Computing the average inter-sample time of event-triggered control using quantitative automata

de Albuquerque Gleizer, Gabriel; Mazo, Manuel

DOI
10.1016/j.nahs.2022.101290

Publication date
2023

Document Version
Final published version

Published in
Nonlinear Analysis: Hybrid Systems

Nonlinear Analysis: Hybrid Systems 47 (2023) 101290

Contents lists available at ScienceDirect

Nonlinear Analysis: Hybrid Systems

journal homepage: www.elsevier.com/locate/nahs

Computing the average inter-sample time of event-triggered control using quantitative automata ☆

Gabriel de Albuquerque Gleizer *, Manuel Mazo Jr

TU Delft, Mekelweg 2, Delft, 2628 CD, ZH, The Netherlands

ARTICLE INFO

Article history:
Received 29 September 2021
Received in revised form 23 March 2022
Accepted 10 September 2022
Available online xxxx

Keywords:
Event-triggered control
Hybrid systems
Abstractions

ABSTRACT

Event-triggered control (ETC) is a major recent development in cyber–physical systems due to its capability of reducing resource utilization in networked devices. However, while most of the ETC literature reports simulations indicating massive reductions in the sampling required for control, no method so far has been capable of quantifying these results. In this work, we propose an approach through finite-state abstractions to do formal quantification of the traffic generated by ETC of linear systems, in particular aiming at computing its smallest average inter-sample time (SAIST). The method involves abstracting the traffic model through l-complete abstractions, finding the cycle of minimum average length in the graph associated to it, and verifying whether this cycle is an infinitely recurring traffic pattern. The method is proven to be robust to sufficiently small model uncertainties, which allows its application to compute the SAIST of ETC of nonlinear systems.

1. Introduction

In modern control applications, smart sensors, controllers, and actuators communicate with each other through digital communication networks. The standard networked control approach is periodic *sample-and-hold* control: at every h time units, sensors sample their values, send them through the network to the controller, which then updates its control command to the actuators; the command is held constant in between samples. Obviously, small values of the sampling period h approximate the control performance to that of the idealized continuous controller, but increase bandwidth usage and radio energy consumption in wireless networks. This single parameter therefore limits the size and applicability of networked control systems (NCSs), and a natural question that has arisen is how to design aperiodic sampling approaches. In [1], the idea of sampling based on an event – the error between the current state and the last sampled state exceeding a threshold – was investigated with the name of Lebesgue sampling (after the Lebesgue integration). This idea was further developed in [2], where for the first time a framework for asymptotic stabilization of the origin through an event-based sampling was conceived. This approach is now known as *event-triggered control* (ETC), and, given the enormous reductions in sampling it showed in early simulations, immense interested followed. Significant focus was given on event design to reduce sampling frequency while guaranteeing stability and control performance (e.g. [3–5]), extend ETC to different control structures [5], or improve practical implementation aspects of ETC, such as the *periodic event-triggered control* (PETC) of [6], where event conditions are checked periodically. It is remarkable, however, that until very recently [7], no method to formally compute ETC sampling performance existed. Typically, ETC papers limit their formal results to

☆ This work is supported by the European Research Council through the SENTIENT project, ERC-2017-STG #755953.
* Corresponding author.
E-mail address: g.gleizer@tudelft.nl (G. de A. Gleizer).

https://doi.org/10.1016/j.nahs.2022.101290
1751-570X/

stability, control performance, and Zeno-freeness — the absence of Zeno behavior, or infinitely fast sampling in finite time. Similarly to Zeno-freeness, in PETC it is immediate that its average sampling is in the worst case the same as a baseline periodic control whose sampling period is the same h as the event checking period of PETC. The critical question is, *how significant are the savings provided by ETC?* This is a *quantitative* question, and as such it requires the computation of sampling performance metrics for ETC.

As previously mentioned, only recently there has been investigation of ETC traffic patterns, which can be categorized in two main approaches. The first category [8,9] focuses on understanding the qualitative asymptotic trends of the *inter-sample times* (ISTs) of planar linear systems. In [8], the authors conclude that, under some conditions, the ISTs eventually converge to a fixed value or exhibit an oscillatory pattern. Despite providing very interesting insights, the results are limited to two-dimensional state spaces, and do not provide the quantitative information that we consider crucial. The second category uses symbolic abstractions [10,11], following on the extensive work on state-space partitioning and aggregation for abstractions, see [12]. In [10,11], the prediction of ISTs is focused on the scheduling problem: in this context, a scheduler can use finite-state *traffic models* to request sensor data before events are triggered in order to prevent collisions. However, these traffic models do not capture effectively long-term traffic properties of ETC, which hampers their use for quantitative analysis. Still in the same category, [13] uses a bisimulation-like algorithm that determines the m next ISTs from a given state, followed by a very conservative estimate of the worst-case average IST by taking the minimum average of all such m-length sequences.

The present work tackles the precise computation of the smallest (across initial states) average inter-sample time (SAIST) of LTI systems under PETC. The SAIST constitutes a natural metric which directly translates into average resource utilization in a network. Our approach is based on the abstraction of the model of a closed-loop PETC system into a finite-state weighted transition system (WTS), where the weight of a transition is the IST generated by its outbound state. We show that the smallest-in-average cycle (SAC) of the weighted graph associated with the abstraction, which can be computed using Karp's algorithm [14], provides a lower bound of the PETC's SAIST. Moreover, if $\sigma := k_1k_2...k_m$ is the SAC of the abstraction and σ^ω is a behavior of the concrete system, the lower bound is proven to be the exact SAIST of the PETC system. This observation gives rise to the concept of *smallest-average-cycle-equivalent simulation (SACE simulation)*. In contrast, if such a cyclic behavior is not exhibited by the PETC system, the abstraction can be further refined until the cycle breaks, providing tighter bounds. This gives rise to a semi-algorithm to compute average metrics through abstractions. This is an extension of [7], where this semi-algorithm was proposed and the concept of SACE simulation was introduced; here,

(i) we present a general version of the semi-algorithm for verifying the limit average metric of an infinite-state system, as well as some behavioral conditions for its termination and how to compute uncertainty bounds;
(ii) we prove that, in the general case, working with linear invariant subspaces of a linear map is necessary and sufficient to prove that a given SAC can repeat infinitely often as a sequence of ISTs (this was only informally argued in [7]);
(iii) we show that the algorithm is robust to sufficiently small model uncertainties — this enables us to elaborate on the computability of SAIST of linear systems and, moreover, allows the SAIST computation of nonlinear PETC systems;
(iv) we provide more numerical examples and their associated conclusions, including how to decrease the required amount of computations for the abstraction.

The more general results rely on a behavioral interpretation of dynamical systems [15] and the associated abstraction methods [16,17]. The specialized results for PETC SAIST are based on a combination of quotient-based abstractions [12] and a behavioral-based analysis. Overall, our new results help consolidating the methodology proposed in [7], equipping engineers with a tool to formally estimate the benefits of ETC applications.

This paper follows the following structure: The main problem is stated in Section 2. Background and preliminary results about (quantitative) abstractions, including the basic results from [7], are shown in Section 3. Then, a general pseudo-algorithm to compute limit average metrics of infinite systems is presented in Section 4, while its specialization for PETC SAIST computation is presented in Section 5. Finally, numerical examples are given in Section 6, and conclusions and future work are discussed in 7.

1.1. Notation

We denote by $\mathbb{N}_0$ the set of natural numbers including zero, $\mathbb{N} := \mathbb{N}_0 \setminus \{0\}$, $\mathbb{N}_{\leq n} := \{1, 2, \dots, n\}$, by $\mathbb{Q}$ the set of rational numbers, and by $\mathbb{R}_+$ the set of non-negative reals. For a complex number $z \in \mathbb{C}$, z^* denotes its complex conjugate, $\arg z$ denotes its argument, and $\Im(z)$ denotes its imaginary part. We denote by $|\boldsymbol{x}|$ the Euclidean norm of a vector $\boldsymbol{x} \in \mathbb{R}^n$ and by $|\boldsymbol{A}|$ the 2-induced norm of a matrix $\boldsymbol{A} \in \mathbb{R}^{n\times m}$, but if s is a sequence or set, $|s|$ denotes its length or cardinality, respectively. The set $\mathbb{S}^n$ denotes the set of symmetric matrices in $\mathbb{R}^n$. For a symmetric matrix $\boldsymbol{P} \in \mathbb{S}^n$, we write $\boldsymbol{P} \succ \boldsymbol{0}$ ($\boldsymbol{P} \succeq \boldsymbol{0}$) if $\boldsymbol{P}$ is positive definite (semi-definite). For a set $\mathcal{X} \subseteq \Omega$, we denote by $\mathrm{cl}(\mathcal{X})$ its closure, $\partial\mathcal{X}$ its boundary, and $\bar{\mathcal{X}}$ its complement $\Omega \setminus \mathcal{X}$. We often use a string notation for sequences, e.g., $\sigma = abc$ reads $\sigma(1) = a, \sigma(2) = b, \sigma(3) = c$. Powers and concatenations work as expected, e.g., $\sigma^2 = \sigma\sigma = abcabc$. In particular, σ^ω denotes the infinite repetition of σ. For a relation $\mathcal{R} \subseteq \mathcal{X}_a \times \mathcal{X}_b$, its inverse is denoted as $\mathcal{R}^{-1} = \{(x_b, x_a) \in \mathcal{X}_b \times \mathcal{X}_a : (x_a, x_b) \in \mathcal{R}\}$. Finally, we denote by $\pi_{\mathcal{R}}(\mathcal{X}_a) := \{x_b \in \mathcal{X}_b \mid (x_a, x_b) \in \mathcal{R} \text{ for some } x_a \in \mathcal{X}_a\}$ the natural projection of $\mathcal{X}_a$ onto $\mathcal{X}_b$.

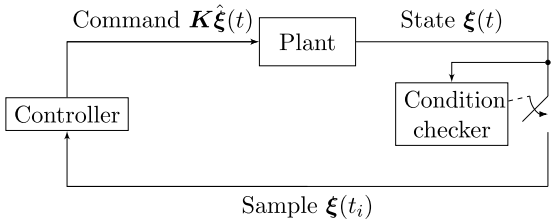

Fig. 1. Block diagram of an ETC system. In PETC, the condition checker is executed periodically.

2. Problem statement

Consider a linear time-invariant plant controlled with sample-and-hold state feedback [18] described by

$$\begin{aligned}\dot{\boldsymbol{\xi}}(t) &= \boldsymbol{A}\boldsymbol{\xi}(t) + \boldsymbol{B}\boldsymbol{\upsilon}(t),\\ \boldsymbol{\upsilon}(t) &= \boldsymbol{K}\hat{\boldsymbol{\xi}}(t),\end{aligned} \tag{1}$$

where $\boldsymbol{\xi}(t) \in \mathbb{R}^{n_\mathrm{x}}$ is the plant's state with initial value $\boldsymbol{x}_0 := \boldsymbol{\xi}(0)$, $\hat{\boldsymbol{\xi}}(t) \in \mathbb{R}^{n_\mathrm{x}}$ is the state measurement available to the controller, $\boldsymbol{\upsilon}(t) \in \mathbb{R}^{n_\mathrm{u}}$ is the control input, n_x and n_u are the state-space and input-space dimensions, respectively, and $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{K}$ are matrices of appropriate dimensions. The measurements are updated to the controller only at specific *sampling times*, with their values being zero-order held on the controller: let $t_i \in \mathbb{R}_+$, $i \in \mathbb{N}_0$ be a sequence of sampling times, with $t_0 = 0$ and $t_{i+1} - t_i > \varepsilon$ for some $\varepsilon > 0$; then $\hat{\boldsymbol{\xi}}(t) = \boldsymbol{\xi}(t_i)$, $\forall t \in [t_i, t_{i+1})$.

In ETC, a *triggering condition* determines the sequence of times t_i. In the case of PETC, this condition is checked only periodically, with a fundamental checking period h. Throughout this paper, we assume the time units have been scaled so that $h = 1$.[1] Fig. 1 depicts a simple diagram of a PETC system. We consider the family of *quadratic triggering conditions* from [6] with an additional maximum inter-sample time condition:

$$t_{i+1} = \inf\left\{k > t_i, k \in \mathbb{N} \;\middle|\; \begin{bmatrix}\boldsymbol{\xi}(k)\\ \boldsymbol{\xi}(t_i)\end{bmatrix}^\mathsf{T} \boldsymbol{Q} \begin{bmatrix}\boldsymbol{\xi}(k)\\ \boldsymbol{\xi}(t_i)\end{bmatrix} > 0 \;\text{ or }\; k - t_i \geq \bar{k}\right\}, \tag{2}$$

where $\boldsymbol{Q} \in \mathbb{S}^{2n_\mathrm{x}}$ is the designed triggering matrix, and $\bar{k}$ is the chosen maximum inter-sample time.[2] Observing Eq. (2), we note that the inter-sample time $t_{i+1} - t_i$ is a function of $\boldsymbol{x}_i := \boldsymbol{\xi}(t_i)$; denoting $\kappa := (t_{i+1} - t_i)$ as the inter-sample time, it follows that

$$\begin{aligned}\kappa(\boldsymbol{x}_i) &= \min\left\{k \in \{1, 2, \dots \bar{k}\} \mid \boldsymbol{x}_i^\mathsf{T}\boldsymbol{N}(k)\boldsymbol{x}_i > 0 \text{ or } k = \bar{k}\right\},\\ \boldsymbol{N}(k) &:= \begin{bmatrix}\boldsymbol{M}(k)\\ \mathbf{I}\end{bmatrix}^\mathsf{T} \boldsymbol{Q} \begin{bmatrix}\boldsymbol{M}(k)\\ \mathbf{I}\end{bmatrix},\\ \boldsymbol{M}(k) &:= \boldsymbol{A}_\mathrm{d}(k) + \boldsymbol{B}_\mathrm{d}(k)\boldsymbol{K} := \mathrm{e}^{\boldsymbol{A}hk} + \int_0^{hk} \mathrm{e}^{\boldsymbol{A}\tau}\mathrm{d}\tau\boldsymbol{B}\boldsymbol{K}.\end{aligned} \tag{3}$$

where $\mathbf{I}$ denotes the identity matrix. Thus, the event-driven evolution of sampled states can be compactly described by the recurrence

$$\boldsymbol{\xi}(t_{i+1}) = \boldsymbol{M}(\kappa(\boldsymbol{\xi}(t_i)))\boldsymbol{\xi}(t_i). \tag{4}$$

Clearly, each initial condition $\boldsymbol{x}_0 \in \mathbb{R}^{n_\mathrm{x}}$ leads to infinite sequences of samples $\{\boldsymbol{x}_i\}$ and inter-sample times $\{k_i(\boldsymbol{x}_0)\}$, defined recursively as

$$\begin{aligned}\boldsymbol{x}_{i+1} &= \boldsymbol{M}(\kappa(\boldsymbol{x}_i))\boldsymbol{x}_i\\ k_i(\boldsymbol{x}_0) &:= \kappa(\boldsymbol{x}_i).\end{aligned} \tag{5}$$

Therefore, we can attribute an *average inter-sample time* (AIST) to every initial state:

$$\mathrm{AIST}(\boldsymbol{x}) := \liminf_{n\to\infty} \frac{1}{n+1}\sum_{i=0}^{n} k_i(\boldsymbol{x}).$$

[1] This time re-scaling can be achieved by simply multiplying $\boldsymbol{A}$ and $\boldsymbol{B}$ with h.

[2] Typically, a maximum inter-sample time exists naturally for a system with (P)ETC (see [19]). Still, one may want to set a smaller maximum inter-sample time so as to establish a "heart beat" of the system. In any case, this is a necessity if one wants to obtain a finite-state simulation of the system, which is what we do in this paper.

Using lim inf instead of lim lets us use the limit lower bound in case the regular limit does not exist, making the AIST metric well-defined.

Objective of this paper We want to devise a method to compute the exact *smallest average inter-sample time* (SAIST) of the PETC system (1)–(2); i.e., the minimal AIST across all possible initial conditions:

$$\text{SAIST} := \inf_{\boldsymbol{x}\in\mathbb{R}^{n_\mathrm{x}}} \liminf_{n\to\infty} \frac{1}{n+1}\sum_{i=0}^{n} hk_i(\boldsymbol{x}). \tag{6}$$

Furthermore, we want to understand the cases where the exact SAIST computation is not possible, and quantify the estimation error if the best we can obtain is an approximation.

