# Abstracted models for scheduling of event-triggered control data traffic

Mazo, M.; Sharifi-Kolarijani, A.; Adzkiya, D.; Hop, C.

**DOI**
10.1007/978-3-319-78449-6_10

**Publication date**
2018

**Document Version**
Final published version

**Published in**
Control Subject to Computational and Communication Constraints

# Chapter 10
# Abstracted Models for Scheduling of Event-Triggered Control Data Traffic


**M. Mazo Jr., A. Sharifi-Kolarijani, D. Adzkiya and C. Hop**

**Abstract** Event-Triggered control (ETC) implementations have been proposed to overcome the inefficiencies of periodic (time-triggered) controller designs, namely the over-exploitation of the computing and communication infrastructure. However, the potential of aperiodic Event-Triggered techniques to reuse the freed bandwidth, and to reduce energy consumption on wireless settings, has not yet been truly reached. The main limitation to fully exploit ETC's great traffic reductions lies on the difficulty to predict the occurrence of controller updates, forcing the use of conservative scheduling approaches in practice. Having a model of the timing behaviour of ETC is of paramount importance to enable the construction of model-based schedulers for such systems. Furthermore, on wireless control systems these schedulers allow to tightly schedule listening times, thus reducing energy consumption. In this chapter we describe an approach to model ETC traffic employing ideas from the symbolic abstractions literature. The resulting models of traffic are timed-automata. We also discuss briefly how these models can be employed to automatically synthesize schedulers.

M. Mazo Jr. (✉) · A. Sharifi-Kolarijani · C. Hop
Delft University of Technology, Delft, The Netherlands
e-mail: m.mazo@tudelft.nl

A. Sharifi-Kolarijani
e-mail: a.sharifikolarijani@tudelft.nl

C. Hop
e-mail: c.m.j.hop@student.tudelft.nl

D. Adzkiya
Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia
e-mail: dieky@matematika.its.ac.id



S. Tarbouriech et al. (eds.), *Control Subject to Computational and Communication Constraints*, Lecture Notes in Control and Information Sciences 475, https://doi.org/10.1007/978-3-319-78449-6_10

## 10.1 Introduction

A surge of Event-Triggered control (ETC) implementation strategies has appeared in the last decade promising to alleviate the inefficiencies of periodic (time-triggered) controller designs. Periodic controller implementations abuse the computing and communication infrastructures employing periodic feedback independently of the current state of the system. In ETC these inefficiencies are mitigated by letting the sensors decide (employing their limited computation capabilities) whether a measurement is worth transmitting for the computation of corrective actions. This results in aperiodic transmissions of measurements and computation of corrective actions. The amount of transmissions is often orders of magnitude smaller than the amount required to achieve a similar performance with periodic control.

However, this potential of aperiodic Event-Triggered techniques, to release channel capacity and reduce energy consumption on wireless settings, has not yet been truly reached. See e.g., [1] in which a tenfold traffic reduction resulted in barely a 57% energy consumption reduction of a wireless sensing infrastructure. The main limitation to fully exploit ETC's great traffic reductions lies on the difficulty to predict the occurrence of controller updates. Having a model of the timing behaviour of ETC is therefore of paramount importance to enable the construction of model-based schedulers for such systems, and to tightly schedule listening times on wireless communications to reduce energy consumption.

Different from controller/scheduler co-design approaches, see e.g., [2–4], we suggest to retain the separation of concerns between controller design and scheduling by defining a proper interface between these two realms. We propose a formal approach to derive models that capture the timing behaviour (of controller updates) of a family of Event-Triggered strategies for Linear Time-Invariant systems [5]. The constructed models provide an over-approximation of all the updates' timing behaviours generated by the aperiodic ETC system. Then, techniques from games over timed automata (TA) can be leveraged to synthesise schedulers [6].

Inspired by the state-dependent sampling proposed in [7], we employ a two-step approach to compute sampling intervals associated to states: first, the state space is partitioned (abstracted) into a finite number of convex polyhedral cones (pointed at the origin); then, for each conic region the time interval in which events can be originated is computed using a convex embedding approach [8] and Linear Matrix Inequalities derived from Lyapunov conditions. In the resulting timing models, transitions among discrete states (associated to the conic regions) are derived through reachability analysis over the sampling intervals computed earlier, see e.g., [9].

Furthermore, we show that the resulting models of traffic can be alternatively encoded as TA [10]. This allows us to additionally address the problem of scheduling the access to a shared resource by multiple ETC systems by solving games over TA [6, 11]. To this end, we enrich the constructed models of ETC traffic with actions that trade communication traffic for control performance. Then, a scheduler is synthesised as a strategy providing actions (for one player) that prevents the set of ETC tasks and

shared resource (the other player) from entering in a conflict situation. Useful classes of games for such a synthesis are readily solvable in tools like UPPAAL-TIGA [12].

## 10.2 Mathematical Preliminaries

We start by introducing some common notation employed throughout the chapter, and the basic theoretical notions of finite-state abstractions, timed automata, and Event-Triggered control that support the remainder of the chapter.

### *10.2.1 Notation*

We use $\mathbb{R}^n$ to denote the $n$-dimensional Euclidean space, $\mathbb{R}^+$ and $\mathbb{R}_0^+$ to denote the positive and nonnegative reals, respectively, $\mathbb{N}$ is the set of positive integers, and $\mathbb{IR}^+$ is the set of all closed intervals $[a, b]$ such that $a, b \in \mathbb{R}^+$ and $a \leqslant b$. For any set $S$, we denote by $2^S$ the set of all subsets of $S$, i.e., the power set of $S$. The sets of all $m \times n$ real-valued matrices and the set of all $n \times n$ real-valued symmetric matrices are denoted by $\mathscr{M}_{m\times n}$ and $\mathscr{M}_n$, respectively. Given a matrix $M$, $M \preceq 0$ (or $M \succeq 0$) indicates that $M$ is a negative (or positive) semidefinite matrix and $M \prec 0$ (or $M \succ 0$) denotes $M$ is a negative (positive) definite matrix. For a given matrix $M$, we denote by $[M]_{(i,j)}$ its $i$-th row, $j$-th column entry. The largest integer not greater than $x \in \mathbb{R}$ is denoted by $\lfloor x \rfloor$ and $|y|$ denotes the Euclidean norm of a vector $y \in \mathbb{R}^n$. Given two sets $Z_a$ and $Z_b$, every relation $Q \subseteq Z_a \times Z_b$ admits $Q^{-1} = \{(z_b, z_a) \in Z_b \times Z_a | (z_a, z_b) \in Q\}$ as its inverse relation. For $Q \subseteq Z \times Z$, an equivalence relation on a set $Z$, $[z]$ denotes the equivalence class of $z \in Z$ and $Z/Q$ denotes the set of all equivalence classes. For a set $A \subseteq \mathbb{R}^n$ we denote its Lebesgue measure by $\mu(A)$.

