis, and Presburger arithmetic. In: Hu AJ, Vardi MY (eds) Proceedings of the 10th international conference on computer aided verification (CAV'98), volume 1427 of lecture notes in computer science. Springer, Berlin, pp 268–279

[CJL$^+$09] Cassez F, Jensen JJ, Larsen KG, Raskin J-F, Reynier P-A (2009) Automatic synthesis of robust and optimal controllers—an industrial case study. In: Majumdar R, Tabuada P (eds) Proceedings of the 12th international workshop on hybrid systems: computation and control (HSCC'09), volume 5469 of lecture notes in computer science. Springer, Berlin, pp 90–104

[DJL$^+$14] David A, Jensen PG, Larsen KG, Legay A, Lime D, Sørensen MG, Taankvist JH (2014) On time with minimal expected cost! In: Cassez F, Raskin J-F (eds) 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. Springer, Berlin, pp 129–145

[DJL$^+$15] David A, Jensen PG, Larsen KG, Mikucionis M, Taankvist JH (2015) Uppaal Stratego. In: Baier C, Tinelli C (eds) 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. Springer, Berlin, pp 206–211

[Fre08] Frehse G (2008) PHAVer: algorithmic verification of hybrid systems past HyTech. STTT 10(3):263–279

[JST11] Jha S, Seshia SA, Tiwari A (2011) Synthesis of optimal switching logic for hybrid systems. In: Chakraborty S, Jerraya A, Baruah SK, Fischmeister S (eds) 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. ACM, pp 107–116

[Mar11] Markey N (2011) Verification of embedded systems—algorithms and complexity. Mémoire d'habilitation, École Normale Supérieure de Cachan, France, April

[MFÅL15] Miremadi S, Fei Z, Åkesson K, Lennartson B (2015) Symbolic supervisory control of timed discrete event systems. IEEE Trans Control Syst Technol 23(2):584–597

[Mon10] Monniaux D (2010) Quantifier elimination by lazy model enumeration. In: Touili T, Cook B, Jackson PB (eds) Computer aided verification, 22nd international conference, CAV 2010, Edinburgh, UK, July 15–19, 2010. Proceedings, volume 6174 of lecture notes in computer science. Springer, Berlin, pp 585–599

[PHM14] Phan A-D, Hansen MR, Madsen J (2014) EHRA: specification and analysis of energy-harvesting wireless sensor networks. In: Iida S, Meseguer J, Ogata K (eds) Specification, algebra, and software—essays dedicated to Kokichi Futatsugi, volume 8373 of lecture notes in computer science. Springer, Berlin, pp 520–540

[Qua] Quasimodo. Quantitative system properties in model-driven design of embedded systems. http://www.quasimodo.aau.dk/

[vBHLO17] von Bochmann G, Hilscher M, Linker S, Olderog E-R (2017) Synthesizing and verifying controllers for multi-lane traffic maneuvers. Formal Asp Comput 29(4):583–600

[Wol] Wolfram Research, Inc. (2017) Mathematica, Version 11.2. Champaign, IL

[ZZKL12] Zhao H, Zhan N, Kapur D, Larsen KG (2012) A "hybrid" approach for synthesizing optimal controllers of hybrid systems: a case study of the oil pump industrial example. In: Giannakopoulou D, Méry D (eds) FM 2012: formal methods—18th international symposium, Paris, France, August 27–31, 2012. Proceedings, volume 7436 of lecture notes in computer science. Springer, Berlin, pp 471–485

Received 2 August 2019

Accepted in revised form 17 September 2020 by Erik de Vink, Ana Cavalcanti, Jan Peleska and Bill Roscoe

Published online 24 October 2020