The way we define SAIST implies that we do not expect that a system's AIST is irrespective of its initial conditions; as we shall see later in Section 6, it is possible that multiple AISTs are observed. Hence, in these cases, we conservatively take the smallest possible one. We argue that the SAIST is an adequate – in fact, fundamental – metric to inform designers about the average resource utilization that an ETC implementation is expected to achieve. However, the mere application of Eq. (6) is largely unpromising: how can one choose a sufficiently large n, or how can one exhaustively search for initial states to obtain one that yields the SAIST? For this reason, we approach the SAIST computation problem through finite-state abstractions, which we introduce next.

3. Background and preliminary results

An *abstraction* is a simpler description of a system that preserves desired properties. When working with abstractions, we refer to the original system as the *concrete* system. In this paper, we work with *finite-state abstractions* using the framework of [12] and its *transition systems*. Later, we equip these systems with weights following [20], which allows us to derive metrics such as the SAIST. We then present a special type of finite-state abstraction that preserves SAIST, which we introduced in [7].

3.1. Transition systems and abstractions

In [12], Tabuada presents the notion of generalized transition systems:

Definition 1 (*Transition System [12]*)**.** A system $\mathcal{S}$ is a tuple $(\mathcal{X}, \mathcal{X}_0, \mathcal{E}, \mathcal{Y}, H)$ where:

- $\mathcal{X}$ is the set of states,
- $\mathcal{X}_0 \subseteq \mathcal{X}$ is the set of initial states,
- $\mathcal{E} \subseteq \mathcal{X} \times \mathcal{X}$ is the set of edges, or transitions,
- $\mathcal{Y}$ is the set of outputs, and
- $H : \mathcal{X} \to \mathcal{Y}$ is the output map.

Here we have omitted the action set $\mathcal{U}$ from the original definition because we are solely interested in autonomous systems like (5). A system is said to be *finite-state* (*infinite-state*) if the cardinality of $\mathcal{X}$ is finite (infinite). System $\mathcal{S}$ is said to be *non-blocking* if $\forall x \in \mathcal{X}, \exists x' \in \mathcal{X} : (x, x') \in \mathcal{E}$. We call $x_0x_1x_2\ldots$ an *infinite internal behavior*, or *run* of $\mathcal{S}$ if $x_0 \in \mathcal{X}_0$ and $(x_i, x_{i+1}) \in \mathcal{E}$ for all $i \in \mathbb{N}$, and $y_0y_1\ldots$ its corresponding *infinite external behavior*, or *trace*, if $H(x_i) = y_i$ for all $i \in \mathbb{N}$. We denote by $B_{\mathcal{S}}(r)$ the external behavior from a run $r = x_0x_1\ldots$ (in the case above, $B_{\mathcal{S}}(r) = y_0y_1\ldots$), by $\mathcal{B}^{\omega}_{x}(\mathcal{S})$ the set of all infinite external behaviors of $\mathcal{S}$ starting from state x, and by $\mathcal{B}^{\omega}(\mathcal{S}) := \bigcup_{x\in\mathcal{X}_0} \mathcal{B}^{\omega}_{x}(\mathcal{S})$ the set of all infinite external behaviors of $\mathcal{S}$. Finally, $\mathcal{B}^{\leq n}(\mathcal{S})$ is the set of all prefixes of length $\leq n$ of each trace in $\mathcal{B}^{\omega}(\mathcal{S})$ (equivalently, the set of its $(\leq n)$-long external behaviors), and $\mathcal{B}^{+}(\mathcal{S})$ is the set of all finite prefixes in $\mathcal{B}^{\omega}(\mathcal{S})$. A finite sequence β is called *transient* if there exists a finite l such that $\gamma\beta\alpha \in \mathcal{B}^{\omega}(\mathcal{S})$ implies that $|\gamma| \leq l$ and β is not a subsequence of α; equivalently, β cannot occur infinitely often in any infinite behavior of $\mathcal{S}$.

The ideas of simulation and bisimulation are paramount to establish formal relations between two transition systems.

Definition 2 (*Simulation Relation [12]*)**.** Consider two transition systems $\mathcal{S}_a$ and $\mathcal{S}_b$ with $\mathcal{Y}_a = \mathcal{Y}_b$. A relation $\mathcal{R} \subseteq \mathcal{X}_a \times \mathcal{X}_b$ is a simulation relation from $\mathcal{S}_a$ to $\mathcal{S}_b$ if the following conditions are satisfied:

(i) for every $x_{a0} \in \mathcal{X}_{a0}$, there exists $x_{b0} \in \mathcal{X}_{b0}$ with $(x_{a0}, x_{b0}) \in \mathcal{R}$;
(ii) for every $(x_a, x_b) \in \mathcal{R}$, $H_a(x_a) = H_b(x_b)$;
(iii) for every $(x_a, x_b) \in \mathcal{R}$, we have that $(x_a, x'_a) \in \mathcal{E}_a$ implies the existence of $(x_b, x'_b) \in \mathcal{E}_b$ satisfying $(x'_a, x'_b) \in \mathcal{R}$.

When there exists a simulation relation from $\mathcal{S}_a$ to $\mathcal{S}_b$, we say that $\mathcal{S}_b$ simulates $\mathcal{S}_a$, denoted by $\mathcal{S}_a \preceq \mathcal{S}_b$. When $\mathcal{R}$ is a simulation relation from $\mathcal{S}_a$ to $\mathcal{S}_b$ and $\mathcal{R}^{-1}$ is a simulation relation from $\mathcal{S}_b$ to $\mathcal{S}_a$, we say that $\mathcal{S}_a$ and $\mathcal{S}_b$ are *bisimilar*, denoted by $\mathcal{S}_a \cong \mathcal{S}_b$. Weaker but important relations associated with simulation and bisimulation are, respectively, *behavioral inclusion* and *behavioral equivalence*:

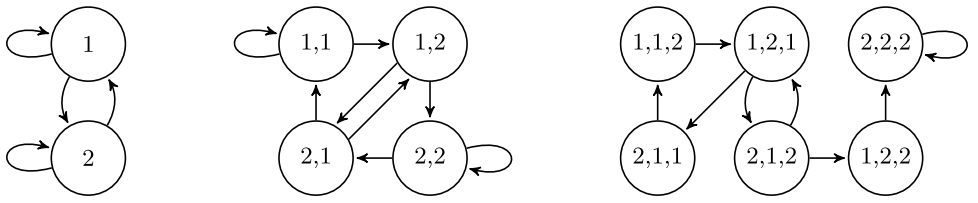

Fig. 2. Example of l-complete PETC traffic models, for $l = 1$ (left), $l = 2$ (middle), and $l = 3$ (right).

Definition 3 (*Behavioral Inclusion and Equivalence [12]*)**.** Consider two systems $\mathcal{S}_a$ and $\mathcal{S}_b$ with $\mathcal{Y}_a = \mathcal{Y}_b$. We say that $\mathcal{S}_a$ is *behaviorally included* in $\mathcal{S}_b$, denoted by $\mathcal{S}_a \preceq_{\mathcal{B}} \mathcal{S}_b$, if $\mathcal{B}^\omega(\mathcal{S}_a) \subseteq \mathcal{B}^\omega(\mathcal{S}_b)$. In case $\mathcal{B}^\omega(\mathcal{S}_a) = \mathcal{B}^\omega(\mathcal{S}_b)$, we say that $\mathcal{S}_a$ and $\mathcal{S}_b$ are *behaviorally equivalent*, which is denoted by $\mathcal{S}_a \cong_{\mathcal{B}} \mathcal{S}_b$.

(Bi)simulations imply behavioral inclusion (equivalence):

Theorem 1 (*[12]*)**.** *Given two systems $\mathcal{S}_a$ and $\mathcal{S}_b$ with $\mathcal{Y}_a = \mathcal{Y}_b$:*

- $\mathcal{S}_a \preceq \mathcal{S}_b \implies \mathcal{S}_a \preceq_{\mathcal{B}} \mathcal{S}_b$;
- $\mathcal{S}_a \cong \mathcal{S}_b \implies \mathcal{S}_a \cong_{\mathcal{B}} \mathcal{S}_b$.

The main difference between simulation and behavioral inclusion is that, in the former, a relationship between states must be established: every transition in the concrete system must have at least one matching transition in the abstraction leading to related states. Behavioral inclusion is oblivious to state-based descriptions of a system: all one needs is that all traces observed in the concrete system can also be observed in the abstraction. A way of building an abstraction based on behavioral inclusion is through an l-complete model:

Definition 4 (*Strongest l-Complete Abstraction (Adapted from [16,17])*)**.** Let $\mathcal{S} := (\mathcal{X}, \mathcal{X}_0, \mathcal{E}, \mathcal{Y}, H)$ be a transition system and $\mathcal{X}_l \subseteq \mathcal{Y}^l$ be the set of all l-long subsequences of all behaviors in $\mathcal{B}^\omega(\mathcal{S})$. The system $\mathcal{S}_l = (\mathcal{X}_l, \mathcal{B}^l(\mathcal{S}), \mathcal{E}_l, \mathcal{Y}, H)$ is called the *strongest l-complete abstraction (SlCA)* of $\mathcal{S}$, where

- $\mathcal{E}_l = \{(k\sigma, \sigma k') \mid k, k' \in \mathcal{Y}, \sigma \in \mathcal{Y}^{l-1}, k\sigma, \sigma k' \in \mathcal{X}_l\}$.
- $H(k\sigma) = k$.

The idea behind the SlCA is to encode the states as the l-long behavior fragments of the concrete system. The transitions follow the "*domino rule*": e.g., if the last 4 elements of the behavior up to a given time are *abcd*, after one step the first 3 elements must be *bcd*; thus, from having observed *abcd* alone, a transition from state *abcd* can lead to any state starting with *bcd*. Finally, the output of a state is its first element. To illustrate how successive l-complete approximations of a system operate, consider a system $\mathcal{S}$ with $\mathcal{Y} = \{1, 2\}$ with behavior set $\{2^\omega, 12^\omega, 212^\omega, (112)^\omega, (121)^\omega, (211)^\omega\}$. Fig. 2 presents its 1-, 2-, and 3-complete abstractions; as one can see, $\mathcal{S}_1$ and $\mathcal{S}_2$ have the trivial set of all possible behaviors over the set $\mathcal{Y}$, but $\mathcal{B}^\omega(\mathcal{S}_3)$ is smaller, closer to the concrete behavior set. That is, we have $\mathcal{B}^\omega(\mathcal{S}) \subseteq \mathcal{B}^\omega(\mathcal{S}_3) \subseteq \mathcal{B}^\omega(\mathcal{S}_2) \subseteq \mathcal{B}^\omega(\mathcal{S}_1)$, and in this example $\mathcal{S}_3$ has fewer spurious behaviors than $\mathcal{S}_2$ and $\mathcal{S}_1$.

Remark 1. We have made an adaptation from the original definition from [16], where the system is defined on a behavioral framework [15]; here we present directly a realization of the SlCA as a transition system according to Definition 1. Schmuck et al. [17] showed that different realizations exist for the SlCA of a system, depending on whether you encode states based on past, future, or a mix of past and future observations. In Definition 4, we pick the one based on future observations, which simplifies the encoding (all states are l-long sequences without the need for "no-output yet" characters, see [16]), and is the tightest abstraction from a simulation relation perspective (see [17, Thm. 5]).

In [17, Theorem 9], it is concluded that a quotient-based approach [12] can create an abstraction bisimilar to the SlCA in case the concrete system is future-unique, which is the case of deterministic systems. Thus, we shall use the term l-complete for quotient-based abstractions whose states represent the next l outputs of their related concrete states. How to do it will become clear in Section 5.1, where we build the abstractions of the PETC traffic model. With this in mind, the following fact is a direct consequence of Theorems 6 and 7 from [17].

Proposition 1. *Consider a deterministic system $\mathcal{S}$ and its SlCA $\mathcal{S}_l$ from Definition 4, for some $l \geq 1$. Then, $\mathcal{S} \preceq \mathcal{S}_{l+1} \preceq \mathcal{S}_l$.*

Proposition 1 gives that l-complete abstractions provide a framework of obtaining simulations and their refinements. It is not a surprising result, since encoding states with more elements of the concrete system's behavior constrains the set of behaviors it can generate, even if it increases the number of states in the abstraction.

Remark 2. Bisimulation is obtained when $\mathcal{S}_{l+1} = \mathcal{S}_l$ (modulo the names of the states); it is trivial to see that this only happens when abstracting an autonomous deterministic system if the abstraction is deterministic. In addition, $\lim_{l\to\infty} \mathcal{S}_l \cong_{\mathcal{B}} \mathcal{S}$.

3.2. Quantitative automata

While much of the field of formal methods in control is concerned with qualitative analysis, such as establishing safety, stability, and reachability, often quantitative computations are of interest: examples are computing the decay rate, the maximum overshoot, or our case, the average sampling period of an ETC system. In [20], Chatterjee et al. established a comprehensive framework for quantitative problems on finite-state systems, from which we borrow some definitions and results, while adjusting notation to keep consistency with the previous section.

Definition 5 (*Weighted Transition System (Adapted from [20])*)**.** A *weighted transition system* (WTS) $\mathcal{S}$ is the tuple $(\mathcal{X}, \mathcal{X}_0, \mathcal{E}, \mathcal{Y}, H, \gamma)$, where

- $(\mathcal{X}, \mathcal{X}_0, \mathcal{E}, \mathcal{Y}, H)$ is a *non-blocking* transition system;
- $\gamma : \mathcal{E} \to \mathbb{Q}$ is the *weight function*.

The notation adjustment we have made is including outputs to comply with Tabuada's transition systems; again, we ignore the action set as our scope is limited to autonomous systems.

Given a run $r = x_0x_1...$ of $\mathcal{S}$, we abuse notation denoting by $\gamma(r) = v_0v_1...$ the *sequence of weights* defined by $v_i = \gamma(x_i, x_{i+1})$. A *value function* $\mathrm{Val} : \mathbb{Q}^\omega \to \mathbb{R}$ attributes a value to an infinite sequence of weights $v_0v_1....$ Among the well-studied value functions, the one of our interest is

$$\mathrm{LimAvg}(v) := \liminf_{n\to\infty} \frac{1}{n+1} \sum_{i=0}^{n} v_i.$$

Similarly, for a finite sequence v of length n, let $\mathrm{Avg}(v) := \frac{1}{n+1}\sum_{i=0}^{n} v_i$. We define the smallest and largest LimAvg values of an automaton respectively as $\underline{\mathrm{V}}(\mathcal{S}) := \inf\{\mathrm{LimAvg}(\gamma(r)) \mid r \text{ is a run of } \mathcal{S}\}$ and $\overline{\mathrm{V}}(\mathcal{S}) := \sup\{\mathrm{LimAvg}(\gamma(r)) \mid r \text{ is a run of } \mathcal{S}\}$. Clearly $\underline{\mathrm{V}}(\mathcal{S}) = -\overline{\mathrm{V}}(-\mathcal{S})$, where we denote by $-\mathcal{S}$ the WTS $\mathcal{S}$ with all of its weights negated; thus, we focus on the results for $\underline{\mathrm{V}}$ in what follows. The following theorem is essentially an excerpt from Theorem 3 in [20], which uses the classical result from Karp [14]:

Theorem 2. *Given a finite-state WTS $\mathcal{S}$, $\underline{\mathrm{V}}(\mathcal{S})$ can be computed in $\mathcal{O}(|\mathcal{X}||\mathcal{E}|)$. Moreover, system $\mathcal{S}$ admits a cycle $x_0x_1...x_k$ satisfying $x_i \to x_{i+1}$, $i < k$, and $x_k \to x_0$ s.t. $\mathrm{LimAvg}(\gamma((x_0x_1...x_k)^\omega)) = \underline{\mathrm{V}}(\mathcal{S})$.*

The cycle mentioned above is a *smallest-in-average cycle* (SAC) of the weighted digraph defined by $\mathcal{S}$, and can be recovered in $\mathcal{O}(|\mathcal{X}|)$ using the procedure of [21].