Given an ordinary differential equation of the form $\dot{\xi}(t) = f(\xi(t))$, admitting a unique solution, we denote by $\xi_x : \mathbb{R}_0^+ \to \mathbb{R}^n$ the solution to the initial value problem with $\xi_x(0) = x$. Finally, we also employ the notion of *flow pipe*:

**Definition 10.1** (*Flow Pipe* [9]) The set of reachable states, or flow pipe, from an initial set $X_{0,s}$ in the time interval $[\underline{\tau}_s, \bar{\tau}_s]$ is denoted by:

$$\mathscr{X}_{[\underline{\tau}_s, \bar{\tau}_s]}(X_{0,s}) = \bigcup_{t\in[\underline{\tau}_s, \bar{\tau}_s]} \mathscr{X}_t(X_{0,s}) = \bigcup_{t\in[\underline{\tau}_s, \bar{\tau}_s]} \{\xi_{x_0}(t) \,|\, x_0 \in X_{0,s}\}. \tag{10.1}$$

### 10.2.2 Symbolic Abstractions

We revise in the following the framework from [13] to relate different models of a system.

**Definition 10.2** (*Generalized Transition System* [13])
A system $S$ is a tuple $(X, X_0, U, \longrightarrow, Y, H)$ consisting of:

- a set of states $X$;
- a set of initial states $X_0 \subseteq X$;
- a set of inputs $U$;
- a transition relation $\longrightarrow \subseteq X \times U \times X$;
- a set of outputs $Y$;
- an output map $H : X \to Y$.

We say a system is finite-state (or infinite-state) when $X$ is a finite (or infinite) set, and that the system $S$ is metric if the output set $Y$ is a metric space (with some appropriately defined metric).

**Definition 10.3** (*Approximate Simulation Relation* [13]) Consider two metric systems $S_a$ and $S_b$ with $Y_a = Y_b$, and let $\varepsilon \in \mathbb{R}_0^+$. A relation $R \subseteq X_a \times X_b$ is an $\varepsilon$-approximate simulation relation from $S_a$ to $S_b$ if the following three conditions are satisfied:

1. $\forall x_{a0} \in X_{a0}, \exists x_{b0} \in X_{b0}$ such that $(x_{a0}, x_{b0}) \in R$;
2. $\forall (x_a, x_b) \in R$ we have $d(H_a(x_a), H_b(x_b)) \leqslant \varepsilon$;
3. $\forall (x_a, x_b) \in R$ for all $(x_a, u_a, x_a') \in \underset{a}{\longrightarrow}, \exists (x_b, u_b, x_b') \in \underset{b}{\longrightarrow}$ satisfying $(x_a', x_b') \in R$.

Whenever an $\varepsilon$-approximate simulation relation from $S_a$ to $S_b$ exists we write $S_a \preceq^{\varepsilon} S_b$, and say that $S_b$ $\varepsilon$-approximately simulates $S_a$. Intuitively, under some technical conditions, $S_a \preceq^{\varepsilon} S_b$ implies that all possible output sequences that $S_a$ can produce are contained in the set of output sequences that $S_b$ can generate.

Let us also introduce the following alternative notion of *quotient system* (see e.g., [13] for the traditional definition):

**Definition 10.4** (*Power Quotient System* [5]) Let $S = (X, X_0, U, \longrightarrow, Y, H)$ be a system and $R$ be an equivalence relation on $X$. The power quotient of $S$ by $R$, denoted by $S_{/R}$, is the system $(X_{/R}, X_{/R,0}, U_{/R}, \underset{/R}{\longrightarrow}, Y_{/R}, H_{/R})$ consisting of:

- $X_{/R} = X/R$;
- $X_{/R,0} = \{x_{/R} \in X_{/R} | x_{/R} \cap X_0 \neq \varnothing\}$;
- $U_{/R} = U$;
- $(x_{/R}, u, x'_{/R}) \in \underset{/R}{\longrightarrow}$ if $\exists (x, u, x') \in \longrightarrow$ in $S$ with $x \in x_{/R}$ and $x' \in x'_{/R}$;
- $Y_{/R} \subset 2^Y$;
- $H_{/R}(x_{/R}) = \underset{x \in x_{/R}}{\cup} H(x)$.

In the case considered in this chapter we rarely are able to compute such power quotient systems. In particular, the transition relation and output maps in general need to be over-approximated. We introduce the following relaxed version of the previous definition, followed by a Lemma establishing the relation between such quotient systems and the original concrete system.

**Definition 10.5** (*Approximate Power Quotient System* [5])
Let $S = (X, X_0, U, \longrightarrow, Y, H)$ be a system, $R$ be an equivalence relation on $X$, and $S_{/R} = (X_{/R}, X_{/R,0}, U_{/R}, \underset{/R}{\longrightarrow}, Y_{/R}, H_{/R})$ be the power quotient of $S$ by $R$.

An approximate power quotient of $S$ by $R$, denoted by $\bar{S}_{/R}$, is a system $(X_{/R}, X_{/R,0}, U_{/R}, \underset{/\bar{R}}{\longrightarrow}, \bar{Y}_{/R}, \bar{H}_{/R})$ such that:

- $\underset{/\bar{R}}{\rightarrow} \supseteq \underset{/R}{\rightarrow}$,
- $\bar{Y}_{/R} \supseteq Y_{/R}$, and
- $\bar{H}_{/R}(x_{/R}) \supseteq H_{/R}(x_{/R}), \forall x_{/R} \in X_{/R}$.

**Lemma 10.1** [5] *Let $S$ be a metric system, $R$ be an equivalence relation on $X$, and let the metric system $\bar{S}_{/R}$ be the approximate power quotient system of $S$ by $R$. For any*

$$\varepsilon \geqslant \max_{\substack{x \in x_{/R} \\ x_{/R} \in X_{/R}}} d(H(x), \bar{H}_{/R}(x_{/R})),$$

*with $d$ the Hausdorff distance over the set $2^Y$, $\bar{S}_{/R}$ $\varepsilon$-approximately simulates $S$, i.e., $S \preceq_S^{\varepsilon} \bar{S}_{/R}$.*

For any set $Y$, $Y \in 2^Y$, which allows us to employ the Hausdorff distance [14] as a common metric for output sets of the power quotient and the original system.

### *10.2.3 Timed Safety and Timed Game Automata*

The abstraction methodology we propose results in models semantically equivalent to Timed Safety Autamata (TSA) [15]. TSA are a simplified version of the classical *timed automata* [10] (TA). While TA employ *Büchi-acceptance* conditions to specify progress properties, in TSA local invariant conditions are employed to this same end (see [16, Sect. 2] for a detailed discussion). Here, we just recall briefly the definition of TSA from [16]. Let $\Sigma$ be a finite alphabet of actions, and $\mathscr{C}$ a set of finitely many real-valued variables employed to represent clocks. Consider $\sim \in \{>, \geqslant, <, \leqslant\}$, a clock constraint $\delta$ is a conjunctive formula of atomic constraints $c_1 \sim k$ or $c_1 - c_2 \sim k$ for $c_1, c_2 \in \mathscr{C}$, and $k \in \mathbb{N}$. We employ $\mathscr{B}(\mathscr{C})$ to denote the set of all possible clock constraints.