3.3. Quantitative verification through abstractions

In [7], we have presented some basic results about the relationship between the SAISTs of a system and its abstraction. First, we start with a simplifying condition for weighted transition systems: a WTS is called *simple* if for all $(x, x') \in \mathcal{E}$, $\gamma(x, x') = H(x)$, i,e., the weight of a transition is equal to the output of its outbound state. Throughout this paper, when working with a transition system with $\mathcal{Y} \subset \mathbb{Q}$, we omit the weight function γ, implying that we have a simple WTS. Here, we recall results from [7].

Proposition 2 (*[7]*)**.** *If two simple WTSs $\mathcal{S}_a$ and $\mathcal{S}_b$ satisfy $\mathcal{S}_a \preceq_{\mathcal{B}} (\cong_{\mathcal{B}}) \mathcal{S}_b$, then $\underline{\mathrm{V}}(\mathcal{S}_a) \geq (=) \underline{\mathrm{V}}(\mathcal{S}_b)$ and $\overline{\mathrm{V}}(\mathcal{S}_a) \leq (=) \overline{\mathrm{V}}(\mathcal{S}_b)$.*

Proof. Since the systems are simple, $\mathcal{V}(\mathcal{S}_s) = \mathcal{B}^\omega(\mathcal{S}_s), s \in \{a, b\}$. Thus, $\overline{\mathrm{V}}(\mathcal{S}_s) = \inf\{\mathrm{LimAvg}(\{y_i\}) \mid \{y_i\} \in \mathcal{V}(\mathcal{S}_s)\} = \inf\{\mathrm{LimAvg}(\{y_i\}) \mid \{y_i\} \in \mathcal{B}^\omega(\mathcal{S}_s)\}$. Since $\mathcal{B}^\omega(\mathcal{S}_a) \subseteq (=) \mathcal{B}^\omega(\mathcal{S}_b)$, and the inferior of a function on a set can only be smaller than that of a set contained in it, we have $\underline{\mathrm{V}}(\mathcal{S}_a) \geq (=) \underline{\mathrm{V}}(\mathcal{S}_b)$. For $\overline{\mathrm{V}}$, the same reasoning is applied symmetrically. □

From Proposition 2, abstractions that simulate the concrete system provide a way to underestimate the SAIST and overestimate the LAIST, thanks to Theorem 2. Equality can be achieved with the following type of abstraction.

Definition 6 (*Smallest-Average-Cycle-Equivalent Simulation [7]*)**.** Consider two simple WTSs $\mathcal{S}_a$ and $\mathcal{S}_b$ satisfying $\mathcal{S}_a \preceq \mathcal{S}_b$. Let $\mathrm{SAC}(\mathcal{S}_b)$ be the set of smallest-in-average cycles of $\mathcal{S}_b$. If there exists a behavior of the form $dc^\omega \in \mathcal{B}^\omega(\mathcal{S}_a)$ where d is finite and $c \in \mathrm{SAC}(\mathcal{S}_b)$, then $\mathcal{S}_b$ is a *smallest-average-cycle-equivalent (SACE) simulation* of $\mathcal{S}_a$.

A SACE simulation is a normal simulation with the added requirement that at least one of the SACs of the abstraction is an actual recurrent behavior of the concrete system, after some finite transient. Clearly, SACE simulation is stronger than simulation but significantly weaker than bisimulation. Equivalently, a *largest-average-cycle-equivalent simulation*, or LACE simulation, can be defined using the maximum average cycle instead. The following result is a straightforward conclusion from Proposition 2 and Theorem 2.

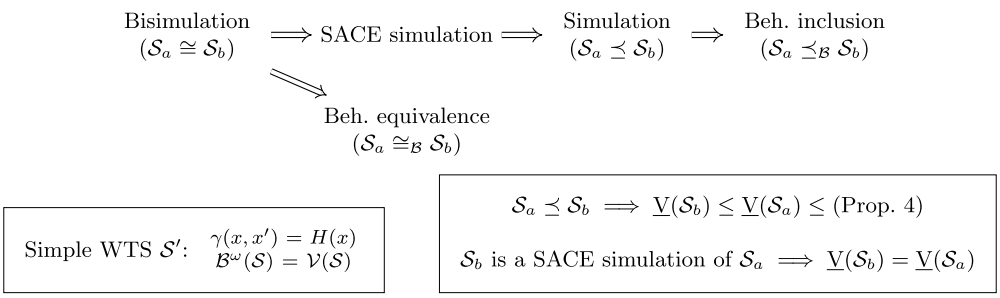

Fig. 3. Summary of relations and results in Section 3. Double-arrows ($\implies$) mean implication.

Proposition 3 (*[7]*).[3] *Consider two simple WTSs $\mathcal{S}_a$ and $\mathcal{S}_b$; if $\mathcal{S}_b$ is a finite-state SACE simulation of $\mathcal{S}_a$, then $\underline{\mathrm{V}}(\mathcal{S}_a) = \underline{\mathrm{V}}(\mathcal{S}_b)$.*

Remark 3. In fact, to use Definition 6 and Proposition 3, it is not needed that the WTSs are simple. One can always turn a WTS into an equivalent simple one by adding artificial states: suppose that (x, y) and (x, z) belong to $\mathcal{E}$ and $\gamma(x, y) = a \neq \gamma(x, z) = b$. Add artificial states y' and z' and replace the aforementioned transitions with (x, y'), (x, z'), (y', y), (z', z), setting $\gamma(x, y') = \gamma(x, z') = 0$, $\gamma(y', y) = a$ and $\gamma(z', z) = b$. Applying this procedure to the whole system gives a simple WTS, and again behaviors are equal to sequences of weights. The LimAvg value of any run of this modified system is half of the value of the original equivalent run (since we are adding zeros at every other transition).

For the cases where obtaining a SACE simulation of $\underline{\mathrm{V}}(\mathcal{S}_a)$ is not possible, one may still be interested in computing an estimate of the error $\underline{\mathrm{V}}(\mathcal{S}_a) - \underline{\mathrm{V}}(\mathcal{S}_b)$. In [7], the maximal value $\overline{\mathrm{V}}(\mathcal{S}_b)$ was used to this end, but a better approximation can be found by inspecting the maximal average cycle of the attractors of $\mathcal{S}_b$.

Proposition 4. *Let $\mathcal{S}_a := (\mathcal{X}_a, \mathcal{X}_a, \mathcal{E}_a, \mathcal{Y}, H_a)$ and $\mathcal{S}_b := (\mathcal{X}_b, \mathcal{X}_b, \mathcal{E}_b, \mathcal{Y}, H_b)$ be two simple WTSs, $\mathcal{R}$ be a simulation relation from $\mathcal{S}_a$ to $\mathcal{S}_b$, and $\mathcal{A} \subseteq \mathcal{X}_b$ be a strongly forward invariant set[4] of $\mathcal{S}_b$. If there exists $x_b \in \mathcal{A}$ such that $(x_a, x_b) \in \mathcal{R}$ for some $x_a \in \mathcal{X}_a$, then $\underline{\mathrm{V}}(\mathcal{S}_a) \leq \overline{\mathrm{V}}((\mathcal{A}, \mathcal{A}, \mathcal{E}_b, \mathcal{Y}, H_b)) \leq \overline{\mathrm{V}}(\mathcal{S}_b)$.*

Proof. First, it is a simple exercise to see that $(\mathcal{X}, \mathcal{X}', \mathcal{E}, \mathcal{Y}, H) \preceq (\mathcal{X}, \mathcal{X}, \mathcal{E}, \mathcal{Y}, H)$ if $\mathcal{X}' \subseteq \mathcal{X}$. Now, take $(x_a, x_b) \in \mathcal{R}$ where $x_b \in \mathcal{A}$. Then, $(\mathcal{X}_a, \{x_a\}, \mathcal{E}_a, \mathcal{Y}, H_a) \preceq \mathcal{S}_a$. At the same time, with the same relation $\mathcal{R}$ we can verify that $(\mathcal{X}_a, \{x_a\}, \mathcal{E}_a, \mathcal{Y}, H_a) \preceq (\mathcal{X}_b, \{x_b\}, \mathcal{E}_b, \mathcal{Y}, H_b)$. Therefore, by Proposition 2, $\underline{\mathrm{V}}((\mathcal{X}_a, \{x_a\}, \mathcal{E}_a, \mathcal{Y}, H_a)) \geq \underline{\mathrm{V}}(\mathcal{S}_a)$, and $\overline{\mathrm{V}}((\mathcal{X}_b, \{x_b\}, \mathcal{E}_b, \mathcal{Y}, H_b)) \geq \overline{\mathrm{V}}((\mathcal{X}_a, \{x_a\}, \mathcal{E}_a, \mathcal{Y}, H_a))$. Because $\overline{\mathrm{V}}(\cdot) \geq \underline{\mathrm{V}}(\cdot)$, we get that $\overline{\mathrm{V}}((\mathcal{X}_b, \{x_b\}, \mathcal{E}_b, \mathcal{Y}, H_b)) \geq \underline{\mathrm{V}}(\mathcal{S}_a)$.

Now, because $\mathcal{A}$ is strongly forward invariant, every run of $(\mathcal{X}_b, \{x_b\}, \mathcal{E}_b, \mathcal{Y}, H_b)$ contains only states in $\mathcal{A}$. Thus, $(\mathcal{X}_b, \{x_b\}, \mathcal{E}_b, \mathcal{Y}, H_b) \cong_{\mathcal{B}} (\mathcal{A}, \{x_b\}, \mathcal{E}_b, \mathcal{Y}, H_b) \preceq (\mathcal{A}, \mathcal{A}, \mathcal{E}_b, \mathcal{Y}, H_b)$. Then, applying Proposition 2 again gives $\overline{\mathrm{V}}((\mathcal{A}, \mathcal{A}, \mathcal{E}_b, \mathcal{Y}, H_b)) \geq \underline{\mathrm{V}}(\mathcal{S}_a)$.

Finally, because $(\mathcal{A}, \mathcal{A}, \mathcal{E}_b, \mathcal{Y}, H_b) \preceq \mathcal{S}_b$, Proposition 2 also gives that $\overline{\mathrm{V}}(\mathcal{S}_b) \geq \overline{\mathrm{V}}((\mathcal{A}, \mathcal{A}, \mathcal{E}_b, \mathcal{Y}, H_b))$. □

When the abstraction $\mathcal{S}_b$ is finite, its smallest strongly invariant sets are simply the attractive strongly connected components (SCCs) of the graph associated with $\mathcal{S}_b$. Obtaining the SCCs of a graph with n vertices and m edges has complexity $\mathcal{O}(n + m)$ [14] and in fact is part of the steps to compute its smallest (or largest) average cycle.

Fig. 3 summarizes the main concepts and preliminary results of this section.

4. Limit average from *l*-complete abstractions

In this section we provide some results on the computation of the infimal limit average of a simple WTS $\mathcal{S}$ through the use of its *SlCA* $\mathcal{S}_l$. The first result is an obvious conclusion from combining Proposition 1 with 2:

Proposition 5. *Consider a simple WTS $\mathcal{S}$ and its SlCA $\mathcal{S}_l$ (Definition 4), for some $l \geq 1$. It holds that $\underline{\mathrm{V}}(\mathcal{S}_l) \leq \underline{\mathrm{V}}(\mathcal{S})$.*

Considering the idea of SACE simulation, a simple conceptual algorithm that can compute the exact value of $\underline{\mathrm{V}}(\mathcal{S})$ is given in Alg. 1. The idea is to increment l until the smallest-in-average cycle of $\mathcal{S}_l$ is verified in the concrete system. The algorithm requires one to be able to compute the *SlCA* of a given system (line 3) and to verify the existence of periodic behavior (line 5); these steps will be discussed for PETC traffic on Section 5. As we will see now, Alg. 1 is in fact a semi-algorithm; depending on the behavior of $\mathcal{S}$, it may not terminate. The following result shows under which conditions there is a finite l such that $\underline{\mathrm{V}}(\mathcal{S}_l) = \underline{\mathrm{V}}(\mathcal{S})$.

[3] The concept of SACE simulation and its related results have been recently expanded to non-autonomous WTSs in [22], where the objective was to *design* sampling strategies instead of evaluating them. This expansion is not needed for the scope of this paper.

[4] A strongly forward invariant set $\mathcal{A} \subseteq \mathcal{X}$ is a set that satisfies $\forall x \in \mathcal{A}, (x, x') \in \mathcal{E} \implies x' \in \mathcal{A}$.

Algorithm 1 Computation of $\underline{\mathrm{V}}(\mathcal{S})$

Input: A simple WTS $\mathcal{S}$ with $\mathcal{Y} \subset \mathbb{Q}$, $|\mathcal{Y}| < \infty$
Output: l, $\mathcal{S}_l$, $\mathtt{V}$, σ

1: $l \leftarrow 1$
2: **while** true **do**
3: Build $\mathcal{S}_l$ ▷ (Definition 4)
4: $\mathtt{V} \leftarrow \underline{\mathrm{V}}(\mathcal{S}_l)$, $\sigma \leftarrow \mathrm{SAC}(\mathcal{S}_l)$ ▷ [14,21]
5: **if** $\sigma^\omega \in \mathcal{B}^\omega(\mathcal{S})$ **then**
6: **return**
7: **end if**
8: $l \leftarrow l + 1$
9: **end while**

Theorem 3. *Consider a simple finite WTS $\mathcal{S}$ and assume that there exists a finite $m \in \mathbb{N}$ such that every infinite behavior $\alpha \in \mathcal{B}^\omega(\mathcal{S})$ satisfies $\mathrm{Avg}(\beta) \geq \underline{\mathrm{V}}(\mathcal{S})$, for every non-transient subsequence β of α with $|\beta| = m$. Then there exists a finite l such that the l-complete simulation $\mathcal{S}_l$ of $\mathcal{S}$ satisfies $\underline{\mathrm{V}}(\mathcal{S}_l) = \underline{\mathrm{V}}(\mathcal{S})$.*

Proof. First we prove that, if β is transient, then there exists l large enough such that β cannot be a subsequence of σ^ω for any cycle σ of $\mathcal{S}_l$. For that, suppose by contradiction that, $\forall L, \exists l \geq L$ for which a cycle σ of $\mathcal{S}_l$ exists s.t. β *is* a subsequence of σ^ω; w.l.o.g., assume that $l > m$. Then, there exists a word $\gamma\beta$ of length l that is a subsequence of σ^ω; hence, $|\gamma| = l - m$. This holds because for any natural number p, $\sigma^p\beta$ is a subsequence of $\sigma^p\sigma^\omega = \sigma^\omega$. Now, by definition of $\mathcal{S}_l$, $\gamma\beta \in \mathcal{B}^l(\mathcal{S})$. Since l can be chosen arbitrarily large, β can occur arbitrarily late in a behavior of $\mathcal{S}$, thus contradicting the fact that it is transient.

Therefore, there exists l large enough such that, for every cycle σ of $\mathcal{S}_l$, every m-long subsequence β of σ^ω is non-transient. From Theorem 2, one such cycle satisfies $\underline{\mathrm{V}}(\mathcal{S}_l) = \mathrm{LimAvg}(\sigma^\omega)$. Let $p := |\sigma|$. Then, σ^m has length pm and as such it can be divided in p non-transient subsequences β_i, not necessarily distinct, of length m. Now,

$$\underline{\mathrm{V}}(\mathcal{S}_l) = \mathrm{LimAvg}(\sigma^\omega) = \mathrm{LimAvg}((\sigma^m)^\omega) = \mathrm{Avg}(\sigma^m) = \frac{1}{p}\sum_{i=1}^{p} \mathrm{Avg}(\beta_i) \geq \underline{\mathrm{V}}(\mathcal{S}).$$

Since, by Proposition 5, $\underline{\mathrm{V}}(\mathcal{S}_l) \leq \underline{\mathrm{V}}(\mathcal{S})$, it holds that $\underline{\mathrm{V}}(\mathcal{S}_l) = \underline{\mathrm{V}}(\mathcal{S})$. □

Theorem 3 states that it is sufficient for it to exist an m large enough such that every "persistent" m-long behavior fragment β of $\mathcal{S}$ has higher or equal average than $\underline{\mathrm{V}}(\mathcal{S})$. Intuitively, constraining the assumption of β occurring infinitely often has the idea of excluding transient behaviors β, which do not affect the LimAvg value. For cases where β can occur infinitely often in some behavior, but β^ω is not a behavior of $\mathcal{S}$, one can construct counterexamples in which $\underline{\mathrm{V}}(\mathcal{S}_l) < \underline{\mathrm{V}}(\mathcal{S})$ for all l:

Example 1. Consider a system $\mathcal{S}$ with behavior set $\mathcal{B}^\omega(\mathcal{S}) = \{(1^n2^n)^\omega \mid n \in \mathbb{N}\}$. Obviously, $\underline{\mathrm{V}}(\mathcal{S}) = 1.5$. However, for any l, $(1^l)^\omega \in \mathcal{B}^\omega(\mathcal{S}_l)$, hence $\underline{\mathrm{V}}(\mathcal{S}_l) = 1$ for any l.

Example 2. Consider the system $\mathcal{S} = ([0, 1], [0, 1], \mathcal{E}, \{0, 1\}, H)$ where $\mathcal{E} = \{(x, x + a \bmod 1)\}$ and $H(x) = 1$ if $x < a$ and 0 otherwise. When a is irrational, $\mathcal{S}$ is called an irrational rotation. Because it is ergodic with respect to the Lebesgue measure [23], $\mathrm{LimAvg}(\alpha) = a$ for any $\alpha \in \mathcal{B}^\omega(\mathcal{S})$. Thus, $\underline{\mathrm{V}}(\mathcal{S}) = a$ is irrational. Since for every finite l, $\underline{\mathrm{V}}(\mathcal{S}_l)$ is a rational number (as a consequence of Theorem 2 and the fact that $\mathcal{S}_l$ is finite), $\underline{\mathrm{V}}(\mathcal{S}_l) \neq \underline{\mathrm{V}}(\mathcal{S})$. Finally, from Proposition 5, $\underline{\mathrm{V}}(\mathcal{S}_l) \leq \underline{\mathrm{V}}(\mathcal{S})$, thus $\underline{\mathrm{V}}(\mathcal{S}_l) < \underline{\mathrm{V}}(\mathcal{S})$ for all finite l.

Note that, for Example 2, the minimum number of 1s in a behavior fragment of length n is $\lfloor na \rfloor$, hence $\underline{\mathrm{V}}(\mathcal{S}_l) = \frac{\lfloor la \rfloor}{l}$, which asymptotically approaches a as l goes to infinity. For Example 1, we cannot obtain this asymptotic approximation.

The conditions in Theorem 3 do not imply that the SAC σ of $\mathcal{S}_l$ satisfies $\sigma^\omega \in \mathcal{B}^\omega(\mathcal{S})$; thus, we may have equality of LimAvg values without a SACE simulation. Therefore, under these conditions, Alg. 1 can be interrupted with the exact value, but with no certificate that this is the case. Its termination is guaranteed when there is a cyclic minimizing behavior, and additionally that the other behaviors have limit average values strictly larger than that of the cycle:

Theorem 4. *Consider a simple WTS $\mathcal{S}$, and suppose $\mathcal{S}$ satisfies the premises of Theorem 3. Furthermore, assume there exists an m-long sequence σ such that $\sigma^\omega \in \mathcal{B}^\omega(\mathcal{S})$, and that every non-transient subsequence β, $|\beta| = m$ of every behavior $\alpha \in \mathcal{B}^\omega(\mathcal{S})$ satisfies $\mathrm{LimAvg}(\beta^\omega) > \underline{\mathrm{V}}(\mathcal{S})$ if β is not a subsequence of σ^ω. Then Alg. 1 terminates with $\mathtt{V} = \underline{\mathrm{V}}(\mathcal{S})$.*

The proof requires some technical results on cyclic permutations of sequences and we leave it for the appendix. The main insight is that the conditions of Theorem 4 imply that, for sufficiently large l, $\mathcal{S}_l$ has only one SAC σ, modulo cyclic permutations, which attains the minimum value; at the same time, for large enough l, this σ satisfies $\sigma^\omega \in \mathcal{B}^\omega(\mathcal{S})$. Hereafter, we say that a system satisfying the premises of Theorem 4 has an *isolated SAC*. This does not mean that the behavior of $\mathcal{S}$ is simple, or that a finite-state bisimulation of it exists:

Example 3. Consider the *doubling map* system $\mathcal{S} = ([0,1],[0,1],\mathcal{E},\{0,1\},H)$ where $\mathcal{E} = \{(x, 2x \bmod 1) \mid x \in [0,1]\}$ and $H(x) = 0$ if $x < 1/2$ and 1 otherwise. The behavior of this system is $(0^+1^+)^\omega$, its smallest cycle is 0^ω with value zero (obtained with $x_0 = 0$). This system does not admit a finite-state bisimulation, but its 1-complete abstraction is $\mathcal{S}_1 = \{\{0,1\},\{0,1\},\{(0,0),(0,1),(1,0),(1,1)\},\{0,1\},\mathrm{Id}\}$, where Id is the identity operator. Clearly, $\mathcal{S}_1$ is a SACE simulation of $\mathcal{S}$ (in fact, it is behaviorally equivalent, but not bisimilar). The system $\mathcal{S}$ satisfies the premises of Theorem 4 with $m = 1$.

Now that we have the general framework for the computation of $\underline{\mathrm{V}}(\mathcal{S})$, we see how to apply it for PETC traffic.

5. Computing the SAIST of PETC

We start by describing the evolution of sampled states and ISTs of a PETC system, cf. Eq. (5), as a transition system following Definition 1:

$$\begin{aligned}\mathcal{S} :=& (\mathbb{R}^{n_\mathrm{x}}, \mathbb{R}^{n_\mathrm{x}}, \mathcal{E}, \mathcal{Y}, H), \text{ where}\\ &\mathcal{E} = \{(\boldsymbol{x}, \boldsymbol{x}') \in \mathbb{R}^{n_\mathrm{x}} \times \mathbb{R}^{n_\mathrm{x}} \mid \boldsymbol{x}' = \boldsymbol{M}(\kappa(\boldsymbol{x}))\boldsymbol{x}\},\\ &\mathcal{Y} = \{1, 2, \dots, \bar{k}\},\\ &H = \kappa.\end{aligned} \tag{7}$$

System $\mathcal{S}$ is our concrete infinite-state system, for which we want develop an algorithm like Alg. 1. For this we need to be able to (i) build an l-complete abstraction of the system, (ii) compute its SAC, and (iii) check if its minimum mean cycle exists in the concrete system. Naturally, Karp's algorithm [14,21] constitute the tool for task (ii). In the next section we present how to obtain l-complete abstractions of $\mathcal{S}$. Then, in Section 5.2, we show how can a cyclic behavior be verified to be trace of $\mathcal{S}$. Finally, we present the full algorithm and discuss its robustness and applicability in subsequent subsections.

5.1. l-complete PETC traffic models

As mentioned in Section 3.1, for autonomous deterministic systems such as $\mathcal{S}$ from Eq. (7), a quotient-based approach can be used to obtain its SlCA $\mathcal{S}_l$. The idea is to divide the state-space $\mathcal{X}$ into regions $\mathcal{X}_{y_1y_2\dots y_l}$, where the first l elements of any behavior in $\mathcal{B}^\omega_x(\mathcal{S})$, for any $x \in \mathcal{X}_{y_1y_2\dots y_l}$, are exactly $y_1, y_2, \dots, y_l$. If $\mathcal{S}$ is deterministic, this division generates a partition, as from one state x there exists only one infinite behavior. In [7,13], we have used this idea to construct finite-state PETC traffic models abstracting system (7), coming up with the following relation:

Definition 7 (*Inter-Sample Sequence Relation [7]*)**.** Given a sequence length l, we denote by $\mathcal{R}_l \subseteq \mathbb{R}^{n_\mathrm{x}} \times \mathcal{Y}^l$ the relation satisfying $(\boldsymbol{x}, k_1k_2\dots k_l) \in \mathcal{R}_l$ if and only if

$$\boldsymbol{x} \in \mathcal{Q}_{k_1}, \tag{8a}$$
$$\boldsymbol{M}(k_1)\boldsymbol{x} \in \mathcal{Q}_{k_2}, \tag{8b}$$
$$\boldsymbol{M}(k_2)\boldsymbol{M}(k_1)\boldsymbol{x} \in \mathcal{Q}_{k_3}, \tag{8c}$$
$$\vdots$$
$$\boldsymbol{M}(k_{l-1})\dots\boldsymbol{M}(k_1)\boldsymbol{x} \in \mathcal{Q}_{k_l}, \tag{8d}$$

where

$$\begin{aligned}\mathcal{Q}_k &:= \mathcal{K}_k \setminus \left(\bigcap_{j=\underline{k}}^{k-1} \mathcal{K}_j\right) = \mathcal{K}_k \cap \bigcap_{j=1}^{k-1} \bar{\mathcal{K}}_j,\\ \mathcal{K}_k &:= \begin{cases}\{\boldsymbol{x} \in \mathcal{X} | \boldsymbol{x}^\mathsf{T}\boldsymbol{N}(k)\boldsymbol{x} > 0\}, & k < \bar{k},\\ \mathbb{R}^{n_\mathrm{x}}, & k = \bar{k}.\end{cases}\end{aligned} \tag{9}$$

Eq. (9), from [11], defines the sets $\mathcal{Q}_k$, containing the states from which the next trigger happens exactly after k time units. Eq. (8) states that a state $\boldsymbol{x} \in \mathbb{R}^{n_\mathrm{x}}$ is related to a state $k_1k_2\dots k_l$ of the abstraction if its generated inter-sample time sequence for the next l samples is $k_1, k_2, \dots, k_l$.

Remark 4. Setting $l = 1$ gives a *quotient state set* [12] of $\mathcal{S}$ in (7), while larger values of l can be seen as refinements using the bisimulation algorithm of [12, Chapter 8].

Definition 8. Given an integer $l \geq 1$, the *l-complete PETC traffic model* is the system $\mathcal{S}_l := (\mathcal{X}_l, \mathcal{X}_l, \mathcal{E}_l, \mathcal{Y}, H_l)$, with

- $\mathcal{X}_l := \pi_{\mathcal{R}_l}(\mathcal{X})$,
- $\mathcal{E}_l = \{(k\sigma, \sigma k') \mid k, k' \in \mathcal{Y}, \sigma \in \mathcal{Y}^{l-1}, k\sigma, \sigma k' \in \mathcal{X}_l\}$,
- $H_l(k_1k_2...k_l) = k_1$.

The model above partitions the state-space $\mathbb{R}^{n_\mathrm{x}}$ of the PETC into subsets associated with the next l inter-sample times these states generate, i.e., it is an l-complete abstraction, but also a quotient-based model. Computing the state set, $\pi_{\mathcal{R}_l}(\mathcal{X})$, requires determining whether or not, for each $k_1k_2...k_l \in \mathcal{Y}^l$, its associated conjunction of quadratic inequalities in Eq. (8) admits a solution $\boldsymbol{x} \in \mathbb{R}^{n_\mathrm{x}}$; only if it does, then $\sigma \in \mathcal{X}_l$. This can be determined using a nonlinear satisfiability-modulo-theories (SMT) solver such as Z3 [24], see Remark 5[5] The output map H_l is the next sample alone, and the transition relation is based on the domino rule, as in Definition 4.

Remark 5. To verify if a sequence $\sigma := k_1k_2...k_l$ exists using an SMT solver, one solves the query $\exists \boldsymbol{x} \in \mathbb{R}^{n_\mathrm{x}}$: Eq. (8) holds. This requires unfolding the memberships of Eq. (8) into the conjunctions of quadratic inequalities by applying Eq. (9). For example, suppose we want to verify the sequence $\sigma = (3, 2)$. First we convert $\boldsymbol{x} \in \mathcal{Q}_3$, which is equivalent to $\boldsymbol{x}^\mathsf{T}\boldsymbol{N}(1)\boldsymbol{x} \leq 0$ and $\boldsymbol{x}^\mathsf{T}\boldsymbol{N}(2)\boldsymbol{x} \leq 0$ and $\boldsymbol{x}^\mathsf{T}\boldsymbol{N}(3)\boldsymbol{x} > 0$ if $3 < \bar{k}$, or just $\boldsymbol{x}^\mathsf{T}\boldsymbol{N}(1)\boldsymbol{x} \leq 0$ and $\boldsymbol{x}^\mathsf{T}\boldsymbol{N}(2)\boldsymbol{x} \leq 0$ if $3 = \bar{k}$. Then we add constraints associated to $\boldsymbol{M}(3)\boldsymbol{x} \in \mathcal{Q}_2$, which are $\boldsymbol{x}^\mathsf{T}\boldsymbol{M}(3)^\mathsf{T}\boldsymbol{N}(1)\boldsymbol{M}(3)\boldsymbol{x} \leq 0$ and $\boldsymbol{x}^\mathsf{T}\boldsymbol{M}(3)^\mathsf{T}\boldsymbol{N}(2)\boldsymbol{M}(3)\boldsymbol{x} > 0$, to the constraint set. Finally, because we are interested in non-trivial solutions, we say that $\boldsymbol{x}$ is non-zero by adding, e.g., $\boldsymbol{x}^\mathsf{T}\boldsymbol{x} > 0$ or $\boldsymbol{x}^\mathsf{T}\boldsymbol{x} = 1$. The final SMT query to check whether (3, 2) is a behavior of the system then becomes "$\exists \boldsymbol{x} \in \mathbb{R}^n$ such that $\boldsymbol{x}^\mathsf{T}\boldsymbol{N}(1)\boldsymbol{x} \leq 0$ and $\boldsymbol{x}^\mathsf{T}\boldsymbol{N}(2)\boldsymbol{x} \leq 0$ and $\boldsymbol{x}^\mathsf{T}\boldsymbol{N}(3)\boldsymbol{x} > 0$ and $\boldsymbol{x}^\mathsf{T}\boldsymbol{M}(3)^\mathsf{T}\boldsymbol{N}(1)\boldsymbol{M}(3)\boldsymbol{x} \leq 0$ and $\boldsymbol{x}^\mathsf{T}\boldsymbol{M}(3)^\mathsf{T}\boldsymbol{N}(2)\boldsymbol{M}(3)\boldsymbol{x} > 0$ and $\boldsymbol{x}^\mathsf{T}\boldsymbol{x} > 0$.

5.2. Verifying SACE equivalence

In this subsection, we are interested in determining whether a sequence of outputs $(k_1k_2...k_m)^\omega =: \sigma^\omega$ is a possible behavior of system $\mathcal{S}$ in Eq. (7). This is equivalent to finding a run $\{\boldsymbol{x}_i\}$ whose trace is σ^ω. From now on, we denote by $\mathcal{Q}_\sigma$, or σ-*cone*, the set of all points $\boldsymbol{x} \in \mathbb{R}^{n_\mathrm{x}}$ satisfying Eq. (8) with $l = m$ and by $\boldsymbol{M}_\sigma := \boldsymbol{M}(k_m)\boldsymbol{M}(k_{m-1})\cdots\boldsymbol{M}(k_1)$. For the formal results, consider the following classes of square matrices:

Definition 9 (*Mixed Matrix*)**.** Consider a matrix $\boldsymbol{M} \in \mathbb{R}^{n\times n}$ and let λ_i, $i \in \mathbb{N}_{\leq n}$ be its eigenvalues sorted such that $|\lambda_i| \geq |\lambda_{i+1}|$ for all i. We say that $\boldsymbol{M}$ is *mixed* if, for all $i < n$, $|\lambda_i| = |\lambda_{i+1}|$ implies that $\Im(\lambda_i) \neq 0$ and $\lambda_i = \lambda_{i+1}^*$.

Remark 6. Mixed matrices cannot have eigenvalues with the same magnitude, except for complex conjugate pairs. Every mixed matrix is diagonalizable, but the converse does not hold (e.g., the identity is not mixed). The set of mixed matrices is full Lebesgue measure. With a non-pathological choice of h,[6] the matrices $\boldsymbol{M}(1), \boldsymbol{M}(2), \ldots \boldsymbol{M}(\bar{k})$ from Eq. (3) are all mixed, even if $\boldsymbol{K}$ is chosen to place poles of $\boldsymbol{A}+\boldsymbol{BK}$ in the same point of the complex plane; it is sensible (but not guaranteed) to expect that their products are also mixed. From a linear systems perspective, all modes of a mixed matrix have different speeds.