**Definition 10.6** (*Timed Safety Automata* [16]) A *timed safety automata* is a tuple $\mathscr{A} = (L, L_0, \Sigma, \mathscr{C}, E, I)$ where

- $L$ is a finite set of locations (or discrete states);
- $L_0 \subseteq L$ is a set of start locations;
- $\Sigma$ is the set of actions;
- $\mathscr{C}$ is the set of clocks;
- $E \subseteq L \times \mathscr{B}(\mathscr{C}) \times \Sigma \times 2^{\mathscr{C}} \times L$ is the set of transitions.
- $I : L \longrightarrow \mathscr{B}(\mathscr{C})$ assigns invariants to locations.

The shorthand notation $l \xrightarrow{g,a,r} l'$ is used to denote $(l, g, a, r, l') \in E$, i.e., a transition from state $l$ to state $l'$ under input symbol $a$, with $r \subseteq \mathscr{C}$ the set of clocks reset when this transition is taken, and a clock constraint $g$ over $\mathscr{C}$ as the guard enabling this transition.

**Definition 10.7** (*Operational Semantics* [16]) The semantics of a timed safety automaton is a transition system (also known as timed transition system) where states are pairs $(l, u)$, with $l \in L$ and $u$ a clock valuation, and transitions are defined by the rules:

- $(l, u) \xrightarrow{d} (l, u + d)$ if $u \models I(l)$ and $(u + d) \in I(l)$ for a scalar $d \in \mathbb{R}^+$;
- $(l, u) \xrightarrow{a} (l', u')$ if $l \xrightarrow{g,a,r} l'$, $u \models g$, $u' = [r \to \mathbf{0}]u$ and $u' \models I(l')$.

*Remark 10.1* In a TSA, the *guards* and *invariants* assert necessary and sufficient conditions respectively for transitions to take place. The sufficient conditions established by *invariants* must not be violated by letting time advance. Therefore, invariants establish upper bounds for the time to take the next transition [15].

*Remark 10.2* Note that a timed automaton is a particular class of hybrid automata in which the only allowed continuous dynamics are of the form $\dot{c} = 1$, and in which *guard* and *invariant* sets are in the form of clock constraints.

TSA evolve over uncountable state spaces, due to its clock variables. Nonetheless, it has been shown that its reachability analysis is decidable [10]. This decidability allows the development of powerful tools for verification and synthesis [10, 17], which can be used to generate schedulers for real-time systems, whose timing is modelled as TSAs [6].

Timed Game Automata (TGA) are an extension of TSA where the set of actions is partitioned into controllable actions (activated by the controller) and uncontrollable actions (activated by the environment or an opponent).

**Definition 10.8** (*Timed Game Automaton* [18]) A timed game automaton is a tuple $\mathscr{G} = (L, L_0, \Sigma_c, \Sigma_u, \mathscr{C}, E, I)$ where

- $(L, L_0, \Sigma_c \cup \Sigma_u, \mathscr{C}, E, I)$ is a timed safety automaton;
- $\Sigma_c$ is a set of controllable actions;
- $\Sigma_u$ is a set of uncontrollable actions;
- $\Sigma_c \cap \Sigma_u = \emptyset$.

A feature of TGA is its modularity. Constructing a TGA for complex systems can be done by constructing a TGA for each part and then combining (or composing) them. The resulting object is denoted a Network of TGA (NTGA), and is constructed through a synchronized parallel composition in which uncontrollable inputs of a TGA are linked to outputs of another TGA. For a more detailed discussion of such composition we refer the reader to [19].

### *10.2.4 Event-Triggered Control for LTI Systems*

We describe next a basic framework for Event-Triggered control in the case of Linear Time Invariant (LTI) control systems with state-feedback. The techniques we present in the remainder of the paper only focus on this class of control systems. Many extensions to this simple framework have been proposed, see for instance the rest of the papers in Part 2 of this book and references therein. See the conclusion of the chapter for a brief discussion on generalizing the current results.

Consider an LTI system without disturbances:

$$\dot{\xi}(t) = A\xi(t) + B\upsilon(t),\ \xi(t) \in \mathbb{R}^n, \upsilon(t) \in \mathbb{R}^m \tag{10.2}$$

and a linear state-feedback controller implemented in a sample-and-hold fashion:

$$\upsilon(t) = \upsilon(t_k) = K\xi(t_k),\ \forall t \in [t_k, t_{k+1}),\ k \in \mathbb{N}. \tag{10.3}$$

The following quadratic triggering mechanism:

$$t_{k+1} := \inf\{t > t_k |\ |\xi(t_k) - \xi(t)|^2 \geqslant \alpha|\xi(t)|^2\}, \tag{10.4}$$

with $\alpha \in \mathbb{R}^+$ a design parameter properly selected, renders the closed-loop system asymptotically stable [20]. Let us denote the inter-sample time associated to a state by:

$$\tau(x) := t_{k+1} - t_k, \quad \text{with } x = \xi(t_k). \tag{10.5}$$

For LTI systems, the solutions $\xi$ in some time interval $[t_k, t_k + \sigma]$ can be easily expressed in terms of the initial condition:

$$\xi(t_k + \sigma) = \Lambda(\sigma)\xi(t_k), \tag{10.6}$$

$$\Lambda(\sigma) = [I + \int_0^\sigma e^{Ar} dr(A + BK)]. \tag{10.7}$$

Thus, the state-dependent inter-sampling times can be rewritten as:

$$\tau(x) = \inf\{\sigma > 0 |\ x^T \Phi(\sigma) x \geqslant 0\}, \tag{10.8}$$

$$\Phi(\sigma) = [I - \Lambda^T(\sigma)][I - \Lambda(\sigma)] - \alpha \Lambda^T(\sigma)\Lambda(\sigma). \tag{10.9}$$

## 10.3 Timing Abstractions of Event-Triggered Control Systems

We are interested in obtaining models capturing the evolution over time of the inter-sample times generated by an ETC loop. Such dynamics are actually provided by the following system:

$$S = (X, X_0, U, \longrightarrow, Y, H)$$

where

- $X = \mathbb{R}^n$;
- $X_0 \subseteq \mathbb{R}^n$;
- $U = \varnothing$, i.e., the system is autonomous;
- $\longrightarrow \in X \times U \times X$ such that $\forall x, x' \in X : (x, x') \in \longrightarrow$ iff $\xi_x(\tau(x)) = x'$;
- $Y \subset \mathbb{R}^+$;
- $H : \mathbb{R}^n \to \mathbb{R}^+$ where $H(x) = \tau(x)$.

The system $S$ generates as output sequences all possible sequences of inter-sampling intervals that a given ETC loop can exhibit. However, $S$ is an infinite-state system and the map $H$ (a copy of $\tau$) is not an explicit function.

**Problem 10.1** We seek to construct finite-state systems capturing, up to some computable precision, all the possible traffic patterns of an ETC system, i.e., all possible sequences $\{\tau(\xi(t_k))\}_{k\in\mathbb{N}}$.