Definition 10 (*Matrix of Irrational Rotations*)**.** A matrix $\boldsymbol{M} \in \mathbb{R}^{n\times n}$ is said to be of *irrational rotations* if the arguments of all of its complex eigenvalues are irrational multiples of π.

Remark 7. If $\boldsymbol{M}$ has a pair of complex conjugate eigenvalues whose argument is a rational multiple of π, i.e., $p\pi/q$, where $p, q \in \mathbb{N}$, then the corresponding eigenvalues of $\boldsymbol{M}^q$ are real. The set of real matrices of *rational* rotations is Lebesgue-measure zero but dense in $\mathbb{R}^{n\times n}$.

If $\boldsymbol{M}_\sigma$ is mixed and of irrational rotations, one can verify if σ^ω is a behavior of $\mathcal{S}$ from Eq. (7) by checking the linear invariants of $\boldsymbol{M}_\sigma$:

Theorem 5. *Consider system (7) and let $\sigma \in \mathcal{Y}^m$, $m \in \mathbb{N}$, be a sequence of outputs. (i) If $\boldsymbol{M}_\sigma$ is nonsingular and there exists a linear invariant $\mathcal{A}$ of $\boldsymbol{M}_\sigma$ such that $\mathcal{A} \setminus \{\boldsymbol{0}\} \subseteq \mathcal{Q}_\sigma$, then $\sigma^\omega \in \mathcal{B}^\omega(\mathcal{S})$. Moreover, if (ii) $\boldsymbol{M}_\sigma$ is additionally mixed and of irrational rotations, then $\sigma^\omega \in \mathcal{B}^\omega(\mathcal{S})$ implies that there exists a linear invariant $\mathcal{A}$ of $\boldsymbol{M}_\sigma$ such that $\mathcal{A} \subseteq \mathrm{cl}(\mathcal{Q}_\sigma)$.*

To avoid a long detour in our exposition, we leave the proof to the appendix, instead providing here a depiction of the idea behind it: In Fig. 4, we have $m = 1$ and $\mathcal{Y} = \{1, 2\}$, and the blue cone splits $\mathbb{R}^3$, the state space, in $\mathcal{Q}_1$ and $\mathcal{Q}_2$; the two

[5] Alternatively, this query may be solved approximately through convex relaxations as proposed in [11]. Using relaxations implies finding inter-sample sequences that may not be exhibited by the real system. This still generates a simulation relation, but containing more spurious behaviors.

[6] Typically, only countably many values of h will render $\boldsymbol{M}(k)$ non-mixed for a given k.

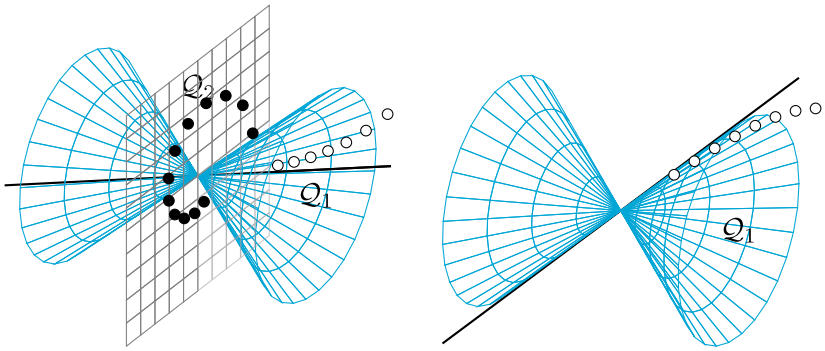

Fig. 4. Illustration of Theorem 5 in $\mathbb{R}^3$. The blue cone splits $\mathbb{R}^3$ into $\mathcal{Q}_1$ and $\mathcal{Q}_2$ the line is an invariant of $\boldsymbol{M}(1)$ and the plane is an invariant of $\boldsymbol{M}(2)$. Points indicate distinct sample trajectories $\{\boldsymbol{x}_i\}$.

plots have different matrices $\boldsymbol{M}(1)$. Runs $\{\boldsymbol{x}_i\}$ that generate the trace 1^ω are solutions of the linear system $\boldsymbol{x}_{i+1} = \boldsymbol{M}(1)\boldsymbol{x}_i$, one such example being depicted with white dots. Likewise, black dots show a run generating the trace 2^ω, and it has to be a solution of $\boldsymbol{x}_{i+1} = \boldsymbol{M}(2)\boldsymbol{x}_i$. In the example on the left, the black line is supported by one real eigenvector of $\boldsymbol{M}(1)$ and, as it belongs to $\mathcal{Q}_1$, at least solutions on top of this eigendirection are runs of the PETC system $\mathcal{S}$. In our example, this eigenvector is associated with a dominant mode of $\boldsymbol{M}(1)$, so solutions starting close to it converge towards it. The plane depicted on the left of Fig. 4 is an invariant of $\boldsymbol{M}(2)$ associated to complex conjugate eigenvalues. Solutions starting in this plane stay in this plane, spiraling towards the origin (in case the PETC implementation is stabilizing), which confirms that 2^ω is also a behavior of $\mathcal{S}$. The example on the right shows the defective case where the converse does not hold: for that, assume that $\mathcal{Q}_1$ does not include its depicted blue boundary; however, the black line representing an eigendirection of $\boldsymbol{M}(1)$ runs precisely on this boundary. In this example, the white dots represent a run $\{\boldsymbol{x}_i\}$ in $\mathcal{Q}_1$, thus generating the trace 1^ω, but no invariant of $\boldsymbol{M}(1)$ is a subset of $\mathcal{Q}_1$. Because the depicted mode of $\boldsymbol{M}(1)$ is dominant, there are solutions that start close to its associated eigendirection that stay in $\mathcal{Q}_1$ forever.

Based on Theorem 5, in the non-defective cases we can verify a cyclic behavior σ^ω by taking the finitely many linear invariants $\mathcal{A}$ of $\boldsymbol{M}_\sigma$ and checking if $\mathcal{A} \setminus \{\boldsymbol{0}\} \subseteq \mathcal{Q}_\sigma$, or, more explicitly, taking $\sigma = k_1 k_2 \ldots k_m$,

$$\begin{aligned} \mathcal{A} \setminus \{\boldsymbol{0}\} &\subseteq \mathcal{Q}_{k_1}, \\ \boldsymbol{M}(k_1)\mathcal{A} \setminus \{\boldsymbol{0}\} &\subseteq \mathcal{Q}_{k_2}, \\ &\vdots \\ \boldsymbol{M}(k_{m-1})\ldots\boldsymbol{M}(k_1)\mathcal{A} \setminus \{\boldsymbol{0}\} &\subseteq \mathcal{Q}_{k_m}. \end{aligned} \tag{10}$$

Because each $\mathcal{Q}_k$ is an intersection of quadratic sets (see Eq. (9)), we must be able to check whether a linear space is a subset of a given quadratic set, which is nothing but a positive-(semi)definiteness check:

Proposition 6 ([7]). *Let $\mathcal{A}$ be a linear subspace with basis $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_m$, and let $\boldsymbol{V}$ be the matrix composed of the vectors $\boldsymbol{v}_i$ as columns. Let $\boldsymbol{Q} \in \mathbb{S}^n$ be a symmetric matrix and define $\mathcal{Q}_n := \{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{x}^\mathsf{T}\boldsymbol{Q}\boldsymbol{x} \geq 0\}$ and $\mathcal{Q}_s := \{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{x}^\mathsf{T}\boldsymbol{Q}\boldsymbol{x} > 0\}$. Then, $\mathcal{A} \setminus \{\boldsymbol{0}\} \subseteq \mathcal{Q}_n$ (resp. $\mathcal{Q}_s$) if and only if $\boldsymbol{V}^\mathsf{T}\boldsymbol{Q}\boldsymbol{V} \succeq \boldsymbol{0}$ (resp. $\boldsymbol{V}^\mathsf{T}\boldsymbol{Q}\boldsymbol{V} \succ \boldsymbol{0}$).*

5.3. SACE simulation algorithm

Combining the l-complete traffic models from Section 5.1 with the stopping criterion based on checking linear invariants from Section 5.2, we specialize Algorithm 1 into Algorithm 2 to generate a finite-state SACE simulation of the PETC traffic model $\mathcal{S}$, together with the computation of its SAIST $\underline{\mathrm{V}}(\mathcal{S})$. In the outer loop, the relation $\mathcal{R}_l$ and corresponding finite-state system $\mathcal{S}_l$ are built, followed by the computation of one of its SACs σ. Then, an inner loop looks for linear subspaces $\mathcal{A}$ of $\boldsymbol{M}_\sigma$ satisfying $\mathcal{A} \setminus \{\boldsymbol{0}\} \subseteq \mathcal{Q}_\sigma$ (Theorem 5); because $\boldsymbol{M}_\sigma$ is assumed to be mixed and of irrational rotations[7], it suffices to verify 1-dimensional subspaces for real eigenvectors and 2-dimensional subspaces for complex conjugate ones[8]; if one is found, the algorithm terminates. Otherwise, l is incremented and the main loop is repeated. Hereafter, we say that a linear invariant subspace of a mixed matrix is *basic* if it is the span of a real eigenvector or of a pair of complex conjugate eigenvectors.

[7] Any matrix is arbitrarily close to a mixed matrix of irrational rotations; numerically checking if it is otherwise is not robust. A more thorough discussion about this is available in Section 5.4.

[8] If a larger dimensional subspace $\mathcal{A}'$ is a subset of $\mathcal{Q}_\sigma$, any smaller dimensional subspace $\mathcal{A} \subset \mathcal{A}'$ will also be. Thus, there is no benefit in verifying subspaces that are combinations of smaller real linear subspaces.

Algorithm 2 PETC SAIST computation algorithm

Input: $\mathcal{Y}$ and $\boldsymbol{M}(k)$, $\mathcal{Q}_k$, $\forall k \in \mathcal{Y}$
Output: l, $\mathcal{S}_l$, σ, SAIST

$l \leftarrow 1$
while true **do**
  Build $\mathcal{R}_l$ and $\mathcal{S}_l$ ▷ (Definitions 7 and 8)
  SAIST $\leftarrow \underline{\mathrm{V}}(\mathcal{S}_l)$, $\sigma \leftarrow \mathrm{SAC}(\mathcal{S}_l)$ ▷ [14,21]
  for all $\mathcal{A} \in \mathrm{BILS}(\boldsymbol{M}_\sigma)$ **do** ▷ BILS = basic invariant linear subspaces
    if $\mathcal{A}$ satisfies Eq. (10) with $k_1, k_2, ..., k_m = \sigma$ **then**
      return
    end if
  end for
  $l \leftarrow l + 1$
end while

In order to state formal results about the correctness of Algorithm 2, we need to account for the conditions in Theorem 5.

Definition 11 (*Normalized Distance*)**.** The *normalized distance* between a point $\boldsymbol{x} \in \mathbb{R}^n$ and a set $\mathcal{A} \subseteq \mathbb{R}^n$, denoted by $d_{\mathrm{n}}(\boldsymbol{x}, \mathcal{A})$ is defined as $\inf_{\boldsymbol{l} \in \mathcal{A}} \left(1 - \frac{\boldsymbol{l}^{\mathrm{T}}\boldsymbol{x}}{|\boldsymbol{l}||\boldsymbol{x}|}\right)$. The normalized distance between two sets is $d_{\mathrm{n}}(\mathcal{A}, \mathcal{A}') := \inf_{\boldsymbol{l} \in \mathcal{A}} d_{\mathrm{n}}(\boldsymbol{l}, \mathcal{A}')$.

As the quantity $\frac{\boldsymbol{l}^{\mathrm{T}}\boldsymbol{x}}{|\boldsymbol{l}||\boldsymbol{x}|}$ is the cosine of the angle between the vectors $\boldsymbol{l}$ and $\boldsymbol{x}$, the normalized distance varies between 0 and 1, measuring how close $\boldsymbol{x}$ is, modulo magnitude, to the set $\mathcal{A}$. It is a more sensible choice of distance when dealing with homogeneous sets than the Euclidean distance, which would be zero as the origin is always in or arbitrarily close to such sets. This distance is needed for some technical results that come later, as well as for the following definition.

Definition 12 (*Regularity*)**.** A sequence of ISTs $\sigma := k_1 k_2 ... k_m$ is said to be *regular* if (i) $\boldsymbol{M}_\sigma$ is nonsingular, mixed, and of irrational rotations, and (ii) for every invariant linear subspace $\mathcal{A}$ of $\boldsymbol{M}_\sigma$, we have that $d_{\mathrm{n}}(\mathcal{A}, \partial\mathcal{Q}_\sigma) \geq \epsilon$ for some $\epsilon > 0$.

Regularity of a sequence σ prevents that one of the invariants of $\boldsymbol{M}_\sigma$ intersect $\partial\mathcal{Q}_\sigma$ (the case in the right of Fig. 4), requiring a minimal ϵ clearance to its boundary. The following result establishes conditions for the termination of Alg. 2; the proof is in Appendix A.

Theorem 6. *Suppose that $\mathcal{S}$ from Eq. (7) has an isolated smallest-in-average cycle σ that is regular. Then, Alg. 2 terminates with* SAIST $= \underline{\mathrm{V}}(\mathcal{S})$.

The conditions of Theorem 6 are the same behavioral conditions as in Theorem 4: the system must exhibit a minimizing periodic behavior, and competing infinite behaviors must be composed of subsequences that have average value strictly larger than the minimal value. Additionally, the smallest cycle must be regular, which is not a limiting assumption. Therefore, the algorithm may not terminate when, for example, a minimizing behavior is aperiodic. In this case, we may still expect increasingly better estimates of $\underline{\mathrm{V}}(\mathcal{S})$ with larger values of l.

5.4. Robustness and computability

Algorithm 2 relies on the matrices $\boldsymbol{M}(k)$ from Eq. (3), whose elements are typically transcendental. Therefore, one may wonder if the algorithm, or more generically a given l-complete SACE traffic model, is robust to small round-off errors when computing those matrices, as well as other small model mismatches. In this section, we are going to see that this is true given that some mild assumptions are satisfied. For this, we need proper definitions.

Definition 13 (*Perturbed PETC System*)**.** Given a PETC system (1)–(2) and its data $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{K}, \boldsymbol{Q}, \bar{k}$, the PETC system with data $\tilde{\boldsymbol{A}}, \tilde{\boldsymbol{B}}, \tilde{\boldsymbol{K}}, \tilde{\boldsymbol{Q}}, \bar{k}$ is called a δ-perturbation of the former if $\left|\boldsymbol{A} - \tilde{\boldsymbol{A}}\right| \leq \delta$, $\left|\boldsymbol{B}\boldsymbol{K} - \tilde{\boldsymbol{B}}\tilde{\boldsymbol{K}}\right| \leq \delta$, and $\left|\boldsymbol{Q} - \tilde{\boldsymbol{Q}}\right| \leq \delta$. Furthermore, the traffic model $\tilde{\mathcal{S}}$ cf. Eq. (7) of a δ-perturbation of system (1)–(2) is denoted a δ-perturbation of $\mathcal{S}$.

Remark 8. Considering Footnote 1, Definition 13 also encompasses variations in the actual checking period h.

Definition 14 (ϵ-*Inflation*)**.** The ϵ-*inflation* of a quadratic cone $\{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{x}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{x} \geq (>) 0\}$ is the set $\{\boldsymbol{x} \in \mathbb{R}^n \mid \boldsymbol{x}^{\mathrm{T}}(\boldsymbol{Q} + \epsilon\mathbf{I})\boldsymbol{x} \geq (>) 0\}$, for $\epsilon > 0$. An ϵ-inflation of the intersection of quadratic cones is defined as the intersection of the ϵ-inflations.

Let $\mathcal{P}_\delta(\mathcal{S})$ be the set of all δ-perturbations of $\mathcal{S}$. We have the following results.

Proposition 7. *Let $\mathcal{S}$, Eq. (7), be the traffic model of system (1)–(2). If $\mathcal{S}_l$ is an l-complete model thereof (Definition 8), then there exists $\delta > 0$ such that $\mathcal{S}_l$ is an l-complete model of every $\tilde{\mathcal{S}} \in \mathcal{P}_\delta(\mathcal{S})$ if there exists an $\epsilon > 0$ such that the following conditions hold:*

- *For every $\sigma \in \mathcal{B}^l(\mathcal{S})$, there exists $\boldsymbol{x} \in \mathcal{Q}_\sigma$ s.t. $d_{\mathrm{n}}(\boldsymbol{x}, \partial\mathcal{Q}_\sigma) > \epsilon$; and*
- *for every $\sigma \notin \mathcal{B}^l(\mathcal{S})$, every ϵ-inflation of $\mathcal{Q}_\sigma$ is empty.*

Proof. By Definition 8, $\mathcal{S}_l$ is an SICA of every $\tilde{\mathcal{S}} \in \mathcal{P}_\delta(\mathcal{S})$ if

1. $\sigma \in \mathcal{B}^l(\mathcal{S}) \implies \sigma \in \mathcal{B}^l(\tilde{\mathcal{S}}), \forall \tilde{\mathcal{S}} \in \mathcal{P}_\delta(\mathcal{S})$, and
2. $\sigma \notin \mathcal{B}^l(\mathcal{S}) \implies \sigma \notin \mathcal{B}^l(\tilde{\mathcal{S}}), \forall \tilde{\mathcal{S}} \in \mathcal{P}_\delta(\mathcal{S})$.

For item 1, we must have a non-zero vector $\boldsymbol{x} \in \tilde{\mathcal{Q}}_\sigma$, where $\tilde{\mathcal{Q}}_\sigma$ is the σ-cone of the δ-perturbation $\tilde{\mathcal{S}}$. Because $d_{\mathrm{n}}(\boldsymbol{x}, \partial\mathcal{Q}_\sigma) > \epsilon$, we have that the normalized distance to the complement of $\mathcal{Q}_\sigma$ satisfies $d_{\mathrm{n}}(\boldsymbol{x}, \bar{\mathcal{Q}}_\sigma) > \epsilon$. By continuity, this implies that $d_{\mathrm{n}}(\boldsymbol{x}, \bar{\tilde{\mathcal{Q}}}_\sigma) > 0$ for small enough δ, and hence $\boldsymbol{x} \in \tilde{\mathcal{Q}} \implies \sigma \in \mathcal{B}^l(\tilde{\mathcal{S}})$. Likewise, for item 2, we cannot have a vector $\boldsymbol{x} \in \tilde{\mathcal{Q}}_\sigma$; by continuity, for small enough δ, $\tilde{\mathcal{Q}}_\sigma$ is a subset of the ϵ-inflation of $\mathcal{Q}_\sigma$, which is empty, and therefore $\sigma \notin \mathcal{B}^l(\tilde{\mathcal{S}})$. □

The conditions in Proposition 7 rule out marginal cases of degeneracy, and are expected to hold in general for sufficiently small ϵ. I.e., if σ is a behavior of $\mathcal{S}$ and $\mathcal{Q}_\sigma$ has a non-empty interior (equivalent to $d_{\mathrm{n}}(\boldsymbol{x}, \partial\mathcal{Q}_\sigma) > \epsilon$ for some $\boldsymbol{x}$ and ϵ), then sufficiently small perturbations to the sets whose intersection gives $\mathcal{Q}_\sigma$ do not render it empty; symmetrically, if σ is *not* a behavior of $\mathcal{S}$, not only $\mathcal{Q}_\sigma$ must be empty, but small perturbations on the sets whose intersection composes $\mathcal{Q}_\sigma$ must retain its emptiness, thus not creating a new behavior.

Proposition 8. *Let σ^ω be a cyclic behavior of $\mathcal{S}$ from Eq. (7). Then, if σ is regular, there exists some $\delta > 0$ such that $\sigma^\omega \in \mathcal{B}^\omega(\tilde{\mathcal{S}})$, for all $\tilde{\mathcal{S}} \in \mathcal{P}_\delta(\mathcal{S})$.*

Proof. From Theorem 5, we have that $\sigma^\omega \in \mathcal{B}^\omega(\mathcal{S}) \implies \mathcal{A} \subseteq \mathrm{cl}(\mathcal{Q}_\sigma)$ for a basic linear invariant subspace $\mathcal{A}$ of $\boldsymbol{M}_\sigma$. From regularity of σ, $d_{\mathrm{n}}(\mathcal{A}, \partial\mathcal{Q}_\sigma) > \epsilon$. Together with $\mathcal{A} \subseteq \mathrm{cl}(\mathcal{Q}_\sigma)$, we have that $d_{\mathrm{n}}(\mathcal{A}, \bar{\mathcal{Q}}_\sigma) > \epsilon$. Since σ is regular, $\boldsymbol{M}_\sigma$ is mixed by definition. Then, by continuity of eigenvalues and eigenvectors, for small enough δ, the perturbed eigenvalues $\tilde{\lambda}_i$ are qualitatively unchanged: $\lambda_i \in \mathbb{R} \implies \tilde{\lambda}_i \in \mathbb{R}$, $\Im(\lambda_i) \neq 0 \implies \Im(\tilde{\lambda}_i) \neq 0$, and $|\lambda_i| > |\lambda_{i+1}| \implies |\tilde{\lambda}_i| > |\tilde{\lambda}_{i+1}|$. Thus, if $\mathcal{A}$ is a line associated to a real eigenvalue, so is the corresponding basic linear subspace $\tilde{\mathcal{A}}$ of $\tilde{\boldsymbol{M}}_\sigma$; and likewise if $\mathcal{A}$ is a plane corresponding to complex conjugate eigenvalues of irrational rotations: even if $\tilde{\boldsymbol{M}}_\sigma$ is not of irrational rotations, the plane $\tilde{\mathcal{A}}$ is one of its invariants. In addition, $d_{\mathrm{n}}(\mathcal{A}, \tilde{\mathcal{A}}) < d$, where d diminishes with δ. Hence, for small enough δ we have that $d_{\mathrm{n}}(\mathcal{A}, \bar{\mathcal{Q}}_\sigma) > \epsilon \implies d_{\mathrm{n}}(\tilde{\mathcal{A}}, \bar{\tilde{\mathcal{Q}}}_\sigma) > 0 \implies \tilde{\mathcal{A}} \setminus \{\boldsymbol{0}\} \subseteq \tilde{\mathcal{Q}}_\sigma$. Therefore, applying again Theorem 5, we conclude that $\sigma^\omega \in \mathcal{B}^\omega(\tilde{\mathcal{S}}), \forall \tilde{\mathcal{S}} \in \mathcal{P}_\epsilon(\mathcal{S})$. □

These two propositions combined give the following result:

Theorem 7. *Let $\mathcal{S}$, Eq. (7), be the traffic model of system (1)–(2), and let $\mathcal{S}_l$ be its SACE simulation. If its smallest-in-average cycle σ is regular and $\mathcal{S}_l$ satisfies the premises of Proposition 7, then there exists $\delta > 0$ such that $\mathcal{S}_l$ is SACE simulation of every $\tilde{\mathcal{S}} \in \mathcal{P}_\delta(\mathcal{S})$.*

Theorem 7 has two interesting implications. The first is that sufficiently small round-off errors on the matrices $\boldsymbol{M}(k)$ and $\boldsymbol{Q}$ of Eq. (3) do not affect the correct computation of $\underline{\mathrm{V}}(\mathcal{S})$; hence, $\underline{\mathrm{V}}(\mathcal{S})$ is computable for a class of linear systems, even though $\boldsymbol{M}(k)$ typically contains transcendental numbers. The second implication is that, informally, we can apply our method to nonlinear systems, as long as the closed loop ETC system is asymptotically stable and the involved functions are sufficiently smooth. Asymptotic stability implies that the state converges to a ball of any radius, no matter how small, in finite time; therefore, the sequence of sampling times up to this point do not affect the system's SAIST. Inside a sufficiently small ball, the nonlinear flow belongs to a convex combination of δ-perturbations of its linearization about the equilibrium. If the linearized system $\mathcal{S}$ satisfies the premises of Theorem 7, the SAIST of the nonlinear system is equal to $\underline{\mathrm{V}}(\mathcal{S})$.

5.5. An improved algorithm

Verifying the existence of each l-long behavior to obtain $\mathcal{R}_l$ and $\mathcal{S}_l$ (line 3 of Alg. 2) has exponential complexity on the number of variables [13,25]. To reduce the number of times these problems are solved, we propose a more efficient refinement approach than performing the full $(l+1)$-complete abstraction. At every iteration of Alg. 2, we only refine the states of the abstraction associated with the previous SAC. This procedure is explained in Algorithm 3, where $\mathcal{X}_\sigma$ is the set of states that compose the SAC.[9]

[9] In fact, the corrected Karp's algorithm in [21] returns sequence of states that generate the SAC, from which determining the SAC is trivial. Algorithm 3 needs the states, hence we use the function SAC* which returns both the states $\mathcal{X}_\sigma$ and the behavioral cycle σ itself.

Algorithm 3 Fast PETC SAIST computation algorithm

Input: $\mathcal{S}_1$ and $\boldsymbol{M}(k)$, $\mathcal{Q}_k$, $\forall k \in \mathcal{Y}$
Output: d, $\mathcal{S}_d$, σ, $\texttt{SAIST}$

$d \leftarrow 1$ ▷ d is the depth of the algorithm
while true **do**
  $\texttt{SAIST} \leftarrow \underline{\mathrm{V}}(\mathcal{S}_l)$, $(\mathcal{X}_\sigma, \sigma) \leftarrow \mathrm{SAC}^*(\mathcal{S}_l)$ ▷ [14,21]
  for all $\mathcal{A} \in \mathrm{BILS}(\boldsymbol{M}_\sigma)$ **do** ▷ BILS = basic invariant linear subspaces
    if $\mathcal{A}$ satisfies Eq. (10) with $k_1, k_2, \ldots, k_m =: \sigma$ **then**
      return
    end if
  end for
  $\mathcal{X}_{d+1} \leftarrow \mathcal{X}_d \setminus \mathcal{X}_\sigma$ ▷ Remove states to be refined
  for all $(\alpha, \beta) \in \mathcal{E}_d$ such that $\alpha \in \mathcal{X}_\sigma$ and $|\alpha| \leq |\beta|$ **do** ▷ If $|\alpha| > |\beta|$ no new candidate is generated via the domino rule
    $k \leftarrow \beta(|\alpha|)$ ▷ The $|\alpha|$-th element of β
    $\gamma \leftarrow \alpha k$ ▷ New candidate: sequence of length $|\alpha| + 1$ given the Domino rule
    if $\exists \boldsymbol{x} \in \mathbb{R}^{n_\mathrm{x}}$ such that $k_1 k_2 \ldots k_l =: \gamma$ satisfies Eqs. (8)–(9) **then** ▷ Nonlinear SMT
      $\mathcal{X}_{d+1} \leftarrow \mathcal{X}_{d+1} \cup \{\gamma\}$
    end if
  end for
  $\mathcal{E}_{d+1} \leftarrow \{(k\sigma, \sigma k') \mid k, k' \in \mathcal{Y}, k\sigma, \sigma k' \in \mathcal{X}_{d+1}\}$ ▷ Domino rule
  $H_{d+1}(k\sigma) \leftarrow k$ for all $k\sigma \in \mathcal{X}_{d+1}$
  $\mathcal{S}_{d+1} \leftarrow (\mathcal{X}_{d+1}, \mathcal{X}_{d+1}, \mathcal{E}_{d+1}, \mathcal{Y}, H_{d+1})$
  $d \leftarrow d + 1$
end while

Table 1
SAIST values for the example of Section 6.1.

a	0.1	0.2	0.3	0.4	0.5
l	50*	15	26	12	10
SAIST	1.572	2.74	3.42	5	6
CPU time [s]	327	41	147	29	45

* Algorithm interrupted before finding a verified cycle.

To illustrate this approach, see Fig. 2, where three steps of this refinement approach are executed in the example of Fig. 5: in depth 3, the SAC is already $(1, 1, 2)^\omega$, but it requires only 6 verifications: 1, 2, (1, 1), (1, 2), (1, 1, 1) (disproved) and (1, 1, 2); the 3-complete model would require up to $2+4+8 = 14$ verifications to obtain the same SAC. The disadvantage of this approach is that the obtained graph is more connected (as we have fewer states but more behaviors), and thus the computation of an upper bound using Proposition 4 often gives too distant values.

6. Numerical examples

In what follows we present three different numerical examples. They can be reproduced by using `ETCetera` [26] (https://gitlab.tudelft.nl/sync-lab/ETCetera), a tool to generate abstractions of ETC systems for scheduling, metric computation (this paper) and sampling strategy design. It contains an implementation of Algorithms 2 and 3, including an interface with Z3 to solve the nonlinear SMT problems associated with verifying a sequence σ (Remark 5). To reproduce the results of this paper, the following scripts within `ETCetera` can be used:

- `examples/nahs_example1_traj.py` for Fig. 6;
- `examples/nahs_example1_table.py` for Table 1 and other data presented in Section 6.1;
- `examples/nahs_example2_table.py` for Table 2 (beware, this takes hours to finish);
- `examples/nahs_example3.py` for the SAIST bounds in Section 6.3 and Fig. 7.

Table 2
SAIST values for the example of Section 6.2.

a	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
l	1	18*	14	8	6	7	6	5	9
SAIST	1	1.921	3	3	3	4	4	4	9.5
CPU time [s]	2	3056	1551	95	185	236	153	40	2955

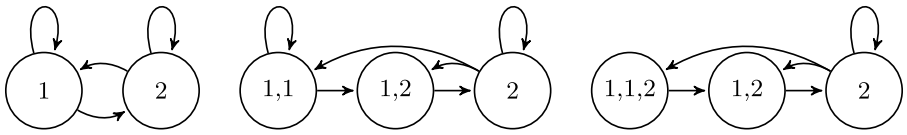

Fig. 5. Illustration of Alg. 3.

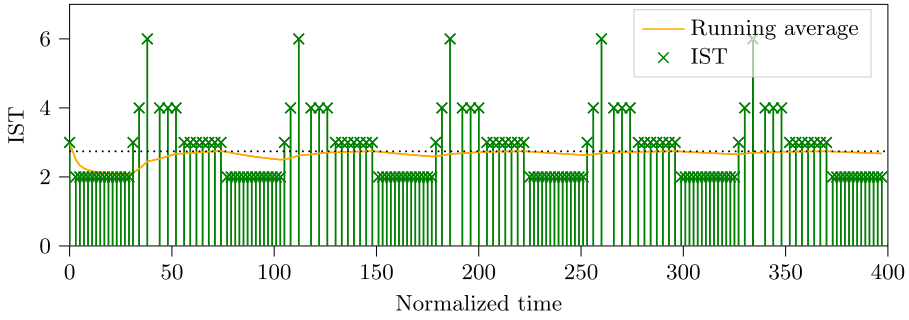

Fig. 6. ISTs and their running average for the example of Section 6.1 with $a = 0.2$ from a pseudo-randomly generated initial state. The dashed black line represents the computed SAIST.

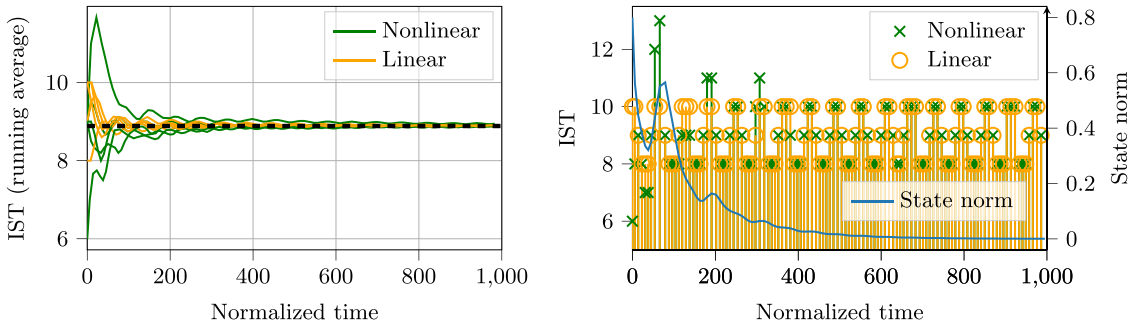

Fig. 7. Left: running average of ISTs of five nonlinear PETC simulations and of five corresponding linear PETC simulations, with the dashed black line representing the estimated SAIST. Right: ISTs for one nonlinear PETC simulation and the corresponding ISTs predicted by the linear PETC model, with the state norm overlaid on a secondary axis.