In order to solve this problem, we propose to abstract the system $S$ by a power quotient system $S_{/R}$ as follows:

$$S_{/R} = (X_{/R}, X_{0/R}, U_{/R}, \underset{/R}{\longrightarrow}, Y_{/R}, H_{/R})$$

where

- $X_{/R} = \mathbb{R}^n_{/R} := \{\mathscr{R}_1, \ldots, \mathscr{R}_q\}$;
- $X_{/R,0} = \{\mathscr{R}_i \mid X_0 \cap \mathscr{R}_i \neq \emptyset\}$;
- $U_{/R} = \varnothing$, i.e., the system is autonomous;
- $(x_{/R}, x'_{/R}) \in \underset{/R}{\longrightarrow}$ if $\exists x \in x_{/R}, \exists x' \in x'_{/R}$ such that $\xi_x(H(x)) = x'$;
- $Y_{/R} \subset 2^Y \subset \mathbb{IR}^+$;
- $H_{/R}(x_{/R}) = [\inf\limits_{x\in x_{/R}} H(x),\ \sup\limits_{x\in x_{/R}} H(x)] := [\underline{\tau}_{x_{/R}}, \bar{\tau}_{x_{/R}}]$.

In general, constructing such a power quotient system is not possible, thus we focus on constructing an approximate power quotient system $\bar{S}_{/R}$.

*Remark 10.3* By Lemma 10.1 we know that $S \preceq^{\varepsilon}_{S} \bar{S}_{/R}$. This can be interpreted as the (approximate) power quotient system producing output sequences $\{T_k\}_{k\in\mathbb{N}}$, $T_k \in \mathbb{R}^+$, such that $\tau(\xi(t_k)) \in T_k$. The fact that all the possible *timing sequences* $\{t_{k+1} - t_k\}_{k\in\mathbb{N}}$ of the ETC system, i.e., output sequences of the infinite system $S$, are captured by $\bar{S}_{/R}$ allows us to employ these abstractions to synthesize *schedulers*, cf. Sect. 10.4.

In the remaining of this section we describe how to select an appropriate equivalence relation $R$, compute the intervals $[\underline{\tau}_{x_{/R}}, \bar{\tau}_{x_{/R}}]$, and determine the transition relation $\underset{/R}{\longrightarrow}$.

### 10.3.1 State Set

In the traditional construction of quotient systems, one bundles together states that produce the same output. In our proposed construction of power quotient systems this is no longer the case, but the precision $\varepsilon$ achieved depends on how close are the outputs of states bundled together, cf. Lemma 10.1. In the case of LTI Event-Triggered systems one can easily characterize states that produce the same output, i.e., states $x, x'$ such that $\tau(x) = \tau(x')$, see e.g., [7, 21]:

**Proposition 10.1** *States lying on the same ray crossing the origin have the same inter-sample time, i.e.,* $\tau(x) = \tau(\lambda x)$, $\forall \lambda \neq 0, x \neq 0$.

Hence, to construct a quotient system of $S$, the abstract states need to be rays in the state space at hand. But there is an infinite number of rays, thus, in order to obtain a finite state abstraction, we suggest to take as abstract states unions of an infinite number of such rays. In particular, polyhedral cones pointed at the origin are a choice which makes the construction of finite state-space partitions relatively easy. We denote such cones by $\mathscr{R}_s$ where $s \in \{1, \ldots, q\}$ and $\bigcup_{s=1}^{q} \mathscr{R}_s = \mathbb{R}^n$ (see Fig. 10.1 for an example in $\mathbb{R}^2$).

In order to construct such a partition of the state-space, we use a so called *isotropic covering*. Consider first the case of partitioning $\mathbb{R}^2$ via cones pointed at the origin. This is easily achieved by first splitting the interval $\Theta = [-\frac{\pi}{2}, \frac{\pi}{2})$ uniformly in a number of sub-intervals $\Theta_s = [\underline{\theta}_s, \overline{\theta}_s)$. Then for each of those intervals one can construct the corresponding cone as:

$$\mathscr{R}_s = \{x \in \mathbb{R}^2 \mid x^T Q_s x \geqslant 0\}, \quad Q_s = \frac{1}{2}\begin{bmatrix} -2\sin\underline{\theta}_s \sin\overline{\theta}_s & \sin(\underline{\theta}_s + \overline{\theta}_s) \\ \sin(\underline{\theta}_s + \overline{\theta}_s) & -2\cos\underline{\theta}_s \cos\overline{\theta}_s \end{bmatrix}. \tag{10.10}$$

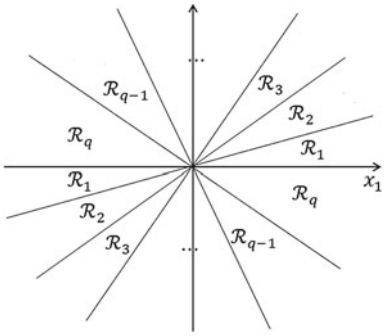


**Fig. 10.1** Example of a state-space partitioning with polyhedral cones in $\mathbb{R}^2$ [5]

*Remark 10.4* Note that even though it may look like we only partition in this form half of the space (as we only ranged in the polar coordinates between $[-\frac{\pi}{2}, \frac{\pi}{2})$ radians), in reality the other half space is covered by this same $\mathscr{R}_s$ sets. To see this, just observe that a point defined in polar coordinates by the pair $(r, \theta)$, i.e., $x_1 = r\cos\theta,\ x_2 = r\sin\theta$, and the point $(-r, \theta)$ (or alternatively $(r, \theta + \pi)$) belong to the same set $\mathscr{R}_s$ as $x^T Q_s x = (-x)^T Q_s (-x)$. Furthermore, this poses no problem in terms of the times associated to the set as $\tau(x) = \tau(-x)$ from Proposition 10.1.

We can generalize this partitioning approach to cover arbitrary higher dimensions as follows. Consider a point $x = [x_1,\ x_2, \ldots x_n]^T \in \mathbb{R}^n$, and define the projection of that point on its $i - j$ coordinates as $(x)_{(i,j)} = (x_i, x_j)$. Now, let the sets defining the partition of the state-space to be defined as:

$$\mathscr{R}_{(s_1,s_2,\ldots s_{n-1})} = \left\{ x \in \mathbb{R}^n \mid \bigwedge_{i=1}^{n-1} (x)_{(i,i+1)}^T Q_{s_i} (x)_{(i,i+1)} \geqslant 0 \right\}. \tag{10.11}$$

By ranging over all possible indices $s = (s_1, s_2, \ldots s_{n-1}) \in \{1, 2, \ldots, m\}^{n-1}$ the whole state space can be covered. Here $m$ denotes the number of intervals employed to subdivide $[-\frac{\pi}{2}, \frac{\pi}{2})$ in constructing the $Q_s$ matrices.

The equivalence relation $R \subseteq \mathbb{R}^n \times \mathbb{R}^n$ in Definition 10.5 is thus given by $(x, x') \in R \Leftrightarrow x, x' \in \mathscr{R}_s$, for some $s$.

### *10.3.2 Output Map*

Constructing the output map $\bar{H}_{/R}$ (and the associated output set $\bar{Y}_{/R}$) boils down to computing the time intervals $[\underline{\tau}_s, \bar{\tau}_s]$ such that $\forall x \in \mathscr{R}_s : \tau(x) \in [\underline{\tau}_s, \bar{\tau}_s]$. In other words, we need to compute lower and upper bounds on the inter-sample times that can be observed for different states (among the infinite number) in a region $\mathscr{R}_s$.

Employing the state transition matrix in (10.9), one can express a necessary condition for $\underline{\tau}$ to be a lower bound of $\tau(x)$ as:

$$x^T \Phi(\sigma)x \leqslant 0, \forall \sigma \in [0, \underline{\tau}] \Rightarrow \underline{\tau}_s \leqslant \tau(x).$$

Note that this condition involves a matrix functional $\Phi(\sigma)$ and thus it cannot be directly checked. To address this issue, we employ the approach from Hetel et al. [8] to construct a convex polytope (in the space of matrices) containing $\Phi(\sigma)$. Then, employing convexity, one can replace the condition involving an infinite number of matrices $\Phi(\sigma)$ by a finite (and thus computable) set of inequalities involving only a finite set of matrices $\underline{\Phi}_\kappa$, with $\kappa \in \mathscr{K}$, i.e., :

$$(x^T \underline{\Phi}_\kappa x \leqslant 0, \forall \kappa \in \mathscr{K}) \Longrightarrow (x^T \Phi(\sigma)x \leqslant 0, \forall \sigma \in [0, \underline{\tau}_s]). \tag{10.12}$$

**Assumption 10.1** Assume that a scalar $\bar{\sigma} > 0$ exists such that $x^T \Phi(\bar{\sigma})x \geqslant 0, \forall x \in \mathbb{R}^n$.

*Remark 10.5* This constant $\bar{\sigma}$ is a global upper bound for the inter-sample times. It can be computed through a line search until the matrix $\Phi(\bar{\sigma})$ becomes positive definite. In general, such an upper-bound may not exist. Think e.g., of a stable real eigenvector $v$ of the open-loop system and a controller setting the control action for $u = Kv = 0$, in this case the system may not trigger new controller updates. A simple solution to this issue is to modify the triggering condition by fixing a certain upper bound for the triggering times $\bar{\sigma}$:

$$\tau(x) = \min\{\bar{\sigma}, \sigma > 0 |\ x^T \Phi(\sigma)x \geqslant 0\}. \tag{10.13}$$

The following Lemma provides the construction of a finite number of LMIs to numerically check condition (10.12). Consider a positive integer $N_{conv} \geqslant 0$ such that $N_{conv} + 1$ is the number of vertices of the polytope employed to cover $\Phi(\sigma)$ in the time interval $[0, \bar{\sigma}]$, and an integer number $l \geqslant 1$ for the number of intervals in which to divide the cover, see Fig. 10.2 for an intuitive illustration.

**Lemma 10.2** [5] *Consider a time bound* $\underline{\tau} \in (0, \bar{\sigma}]$. *If* $x^T \underline{\Phi}_{(i,j),\underline{\tau}} x \leqslant 0$ *holds* $\forall (i, j) \in \mathscr{K}_{\underline{\tau}} = (\{0, \ldots, N_{conv}\} \times \{0, \ldots, \lfloor \frac{\tau l}{\bar{\sigma}} \rfloor\})$, *then:*

$$x^T \Phi(\sigma)x \leqslant 0, \quad \forall \sigma \in [0, \underline{\tau}]$$

*with* $\Phi$ *defined in* (10.9) *and*

$$\underline{\Phi}_{(i,j),\underline{\tau}} = \hat{\underline{\Phi}}_{(i,j),\underline{\tau}} + \underline{\nu} I,$$

$$\hat{\underline{\Phi}}_{(i,j),\underline{\tau}} = \begin{cases} \sum_{k=0}^{i} L_{k,j} (\frac{\bar{\sigma}}{l})^k & \text{if } j < \lfloor \frac{\tau l}{\bar{\sigma}} \rfloor, \\ \sum_{k=0}^{i} L_{k,j} (\underline{\tau} - \frac{j\bar{\sigma}}{l})^k & \text{if } j = \lfloor \frac{\tau l}{\bar{\sigma}} \rfloor, \end{cases}$$

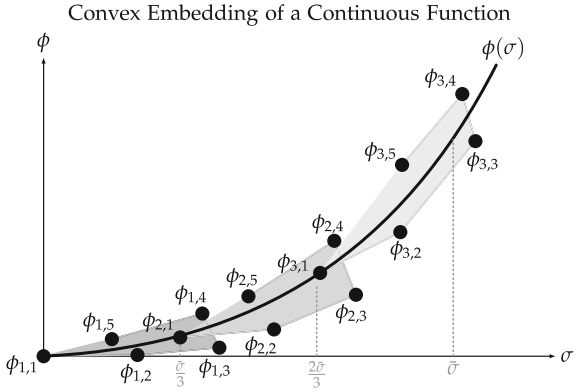


**Fig. 10.2** Polytopic bounding of a (scalar) exponential function, with $N_{conv} = 4$ and $l = 3$

$$\begin{cases} L_{0,j} &= I - \Pi_{1,j} - \Pi_{1,j}^T + (1-\alpha)\Pi_{1,j}^T\Pi_{1,j}, \\ L_{1,j} &= [(1-\alpha)\Pi_{1,j}^T - I]\Pi_{2,j} \\ &\quad +\Pi_{2,j}^T[(1-\alpha)\Pi_{1,j} - I], \\ L_{k\geqslant 2,j} &= [(1-\alpha)\Pi_{1,j}^T - I]\frac{A^{k-1}}{k!}\Pi_{2,j} \\ &\quad +\Pi_{2,j}^T\frac{(A^{k-1})^T}{k!}[(1-\alpha)\Pi_{1,j} - I] \\ &\quad +(1-\alpha)\Pi_{2,j}^T(\sum_{i=1}^{k-1}\frac{(A^{i-1})^T}{i!}\frac{A^{k-i-1}}{(k-i)!})\Pi_{2,j}, \end{cases} \tag{10.14}$$

$$\begin{cases} \Pi_{1,j} &= I + M_j(A+BK), & M_j &= \int_0^{j\frac{\bar{\sigma}}{l}} e^{As}ds, \\ \Pi_{2,j} &= N_j(A+BK), & N_j &= AM_j + I, \end{cases} \tag{10.15}$$

$$\underline{\nu} \geqslant \max_{\substack{\sigma'\in[0,\frac{\bar{\sigma}}{l}] \\ r\in\{0,\ldots,l-1\}}} \lambda_{\max}(\Phi(\sigma' + r\tfrac{\bar{\sigma}}{l}) - \tilde{\Phi}_{N_{conv},r}(\sigma')), \tag{10.16}$$

$$\tilde{\Phi}_{N_{conv},r}(\sigma) = \sum_{k=0}^{N_{conv}} L_{k,r}\sigma^k. \tag{10.17}$$

For a given state $x$ we can now employ this result to compute a lower bound on $\tau(x)$. In order to compute a lower bound $\underline{\tau}_s$ for the bundle of states defined by a conic region $\mathscr{R}_s$, one can leverage the S-procedure as in the following theorem. Before stating the result, we need to define some new set of matrices $\tilde{Q}_s^{(i,j)}$ as:

$$\tilde{Q}_s^{(i,j)} \in \mathscr{M}_n, \text{ such that } \begin{cases} [\tilde{Q}_s^{(i,j)}]_{(i,i)} = [Q_s]_{(1,1)} \\ [\tilde{Q}_s^{(i,j)}]_{(i,j)} = [Q_s]_{(1,2)} \\ [\tilde{Q}_s^{(i,j)}]_{(j,i)} = [Q_s]_{(2,1)} \\ [\tilde{Q}_s^{(i,j)}]_{(j,j)} = [Q_s]_{(2,2)} \\ [\tilde{Q}_s^{(i,j)}]_{(k,l)} = 0 \qquad \text{otherwise} \end{cases} \tag{10.18}$$

where $Q_s \in \mathscr{M}_2$ are as defined in (10.10).