6.1. A two-dimensional linear system

We start by considering the example from [7]: the system (1) with

$$\boldsymbol{A} = \begin{bmatrix} 0 & 1 \\ -2 & 3 \end{bmatrix}, \quad \boldsymbol{B} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad \boldsymbol{K} = \begin{bmatrix} 0 & -5 \end{bmatrix},$$

and the triggering condition of [2], $|\boldsymbol{\xi}(t) - \hat{\boldsymbol{\xi}}(t)| > a|\boldsymbol{\xi}(t)|$ for some $0 < a < 1$, which can be put in the form Eq. (2). Checking time was set to $h = 0.05$, and maximum inter-sample time to $\bar{k} = 20$. Using a Python implementation of Algorithm 2 with Z3 [24] to solve Eq. (8), we attempted to compute its SAIST through a SACE simulation for $a \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$. Table 1 presents the SAIST for each a, as well as the l value (Definition 8) where it was obtained. Only for $a = 0.1$ the algorithm did not terminate before $l = 50$: for this case, the actual $\bar{k}$ of the system was 3, and all $\boldsymbol{M}(k)$, $k \leq 3$, have complex eigenvalues. Thus, it is possible that it does not have periodic behaviors, similarly to the irrational rotation of Example 2. Nonetheless, applying Proposition 4 gives an upper bound for $\underline{V}(\mathcal{S})$ of 1.596; hence, we know that $1.572 \leq \underline{V}(\mathcal{S}) \leq 1.596$, giving an uncertainty of only 0.024. For the other cases, trivial cycles were found for

$a = 0.4$ (5^ω) and $a = 0.5$ (6^ω), but it took a few iterations to break, e.g., the 2^ω loop. Interestingly, the simplest cycles for $a = 0.2$ and $a = 0.3$ had length, respectively, 27 and 28, showing that PETC can often lead to very complex recurring patterns (see an example in Fig. 6). In addition, the case of $a = 0.4$ has two verified cyclic behaviors, 5^ω and 6^ω, while with $a = 0.5$ three cycles are obtained: 6^ω, 7^ω and 8^ω: this confirms that a single PETC system can exhibit multiple different periodic behaviors.

The results were generated on a MacBook Pro 2017 using a single processor. As Table 1 shows, even for $l = 50$ the CPU time was kept under 10 min.

6.2. A three-dimensional linear system

With $n_x = 3$, the computational time involved in solving the existence problem of Eq. (8) increased significantly. Therefore we applied Alg. 3 to system (1)–(2) with

$$\boldsymbol{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & -1 & -1 \end{bmatrix}, \quad \boldsymbol{B} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \boldsymbol{K} = \begin{bmatrix} -2 & -1 & -1 \end{bmatrix},$$

with $h = 0.1, \bar{k} = 20$ and the triggering condition $|\boldsymbol{\xi}(t) - \hat{\boldsymbol{\xi}}(t)| > a|\boldsymbol{\xi}(t)|$. This time, some parallelization was also applied: at most 10 threads of an Intel® Xeon® W-2145 CPU were used, solving multiple instances of Eq. (8) in parallel whenever possible. Table 2 shows the results for multiple choices of a, where l now is the largest length of any state in the abstraction. The algorithm was set to a maximum depth of 200, which was only reached for $a = 0.2$. The CPU times varied dramatically, in some cases taking less than a minute, whilst in others reaching an hour. The most interesting thing we observe is that, even though the SAIST never decreases with a as expected, there is not a consistent increase on its values after $a = 0.3$. This is reasonable considering the results of Section 5.4: for small enough perturbations of the ETC system's parameters, the same cycle may still be present (Proposition 8). Interestingly, for $a = 0.9$ there is a substantial jump in the SAIST value.

6.3. A nonlinear system

Consider now the PETC triggering rule $|\boldsymbol{\xi}(t) - \hat{\boldsymbol{\xi}}(t)| > a|\boldsymbol{\xi}(t)|$ with $h = 0.05, a = 0.452$ applied to the following nonlinear jet engine system [27]:

$$\begin{aligned} \dot{\xi}_1(t) &= -\xi_2(t) - 1.5\xi_1(t)^2 - 0.5\xi_1(t)^3 \\ \dot{\xi}_2(t) &= \upsilon(t), \\ \upsilon(t) &= \hat{\xi}_1(t) - 0.5(\hat{\xi}_1(t)^2 + 1)(y(t) + \hat{\xi}_1(t)^2 y(t) + \hat{\xi}_1(t) y(t)^2), \end{aligned}$$

where $y(t) = (\hat{\xi}_1(t)^2 + \hat{\xi}_2(t))/(\hat{\xi}_1(t)^2 + 1)$. The origin of the closed-loop system is asymptotically stable,[10] therefore we can obtain its SAIST through its linearized model around the origin, which is of the form (1) with

$$\boldsymbol{A} = \begin{bmatrix} 0 & -1 \\ 0 & 0 \end{bmatrix}, \quad \boldsymbol{B} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad \boldsymbol{K} = \begin{bmatrix} 1 & -0.5 \end{bmatrix}.$$

We ran Alg. 2 and stopped it with $l = 100$, obtaining an approximate value of $\underline{\mathrm{V}}(\mathcal{S}) = 8.882$. Using Proposition 4, an upper bound of 8.892 was obtained, thus giving an error of 0.01. Fig. 7 shows ISTs and their running averages for five PETC simulations starting each from a different pseudo-randomly generated initial state, for both the nonlinear model and the linearized model. It can be seen that the running averages in both cases converge to the predicted SAIST value, even though the averages are significantly different in the beginning of the simulation. The right plot shows how the difference between ISTs based on the nonlinear model and the linear model diminish as the state norm approaches zero: in the plotted simulation there is no error after the state norm is below 0.03 (around time instant 400).

7. Conclusions

We have presented a method to compute the sampling performance of PETC, namely its minimum average inter-sample time, by means of an abstraction called SACE simulation. For this we rely on methods of abstracting and refining to obtain tighter simulations, and getting their smallest-in-average cycle through Karp's algorithm. A SACE simulation requires that this cycle, repeated *ad infinitum*, is a behavior of the concrete system; for this, we need to find an invariant of the system, which is possible for PETC of linear systems through the inspection of linear invariants of an associated discrete-time linear system. In the generic case – quotient sets with non-empty interior and linear invariants not touching the boundary of the cones they belong to – a SACE simulation is proven to be robust to small model uncertainties, which allows us to use the presented method to a large class of nonlinear systems. Even if an exact SACE simulation is not obtained, every

[10] For stability analysis of PETC of nonlinear systems, see, e.g., [28].

simulation provides a lower bound to the SAIST, and upper bounds can also be computed from the abstractions. Our numerical results indicate that these bounds can be very close after sufficient refinements.

As with most applications of finite-state abstractions, our approach suffers from the "curse of dimensionality": with a three-dimensional system the computation can reach nearly an hour to complete. In fact, it can be argued that this curse is more severe in our case than in most control and verification applications, since we rely on *strongest* l-complete abstractions, which require no spurious behavior fragments of length up to l. This may prevent the usage of most reachability tools to this end, as over- or under-approximations can create such spurious behaviors or remove potentially important ones. This is one of the reasons why we have used Z3 for our implementation, as it is one of the few exact nonlinear SAT solvers available. Nevertheless, the robustness results we have presented indicate that exactness may not be necessary in most cases. With this in mind, we plan to use approximate nonlinear SMT solvers such as dReal [29] to start addressing the issue of dimensionality. An additional approach to tackle the computational complexity is employing (massive) parallelization: in fact, all candidate l-long behaviors (line 3 of Alg. 2) can be checked in parallel provided sufficient resources are available.

It is interesting to observe that the problem of computing the (smallest) limit average metric of an infinite system is highly dependent on its infinite behavior properties: systems with aperiodic behavior can make it impossible to obtain a SACE simulation, but other pathological behaviors can be even worse, such as the infamous $(1^n2^n)^\omega$, where not even a good approximation can be achieved. Better behavioral understanding of systems is crucial for the further development of quantitative verification methods. Part of this behavioral understanding of ETC sampling is currently the subject of our investigation.

Finally, natural extensions of this line of work are ongoing, such as extending it to systems with disturbances, in particular stochastic noise [30], as well as the usage of abstractions for *synthesis* of sampling strategies that maximize the closed-loop SAIST [22].

CRediT authorship contribution statement

Gabriel de Albuquerque Gleizer: Conceptualization, Methodology, Software, Writing – original draft, Visualization, Investigation, Formal analysis. **Manuel Mazo Jr:** Conceptualization, Supervision, Writing – review & editing, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Proof of Theorem 4

The proof relies on the notion of *cyclic permutations*. A word σ' is called a cyclic permutation of $\sigma := a_0a_1...a_n$ if $\sigma' = a_ia_{i+1}...a_na_0a_1...a_{i-1}$ for some $i \leq n$. For example, the cyclic permutations of 1234 are 1234, 2341, 3412, and 4123. Clearly, all n-long subsequences of σ^ω are precisely the cyclic permutations of σ. Now we introduce the following Lemmas:

Lemma 1. *Let $\sigma \in \mathcal{Y}^n$ and $\sigma' \in \mathcal{Y}^n$ be cyclic permutations of each other. If $\sigma = \alpha a$ and $\sigma' = \alpha b$, where $\alpha \in \mathcal{Y}^{n-1}$ and $a, b \in \mathcal{Y}$, then $a = b$ and thus $\sigma = \sigma'$.*

Proof. Let $\sigma = a_0a_1...a_{n-1}$. Then $\sigma' = a_ia_{i+1}...a_{n-1}a_0...a_{i-1}$ for some $i > 0$ (if $i = 0$ the result is trivial). If their $(n-1)$-long prefixes are equal, then $a_j = a_{j+i \bmod n}$ for all $j < n-1$. In particular, take $j = i-1$; then $a_{i-1} = a_{2i-1 \bmod n} = a_{3i-1 \bmod n} = \cdots = a_{ki-1 \bmod n}$, where k is the smallest number such that $ki - 1 \bmod n = n-1$ (in the worst case, $k = n$, for i and n coprime). Thus, $a_{i-1} = a_{ki-1 \bmod n} = a_{n-1}$, concluding the proof. □

Lemma 2. *Let $\sigma \in \mathcal{Y}^n$ and $\sigma' \in \mathcal{Y}^n$ be cyclic permutations of each other. If $\sigma \neq \sigma'$, then there is a subsequence α of length n of $\sigma\sigma'$ that is not a cyclic permutation of σ.*

Proof. Let $\sigma = a_0a_1...a_{n-1}$. Then $\sigma' = a_ia_{i+1}...a_{n-1}a_0...a_{i-1}$ for some $i > 0$. We have $\sigma\sigma' = a_0a_1...a_{n-1}a_ia_{i+1}...a_{n-1}a_0...a_{i-1}$.

Suppose, for contradiction, that every n-long subsequence of $\sigma\sigma'$ is a cyclic permutation of σ. Let us look at the first nontrivial subsequence, $\sigma_1 := a_1...a_{n-1}a_i$. Because $a_1...a_{n-1}a_0$ is a cyclic permutation of σ, from Lemma 1 we get that $a_0 = a_i$. Now let us apply induction: suppose that for some $J < n$, $a_j = a_{i+j \bmod n}$ for all $j < J$; we are going to show that this also holds for $j = J$. First, suppose that $J < n-i$; then $\sigma_J = a_Ja_{J+1}...a_{n-1}a_ia_{i+1}...a_{i+J-2}a_{i+J-1} = a_Ja_{J+1}...a_{n-1}a_0a_1...a_{J-2}a_{i+J-1}$. Again, because $a_Ja_{J+1}...a_{n-1}a_0a_1...a_{J-1}$ is a cyclic permutation of σ, apply Lemma 1 to obtain $a_{i+J-1} = a_{J-1}$. Second, suppose that $J \geq n-i$. Then, $\sigma_J = a_J...a_{n-1}a_i..a_{n-1}a_0a_1...a_{i+J-n-1} = a_J...a_{n-1}a_0..a_{n-i-1}a_0a_1...a_{i+J-n-1}$. Note that $a_k = a_{k+n \bmod n} = a_{k+n-i}$ as long as $k + n - i < J$, i.e., $k < i + J - n$. Thus, $\sigma_J = a_J...a_{n-1}a_0..a_{n-i-1}a_{n-i}...a_{J-2}a_{i+J-n-1}$. Again, apply Lemma 1 to get that $a_{i+J-n-1} = a_{J-1}$. We have that $J - 1 + i \bmod n = i + J - n - 1$, since $n > J \geq n - i$; our hypothesis is thus confirmed. The fact that $a_j = a_{i+j \bmod n}$ for all $j < n$ implies that $\sigma' = a_ia_{i+1}...a_{n-1}a_0...a_{i-1} = a_0a_1...a_{n-1-i}a_{n-i}...a_{n-1} = \sigma$, which contradicts the fact that $\sigma \neq \sigma'$. □

Proof of Theorem 4. From Theorem 3, there is an l large enough such that $\underline{\mathrm{V}}(\mathcal{S}_l) = \underline{\mathrm{V}}(\mathcal{S})$. It is easy to see that taking $l \geq m$ ensures that σ is a cycle of the graph associated to $\mathcal{S}_l$.

We prove that, because now $\mathrm{LimAvg}(\beta^\omega) > \underline{\mathrm{V}}(\mathcal{S})$ for every β that is not a subsequence of σ^ω (thus not a cyclic permutation of σ), the SAC of $\mathcal{S}_l$ is unique up to cyclic permutations. Suppose, for contradiction, that another cycle σ' is a SAC of $\mathcal{S}_l$, with $|\sigma'| = p$. As in the proof of Theorem 3, we divide $(\sigma')^m$ into p subsequences of length m, obtaining

$$\underline{\mathrm{V}}(\mathcal{S}_l) = \mathrm{LimAvg}((\sigma')^\omega) = \mathrm{LimAvg}(((\sigma')^m)^\omega) = \mathrm{Avg}((\sigma')^m) = \frac{1}{p}\sum_{i=1}^{p}\mathrm{Avg}(\beta_i).$$

If (i) some β_i is not a cyclic permutation of σ, $\frac{1}{p}\sum_{i=1}^{p}\mathrm{Avg}(\beta_i) > \underline{\mathrm{V}}(\mathcal{S}_l)$, which yields the contradiction. Now, suppose (ii) that every β_i is a cyclic permutation of σ; since σ' is not the same cycle as σ, it cannot be that $\beta_i = \beta_j$ for all $i, j \leq p$. If $\beta_i \neq \beta_j$ for some i, j, suppose without loss of generality that they are adjacent in $(\sigma')^\omega$, i.e., either $j = i+1$ or $i = p$ and $j = 1$. Then we have from Lemma 2 that there exists an m-long subsequence of $\beta_i\beta_j$ that is not a cyclic permutation of σ. Thus, σ' has at least one subsequence β' with average larger than $\underline{\mathrm{V}}(\mathcal{S})$, which brings us back to case (i). The contradiction is thus achieved in all cases.

Concluding, $\mathcal{S}_l$ has only one cycle σ (modulo cyclic permutations) that attains its minimum value. Hence, running Karp's algorithm (Theorem 2) retrieves it; by assumption, $\sigma^\omega \in \mathcal{B}^\omega(\mathcal{S})$, thus the algorithm terminates at line 6. □

Appendix B. Proof of Theorem 5

Before the main proof, we need some definitions. Given a map $f : X \to X$ and the discrete-time autonomous system defined by $x_{i+1} = f(x_i)$, we call the *forward orbit* of x the set $\mathcal{O}(x) \coloneqq \{f^n(x) \mid n \in \mathbb{N}\}$. The ω-limit set of x, denoted by $\omega(x)$ is the set of cluster points of $\mathcal{O}(x)$, or alternatively,

$$\omega(x) = \bigcap_{n\in\mathbb{N}} \mathrm{cl}(\{f^k(x) \mid k > n\}).$$

By definition of closure, if $\mathcal{O}(x) \subset \mathcal{A} \subset X$, then $\omega(x) \subset \mathrm{cl}(\mathcal{A})$.

We introduce the following Lemma.