**Theorem 10.2** (Regional Lower Bound Approximation) *Consider a scalar* $\underline{\tau}_s \in (0, \bar{\sigma}]$ *and matrices* $\underline{\Phi}_{\kappa,\underline{\tau}_s}$, $\kappa = (i, j) \in \mathscr{K}_{\underline{\tau}_s}$, *defined as in Lemma 10.2. If there exist scalars* $\underline{\varepsilon}_{\kappa,\underline{\tau}_s} \geqslant 0$ *such that for all* $\kappa \in \mathscr{K}_{\underline{\tau}_s}$ *the following LMIs hold:*

$$\underline{\Phi}_{\kappa,\underline{\tau}_s} + \sum_{i=1}^{n-1} \underline{\varepsilon}_{\kappa,s_i} \tilde{Q}_{s_i}^{(i,i+1)} \preceq 0$$

*the inter-sample time* (10.4) *of the system* (10.2)–(10.3) *is regionally bounded from below by* $\underline{\tau}_s$, $\forall x \in \mathscr{R}_s$.

*Proof* The proof is verbatim the proof on [5], replacing the linear representation of conic partitions in dimensions higher than two, by the alternative quadratic representation of (10.11).

Similarly, one can compute upper bounds $\bar{\tau}_s$ of the inter-sample time for a conic region, employing Lemma 10.3 and Theorem 10.3.

**Lemma 10.3** [5] *Consider a time bound* $\bar{\tau} \in [\underline{\tau}, \bar{\sigma}]$. *If* $x^T \bar{\Phi}_{(i,j),\bar{\tau}} x \geqslant 0$ *holds* $\forall (i, j) \in \mathscr{K}_{\bar{\tau}} = (\{0, \ldots, N_{conv}\} \times \{\lfloor \frac{\bar{\tau} l}{\bar{\sigma}} \rfloor, \ldots, l-1\})$, *then:*

$$x^T \Phi(\sigma) x \geqslant 0, \quad \forall \sigma \in [\bar{\tau}, \bar{\sigma}]$$

*with* $\Phi$ *defined in* (10.9) *and:*

$$\bar{\Phi}_{(i,j),\bar{\tau}} = \bar{\hat{\Phi}}_{(i,j),\bar{\tau}} + \bar{\nu} I,$$

$$\bar{\hat{\Phi}}_{(i,j),\bar{\tau}} = \begin{cases} \sum_{k=0}^{i} L_{k,j} (\frac{(j+1)\bar{\sigma}}{l} - \bar{\tau})^k & \text{if } j = \lfloor \frac{\bar{\tau} l}{\bar{\sigma}} \rfloor, \\ \sum_{k=0}^{i} L_{k,j} (\frac{\bar{\sigma}}{l})^k & \text{if } j > \lfloor \frac{\bar{\tau} l}{\bar{\sigma}} \rfloor, \end{cases}$$

$$\bar{\nu} \leqslant \max_{\substack{\sigma' \in [0, \frac{\bar{\sigma}}{l}] \\ r \in \{0, \ldots, l-1\}}} \lambda_{\min}(\Phi(\sigma' + r\tfrac{\bar{\sigma}}{l}) - \tilde{\Phi}_{N_{conv},r}(\sigma')), \tag{10.19}$$

*where* $L_{k,j}$ *and* $\tilde{\Phi}_{N_{conv},r}$ *are given by* (10.14) *and* (10.17), *respectively.*

**Theorem 10.3** (Regional Upper Bound Approximation) *Consider a scalar* $\bar{\tau}_s \in (0, \bar{\sigma}]$ *and matrices* $\bar{\Phi}_{\kappa,\bar{\tau}_s}$, $\kappa = (i, j) \in \mathscr{K}_{\bar{\tau}_s}$, *defined as in Lemma 10.2. If there exist scalars* $\bar{\varepsilon}_{\kappa,\bar{\tau}_s} \geqslant 0$ *such that for all* $\kappa \in \mathscr{K}_{\bar{\tau}_s}$ *the following LMIs hold:*

$$\bar{\Phi}_{\kappa,\bar{\tau}_s} + \sum_{i=1}^{n-1} \bar{\varepsilon}_{\kappa,s_i} \tilde{Q}_{s_i}^{(i,i+1)} \preceq 0$$

*the inter-sample time* (10.4) *of the system* (10.2)–(10.3) *is regionally bounded from above by* $\bar{\tau}_s$, $\forall x \in \mathscr{R}_s$.

*Proof* Analogous to the proof of Theorem 10.2.

*Remark 10.6* In order to employ Theorem 10.2 (Theorem 10.3) to compute lower (upper) bounds for each of the regions, one needs to apply a line search over $\underline{\tau}_s \in [0, \bar{\sigma}]$ ($\bar{\tau}_s \in [\underline{\tau}_s, \bar{\sigma}]$). This requires checking the feasibility of the LMIs at each step of the line search.

### *10.3.3 Transition Relation*

Finally, the transition relation $\underset{/R}{\longrightarrow}$ of the abstraction is given by:

$$(x_{/R}, x'_{/R}) \in \underset{/R}{\longrightarrow} \Leftrightarrow \mu(\mathscr{X}_{[\underline{\tau}_s, \bar{\tau}_s]}(x_{/R}) \cap x'_{/R}) > 0. \tag{10.20}$$

*Remark 10.7* Equation (10.20) explicitly enforces that the intersection between the sets needs to be strictly larger than just the trivial coincidence in the origin, or one facet of the sets, by requiring that such intersection has non-zero measure.

In other words, to construct the relation one needs to compute which sets $\mathscr{R}_{s'}$ are (non-trivially) intersected by $\mathscr{X}_{[\underline{\tau}_s, \bar{\tau}_s]}(\mathscr{R}_s)$: the reachable set from $\mathscr{R}_s$ in the time interval $[\underline{\tau}_s, \bar{\tau}_s]$. In practice, we can only compute approximations of this reachable set. Nevertheless, in order to construct an approximate abstraction $\bar{S}_{/R}$ it suffices to compute the intersection with outer approximations i.e., $\hat{\mathscr{X}}_{[\underline{\tau}_s, \bar{\tau}_s]}(\mathscr{R}_s)$ such that $\mathscr{X}_{[\underline{\tau}_s, \bar{\tau}_s]}(\mathscr{R}_s) \subseteq \hat{\mathscr{X}}_{[\underline{\tau}_s, \bar{\tau}_s]}(\mathscr{R}_s)$.

*Remark 10.8* Employing an outer approximation of the reachable sets can potentially introduce *spurious* transitions, i.e., $\underset{/R}{\longrightarrow} \subseteq \underset{\bar{/R}}{\longrightarrow}$ but as stated in Lemma 10.1 the desired approximate simulation relation is retained.

Note that $\mathscr{R}_s$ are not compact sets (they are unbounded cones). However, all of those cones share the origin which is an invariant point of the state space. Therefore, it is sufficient to compute the reachable set of $\underline{\mathscr{R}}_s := \mathscr{R}_s \cap \mathscr{E}_s$ for some affine hyperplane $\mathscr{E}_s = \{x | e^T x + c \leqslant 0\}$ with $e \in \mathbb{R}^n$ and $c \neq 0 \in \mathbb{R}$. Then from the convex hull of this polytope in $\mathbb{R}^{n-1}$ and the origin one can construct a non-empty convex subset $\hat{\mathscr{R}}_s$ of $\mathscr{R}_s$ as follows:

$$\hat{\mathscr{R}}_s = \{\lambda x_e \,|\, \lambda \in [0, 1],\ x_e \in \underline{\mathscr{R}}_s\}. \tag{10.21}$$

Now observe that, thanks to the linearity of (10.6) and the fact that all sets are pointed at the origin, the condition (10.20) can be replaced by:

$$\mu(\mathscr{X}_{[\underline{\tau}_s, \bar{\tau}_s]}(\mathscr{R}_s) \cap \mathscr{R}_{s'}) > 0 \Leftrightarrow \mu(\mathscr{X}_{[\underline{\tau}_s, \bar{\tau}_s]}(\hat{\mathscr{R}}_s) \cap \hat{\mathscr{R}}_{s'}) > 0 \tag{10.22}$$

There are many techniques available to compute polytopic outer approximations of reachable sets of polytopes as the set $\hat{\mathscr{X}}_{[\underline{\tau}_s, \bar{\tau}_s]}(\mathscr{R}_s)$. In particular, we employ in our

implementations the approach from [9]. Similarly, there are many tools that enable the computation of intersection of polytopic sets and check that such sets have no empty interior, e.g., [22, 23].

### *10.3.4 Increasing the Precision of the Abstractions*

The precision of the abstractions obtained can be considered from both: the distance of the output traces, i.e., $\varepsilon$ in Lemma 10.1, and in terms of the amount of spurious transitions introduced.

The conservatism introduced through the polytopic embedding, cf. Sect. 10.3.2 can be reduced by increasing $N_{conv}$ and $l$ in Lemmas 10.2 and 10.3. This results in more LMI constraints, but in general leads to a smaller $\varepsilon$ by reducing $|\bar{\tau}_s - \underline{\tau}_s|$ for each $\mathscr{R}_s$. Having tighter bounds for the inter-sample times also reduces the conservatism introduced in computing the reachable sets in Sect. 10.3.3, which in turn reduces the amount of spurious transitions. Similarly, one can employ more precise or tight outer approximations of the reachable sets to reduce spurious transitions.

Finally, one can also refine the conic regions of an abstraction $S_{/R}$ into more regions $\mathscr{R}_s$. As long as in the new abstraction $S_{/R'}$ the equivalence classes are subsets of the classes in the first abstraction $S_{/R}$, the precision of the inter-sample bounds cannot decrease, i.e., $|\bar{\tau}_s - \underline{\tau}_s|$ cannot increase. Formally:

$$\big(\forall(x, x') \in R' \Rightarrow (x, x') \in R\big) \Rightarrow \varepsilon' \leqslant \varepsilon, \tag{10.23}$$

where $S \preceq_S^{\varepsilon} \bar{S}_{/R}$, and $S \preceq_S^{\varepsilon'} \bar{S}_{/R'}$. Note that this does not need to hold if the partition defined by $R'$ is not a refinement of the original partition determined by $R$.

## 10.4 Timed Automata and Scheduling

In this section we briefly show that the abstractions $\bar{S}_{/R}$, whose construction is described in the previous sections, are in fact semantically equivalent to TSA. Then, we illustrate a few possibilities to enrich the obtained abstractions with *controllable actions*, which may be employed to design schedulers for ETC systems on shared resources.

Let us first interpret the semantics of the proposed abstractions $\bar{S}_{/R}$. Note that the system $\bar{S}_{/R}$ only captures discrete events. However, just like the concrete system $S$, the connection with actual time is established through the outputs produced by these models. The abstraction $\bar{S}_{/R}$ is a finite-state dynamical system, but with an infinite output set $\bar{Y}_{/R}$ capturing time intervals. When the last transmitted measurement $x$ satisfies $x \in x_{/R}$, the output $y_{/R} = \bar{H}(x_{/R})$ indicates that the original control system:

1. does not trigger updates during the interval $[0, \underline{\tau}_{x_{/R}})$;
2. may trigger a controller update during the time interval $[\underline{\tau}_{x_{/R}}, \bar{\tau}_{x_{/R}})$; and
3. must trigger an update if $\bar{\tau}_{x_{/R}}$ seconds have elapsed since the last transmission.

In the model $\bar{S}_{/R}$ a controller update of the ETC system (cf. Sect. 10.2.4) is captured by a transition between states $x_{/R} \rightarrow x'_{/R}$ of the abstraction $\bar{S}_{/R}$. As described in Sect. 10.2.3, one can capture the same type of semantics with a TSA. In particular, the TSA $\bar{S}_{TSA} = (L, L_0, \Sigma, \mathscr{C}, E, I)$ has the same semantics as the abstraction $\bar{S}_{/R}$, where:

- the set of locations $L := X_{/R} = \{l_1, \ldots, l_q\}$;
- the set of initial locations $L_0 := X_{/R,0}$;
- the set of actions $\Sigma = \{*\}$ is an arbitrary labeling of discrete transitions (or edges);
- the clock set $\mathscr{C} = \{c\}$ contains a single clock;
- the set of edges $E$ is such that $(l_s, g, a, r, l_{s'}) \in E$ iff $l_s \xrightarrow[/R]{} l_{s'}$, $g = \{\underline{\tau}_s \leqslant c \leqslant \bar{\tau}_s\}$, $a = *$, and $r = \{c := 0\}$;
- the invariant map $I(l_s) := \{0 \leqslant c \leqslant \bar{\tau}_s\}, \forall s \in \{1, \ldots, q\}$.

### 10.4.1 Automatic Synthesis of Schedulers

One may think of a scheduler as a coordinating controller that prevents several systems from entering into a conflict configuration. We consider a set of ETC systems that share a common resource, e.g., a computing or communication platform, and propose to use an NTGA-based approach to synthesize schedulers in this set-up. After each update the shared resource is unavailable for some predefined time interval, and to prevent a conflict (a control loop requesting access while the resource is being used) the scheduler decides which control loop shall be updated next by selecting an update mechanism for each control loop (see next paragraph). The process to automatically synthesize schedulers consists of three steps: first, construct an NTGA associated with the set of NCSs; then, define the set of bad states (representing conflicts); and finally, employ a tool like UPPAAL-Tiga [12] to obtain a safe strategy avoiding the bad states. The NTGA derived from the set of NCSs: $\mathscr{G}^{NCSs}$, is a parallel synchronized composition of the TGA associated with the network: $\mathscr{G}^{net}$, and the TGA associated with the control loops $\mathscr{G}^{cli}$ for all $i \in \{1, \ldots, N\}$.