Lemma 3. *Let $\boldsymbol{M} \in \mathbb{R}^{n\times n}$ be a nonsingular mixed matrix and $\mathcal{Q} \subseteq \mathbb{R}^n$ be a homogeneous set, i.e., it satisfies $\boldsymbol{x} \in \mathcal{Q} \implies \lambda\boldsymbol{x} \in \mathcal{Q}, \forall\lambda \in \mathbb{R}\setminus\{0\}$. If there exists a trajectory $\boldsymbol{\xi} : \mathbb{N} \to \mathbb{R}^{n_\mathrm{x}}$ satisfying $\boldsymbol{\xi}(k+1) = \boldsymbol{M}\boldsymbol{\xi}(k)$ and $\boldsymbol{\xi}(k) \in \mathcal{Q}\ \forall k \in \mathbb{N}$, then there exists a linear subspace $\mathcal{A}$ that is an invariant of $\boldsymbol{M}^q$ and satisfies $\mathcal{A} \subseteq \mathrm{cl}(\mathcal{Q})$, where $q \in \mathbb{N}$. Furthermore, $q = 1$ if $\boldsymbol{M}$ is of irrational rotations.*

Proof. Because $\mathcal{Q}$ is homogeneous, $\boldsymbol{\xi}(k) \in \mathcal{Q}$ for all k implies that the normalized trajectory $\boldsymbol{\xi}(k)/|\boldsymbol{\xi}(k)| \in \mathcal{Q}$ for all k; likewise, for any constant $c \neq 0$, we have that $c\boldsymbol{\xi}(k)/|\boldsymbol{\xi}(k)| \in \mathcal{Q}$. Therefore, let us investigate the "normalized" version of the iteration $\boldsymbol{x}_{i+1} = \boldsymbol{M}\boldsymbol{x}_i$: this is defined by the map $f : B^n \to B^n$, where B^n is the unit ball in $\mathbb{R}^n$ and $f(\boldsymbol{x}) = \boldsymbol{M}\boldsymbol{x}/|\boldsymbol{M}\boldsymbol{x}|$. Our strategy is to first determine what is $\omega(\boldsymbol{x})$; then, we will prove that the set $\{c\omega(\boldsymbol{x}) \mid c \in \mathbb{R}\setminus\{0\}\}$, a radial expansion of $\omega(\boldsymbol{x})$, is a linear subspace of $\boldsymbol{M}$. Because $\omega(\boldsymbol{x}) \subseteq \mathrm{cl}(\mathcal{Q})$ and $\boldsymbol{x} \in \mathcal{Q} \implies c\boldsymbol{x} \in \mathcal{Q}$, we conclude that $\{c\omega(\boldsymbol{x}) \mid c \in \mathbb{R}\setminus\{0\}\} \subseteq \mathrm{cl}(\mathcal{Q})$.

Now we investigate case by case depending on the eigenvalues λ_i of $\boldsymbol{M}$. Since $\boldsymbol{M}$ is mixed, it is diagonalizable, and hence the trajectory $\boldsymbol{\xi}(k)$ can be decomposed as $\sum_{i=1}^{n} a_i\boldsymbol{v}_i\lambda_i^k$, where $\boldsymbol{v}_i$ are the eigenvectors of $\boldsymbol{M}$ satisfying $|\boldsymbol{v}_i| = 1$, and the coefficients a_i are chosen such that $\boldsymbol{\xi}(0) = \sum_{i=1}^{n} a_i\boldsymbol{v}_i$. Let $m \leq n$ such that $a_i = 0$ for $i < m$, hence λ_m is the dominant eigenvalue for this initial condition. Throughout, let $\boldsymbol{x} \coloneqq \boldsymbol{\xi}(0)/|\boldsymbol{\xi}(0)|$.

Case 1: λ_m is real. Then

$$\begin{aligned}\lim_{k\to\infty}\frac{\boldsymbol{\xi}(k)}{|\boldsymbol{\xi}(k)|} &= \lim_{k\to\infty}\frac{a_m\boldsymbol{v}_m\lambda_m^k + \cdots + a_n\boldsymbol{v}_n\lambda_n^k}{|a_m\boldsymbol{v}_m\lambda_m^k + \cdots + a_n\boldsymbol{v}_n\lambda_n^k|}\\ &= \lim_{k\to\infty}\frac{a_m\boldsymbol{v}_m + \cdots + a_n\boldsymbol{v}_n\left(\frac{\lambda_n}{\lambda_m}\right)^k}{\left|a_m\boldsymbol{v}_m + \cdots + a_n\boldsymbol{v}_n\left(\frac{\lambda_n}{\lambda_m}\right)^k\right|} = \lim_{k\to\infty}\frac{a_m\boldsymbol{v}_m}{|a_m\boldsymbol{v}_m|} = \pm a_m\boldsymbol{v}_m.\end{aligned}$$

Hence, the set $\{c\omega(\boldsymbol{x}) \mid c \in \mathbb{R}\setminus\{0\}\}$ is the line $\{\pm c\boldsymbol{v}_m \mid c \in \mathbb{R}\setminus\{0\}\} = \{c\boldsymbol{v}_m \mid c \in \mathbb{R}\setminus\{0\}\}$, which is an invariant of $\boldsymbol{M}$.

For the next cases, λ_m and λ_{m+1} form a complex conjugate pair, thus $\boldsymbol{v}_{i+1} = \boldsymbol{v}_i^*$. Denote by $\theta \coloneqq \arg\lambda_m$.

Case 2: $\theta/\pi \notin \mathbb{Q}$. Using a similar approach as Case 1, we get $\lim_{k\to\infty}\boldsymbol{\xi}(k)/|\boldsymbol{\xi}(k)| = \pm(\boldsymbol{v}_m\mathrm{e}^{\mathrm{i}\theta k} + \boldsymbol{v}_{m+1}\mathrm{e}^{-\mathrm{i}\theta k})$. Because θ is not a rational multiple of π, $\{k\theta \mid k \in \mathbb{N}\}$ is a dense subset of $[0, 2\pi]$ and, therefore, $\omega(\boldsymbol{x}) = \mathrm{cl}\{\pm(\boldsymbol{v}_m\mathrm{e}^{\mathrm{i}\theta k}) + \boldsymbol{v}_{m+1}\mathrm{e}^{-\mathrm{i}\theta k} \mid \theta \in k \in \mathbb{N}\}$ which is equal to the ellipse $\mathcal{B} \coloneqq \{\boldsymbol{v}_m\mathrm{e}^{\mathrm{i}\alpha} + \boldsymbol{v}_{m+1}\mathrm{e}^{-\mathrm{i}\alpha} \mid \alpha \in [0, 2\pi)\}$. The set $\{c\boldsymbol{x} \mid \boldsymbol{x} \in \mathcal{B}, c \in \mathbb{R}\setminus\{0\}\}$ is the unique plane supported by $\boldsymbol{v}_m$ and $\boldsymbol{v}_{m+1}$, and as such is an invariant of $\boldsymbol{M}$.

Case 3: $\theta/\pi = p/q$, where $p, q \in \mathbb{N}$ are co-prime. The m-th and $(m+1)$-th eigenvalues of $\boldsymbol{M}$ have the form $r\mathrm{e}^{\pm\mathrm{i}p\pi/q}$, and as a consequence the corresponding eigenvalues of $\boldsymbol{M}^q$ are $\lambda_m^q = \lambda_{m+1}^q = r^q \in \mathbb{R}$. The geometric multiplicity of λ_m^q is 2, since $\boldsymbol{M}^q$ is also diagonalizable. Thus, we have that $\lim_{k\to\infty}\boldsymbol{\xi}(qk)/|\boldsymbol{\xi}(qk)| = a_m\boldsymbol{v}_i + a_{m+1}\boldsymbol{v}_{i+1} =: \boldsymbol{z}$. Hence, we have $\omega(\boldsymbol{x}) \supseteq \{c\boldsymbol{z} \mid c \in \mathbb{R}\setminus\{0\}\}$, a line that is an invariant of $\boldsymbol{M}^q$. Finally, this line is a subset of $\mathrm{cl}(\mathcal{Q})$, since $\{c\boldsymbol{z} \mid c \in \mathbb{R}\setminus\{0\}\} \subseteq \omega(\boldsymbol{x}) \subseteq \mathrm{cl}(\mathcal{Q})$. □

Proof of Theorem 5. Statement (i), $\mathcal{A} \setminus \{\mathbf{0}\} \subseteq \mathcal{Q}_\sigma$ implies $\sigma^\omega \in \mathcal{B}(\mathcal{S})$, is straightforward. Take any point $\boldsymbol{x} \in \mathcal{A} \subseteq \mathcal{Q}_\sigma$. By definition of $\mathcal{Q}_\sigma$, we have that $\boldsymbol{x} \in \mathcal{Q}_{k_1}$, $\boldsymbol{M}(k_1)\boldsymbol{x} \in \mathcal{Q}_{k_2}$, ..., and $\boldsymbol{M}(k_{m-1})\cdots\boldsymbol{M}(k_1)\boldsymbol{x} \in \mathcal{Q}_{k_m}$. The $(m+1)$-th element of the run starting from initial state $\boldsymbol{x}$ is $\boldsymbol{x}' = \boldsymbol{M}(k_m)\boldsymbol{M}(k_{m-1})\cdots\boldsymbol{M}(k_1)\boldsymbol{x} = \boldsymbol{M}_\sigma\boldsymbol{x}$. Since $\mathcal{A}$ is an invariant of $\boldsymbol{M}_\sigma$ and this matrix is nonsingular, $\boldsymbol{x}' \in \mathcal{A} \setminus \{\mathbf{0}\}$. Thus, the behavior from $\boldsymbol{x}$ is $\sigma\mathcal{B}_{\boldsymbol{x}'}(\mathcal{S})$. Applying the same reasoning recursively with $\boldsymbol{x}'$ in the place of $\boldsymbol{x}$, we conclude that $\mathcal{B}_{\boldsymbol{x}}(\mathcal{S}) = \sigma^\omega$.

Statement (ii) follows from Lemma 3, by applying it with $\mathcal{Q} = \mathcal{Q}_\sigma$ and $\boldsymbol{M} = \boldsymbol{M}_\sigma$, and using the fact that $\mathcal{Q}_\sigma$ is an homogeneous set. □

References

[1] K.J. Åström, B. Bernhardsson, Comparison of Riemann and lebesgue sampling for first order stochastic systems, in: Proceedings of the 41st IEEE Conference on Decision and Control, 2002, Vol. 2, IEEE, 2002, pp. 2011–2016.
[2] P. Tabuada, Event-triggered real-time scheduling of stabilizing control tasks, IEEE Trans. Automat. Control 52 (9) (2007) 1680–1685.
[3] X. Wang, M.D. Lemmon, Event design in event-triggered feedback control systems, in: Decision and Control, 2008. CDC 2008. 47th IEEE Conference on, IEEE, 2008, pp. 2105–2110.
[4] A. Girard, Dynamic triggering mechanisms for event-triggered control, IEEE Trans. Automat. Control 60 (7) (2015) 1992–1997.
[5] W. Heemels, K.H. Johansson, P. Tabuada, An introduction to event-triggered and self-triggered control, in: Decision and Control (CDC), 2012 IEEE 51st Annual Conference on, IEEE, 2012, pp. 3270–3285.
[6] W.P.M.H. Heemels, M.C.F. Donkers, A.R. Teel, Periodic event-triggered control for linear systems, IEEE Trans. Automat. Control 58 (4) (2013) 847–861.
[7] G. de A. Gleizer, M. Mazo Jr., Computing the sampling performance of event-triggered control, in: Proc. of the 24th Int'L Conf. on Hybrid Systems: Computation and Control, HSCC '21, ACM, 2021.
[8] R. Postoyan, R.G. Sanfelice, W.P.M.H. Heemels, Inter-event times analysis for planar linear event-triggered controlled systems, in: Decision and Control, 2019. CDC 2019. 58th IEEE Conference on, IEEE, 2019, pp. 3601–3606.
[9] A. Rajan, P. Tallapragada, Analysis of inter-event times for planar linear systems under a general class of event triggering rules, in: 2020 59th IEEE Conference on Decision and Control, CDC, IEEE, 2020, pp. 5206–5211, http://dx.doi.org/10.1109/CDC42340.2020.9304406.
[10] A.S. Kolarijani, M. Mazo Jr., A formal traffic characterization of LTI event-triggered control systems, IEEE Trans. Control Netw. Syst. (2016).
[11] G. de A. Gleizer, M. Mazo Jr., Scalable traffic models for scheduling of linear periodic event-triggered controllers, IFAC-PapersOnLine 53 (2) (2020) 2726–2732.
[12] P. Tabuada, Verification and Control of Hybrid Systems: A Symbolic Approach, Springer Science & Business Media, 2009.
[13] G. de A. Gleizer, M. Mazo Jr., Towards traffic bisimulation of linear periodic event-triggered controllers, IEEE Control Syst. Lett. 5 (1) (2021) 25–30.
[14] R.M. Karp, A characterization of the minimum cycle mean in a digraph, Discrete Math. 23 (3) (1978) 309–311.
[15] J.C. Willems, Paradigms and puzzles in the theory of dynamical systems, IEEE Trans. Automat. Control 36 (3) (1991) 259–294.
[16] T. Moor, J. Raisch, Supervisory control of hybrid systems within a behavioural framework, Systems Control Lett. 38 (3) (1999) 157–166.
[17] A.-K. Schmuck, P. Tabuada, J. Raisch, Comparing asynchronous l-complete approximations and quotient based abstractions, in: 2015 54th IEEE Conference on Decision and Control, CDC, IEEE, 2015, pp. 6823–6829.
[18] K.J. Åström, B. Wittenmark, Computer-Controlled Systems: Theory and Design, Courier Corporation, 2013.
[19] G. de A. Gleizer, M. Mazo Jr., Self-triggered output feedback control for perturbed linear systems, IFAC-PapersOnLine 51 (23) (2018) 248–253.
[20] K. Chatterjee, L. Doyen, T.A. Henzinger, Quantitative languages, ACM Trans. Comput. Log. (TOCL) 11 (4) (2010) 1–38.
[21] M. Chaturvedi, R.M. McConnell, A note on finding minimum mean cycle, Inform. Process. Lett. 127 (2017) 21–22.
[22] G. de A. Gleizer, K. Madnani, M. Mazo Jr., Self-triggered control for near-maximal average inter-sample time, in: 2021 60th IEEE Conference on Decision and Control (CDC), 2021, pp. 1308–1313, http://dx.doi.org/10.1109/CDC45484.2021.9682986.
[23] W. de Melo, S. van Strien, One-Dimensional Dynamics, in: Ergebnisse der Mathematik und ihrer Grenzgebiete. 3. Folge / A Series of Modern Surveys in Mathematics, Springer, Berlin Heidelberg, 2012.
[24] L. de Moura, N. Bjørner, Z3: An efficient SMT solver, in: International Conference on Tools and Algorithms for the Construction and Analysis of Systems, Springer, 2008, pp. 337–340.
[25] S. Basu, R. Pollack, M.-F. Roy, On the combinatorial and algebraic complexity of quantifier elimination, J. ACM 43 (6) (1996) 1002–1045.
[26] G. Delimpaltadakis, G. de A. Gleizer, I. van Straalen, M. Mazo Jr., ETCetera: beyond event-triggered control, in: 25th ACM International Conference on Hybrid Systems: Computation and Control, in: HSCC '22, Association for Computing Machinery, New York, NY, USA, 2022, http://dx.doi.org/10.1145/3501710.3519523.
[27] G. Delimpaltadakis, M. Mazo Jr., Isochronous partitions for region-based self-triggered control, IEEE Trans. Automat. Control 66 (3) (2020) 1160–1173.
[28] R. Postoyan, A. Anta, W.P.M.H. Heemels, P. Tabuada, D. Nešić, Periodic event-triggered control for nonlinear systems, in: 52nd IEEE Conference on Decision and Control, IEEE, 2013, pp. 7397–7402.
[29] S. Gao, S. Kong, E.M. Clarke, dReal: An SMT solver for nonlinear theories over the reals, in: International Conference on Automated Deduction, Springer, 2013, pp. 208–214.
[30] G. Delimpaltadakis, L. Laurenti, M. Mazo Jr., Abstracting the sampling behaviour of stochastic linear periodic event-triggered control systems, in: 2021 60th IEEE Conference on Decision and Control (CDC), 2021, pp. 1287–1294, http://dx.doi.org/10.1109/CDC45484.2021.9683751.