In order to design schedulers, the models of the systems to schedule need to expose variables enabling the control of their dynamics. More precisely, the TSA $\bar{S}_{TSA}$ needs to be enriched to contain more than one *action* in the set $\Sigma$, resulting in the TGA $\mathscr{G}^{cli}$. Several update mechanisms, providing different controllable actions for the scheduler can be considered:

- the update time is based on a triggering mechanism, where a triggering coefficient is selected from the finite set $\{\alpha_1, \ldots, \alpha_p\}$. In TGA $\mathscr{G}^{cli}$ for each $s \in \{1, \ldots, q\}$, we introduce additional locations $l_s^{\alpha_1}$, …, $l_s^{\alpha_p}$ representing the choice of the triggering coefficient $\alpha_1$, …, $\alpha_p$ selected at state $x_{/R} = \mathscr{R}_s$. For each $s \in \{1, \ldots, q\}$, the edges from $l_s$ to $l_s^{\alpha_1}$, …, $l_s^{\alpha_p}$ are controllable enabling the scheduler to choose the triggering coefficient.
- the update time is forced at a predefined time, which is earlier than the minimum inter-sample time of the active triggering condition. In TGA $\mathscr{G}^{cli}$ for each $s \in$

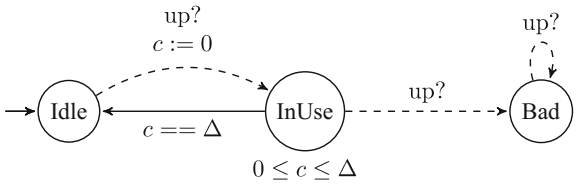


**Fig. 10.3** TGA of a shared resource

$\{1, \ldots, q\}$, we introduce controllable edges originated from $l_s$ that represent earlier controller updates.

- the update time is based on a triggering mechanism but delayed a predefined amount of time to be selected from some set $\{\tau_1^d, \ldots, \tau_r^d\}$. Note that ETC naturally tolerates a maximum amount of delay $\bar{\Delta}$ [20], thus one must select these delays smaller than such $\bar{\Delta}$. In TGA $\mathscr{G}^{cli}$ for each $s \in \{1, \ldots, q\}$, we introduce locations $l_s^{\tau_1^d}, \ldots, l_s^{\tau_r^d}$ that represent the sampled state is in $\mathscr{R}_s$ and the chosen delay is $\tau_1^d, \ldots, \tau_r^d$, respectively. Again, these new edges from $l_s$ to $l_s^{\tau_1^d}, \ldots, l_s^{\tau_r^d}$ are controllable for the scheduler.

Each of these mechanisms can be employed on their own or combined to provide more control handles to the scheduler. The status of the shared resource – available, unavailable and conflict – and the possible transitions among them are modeled by TGA $\mathscr{G}^{net}$ depicted in Fig. 10.3. Thus, the bad states of the NTGA are defined as the set of states such that the location of $\mathscr{G}^{net}$ is *Bad*. For a more detailed treatment of this procedure we refer the reader to the report [19].

## 10.5 Illustrative Examples

In what follows, we illustrate the described abstraction construction on two examples. First, we consider a simple academic two-dimensional LTI system:

$$\begin{aligned} \dot{\xi}_1(t) &= \begin{bmatrix} -14 & 10 \\ -24 & 17 \end{bmatrix} \xi_1(t) + \begin{bmatrix} 1 \\ 2 \end{bmatrix} \nu_1(t), \\ \nu_1(t) &= [9 \quad -6.5]\, \xi_1(t). \end{aligned} \tag{10.24}$$

We employ the following values for the abstraction parameters: the triggering coefficient $\alpha = 0.05$, the upper bound of the inter-sample interval $\bar{\sigma} = 1\,\text{s}$, the order of polynomial approximation $N_{\text{conv}} = 5$, the number of polytopic subdivisions $l = 100$ and the total number of state space partitions $q = 20$.

The resulting abstraction of the closed-loop system (10.24) is provided in Fig. 10.4, depicting $\underline{\tau}_s$ and $\bar{\tau}_s$; $\underline{\tau}_s$ and $\bar{\tau}_s$ in a radial manner, and a representation of the discrete transitions in the resulting TSA. The achieved precision of the abstraction is $\varepsilon = 0.284\,\text{s}$. Figure 10.5 illustrates the validity of the theoretical bounds that we

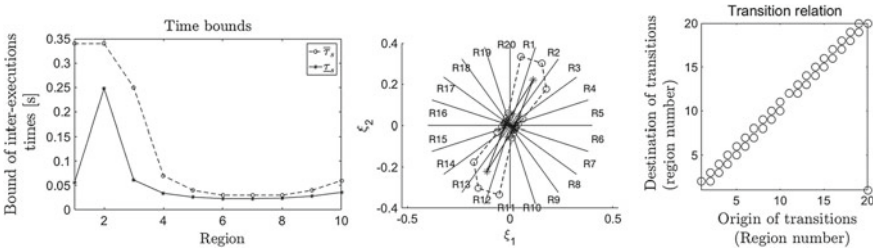


**Fig. 10.4** System (10.24) lower and upper bounds of inter-sample times depicted by solid and dashed curves, respectively (left panel). Spider-web representation of times (note the symmetry) (center panel). Graphic representation of the transition relation (right panel)

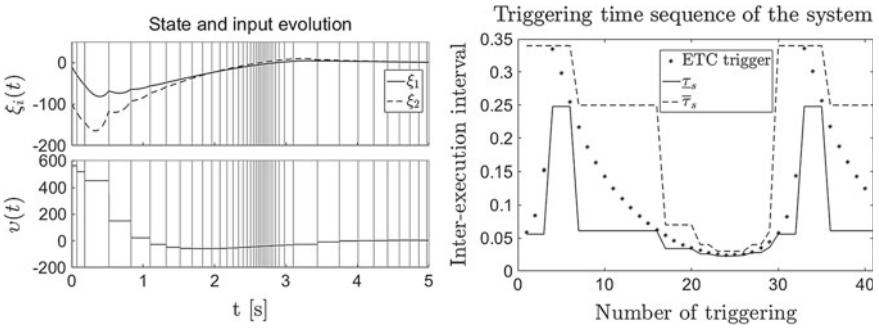


**Fig. 10.5** System (10.24) states and input trajectories (left panel) and triggering times (right panel) of a simulation of the ETC system. Time between triggering (asterisks), predicted lower bound (solid line) and upper bound (dashed line)

found for $\underline{\tau}_s$ (solid line) and $\bar{\tau}_s$ (dashed line). The asterisks represent the inter-sample times sequence during 5 s simulation of the ETC system.

The second example is a somewhat more realistic system: an intelligent vehicle headway controller [24]:

$$\begin{bmatrix} \dot{E}_r(t) \\ \dot{E}_v(t) \\ \dot{a}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -1.43 & -2.149 \end{bmatrix} \begin{bmatrix} E_r(t) \\ E_v(t) \\ a(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0.01077 \end{bmatrix} u(t), \qquad (10.25)$$

$$u(t) = -\begin{bmatrix} 40 & 55.78 & 24.45 \end{bmatrix} [E_r(t) \quad E_v(t) \quad a(t)]^T, \qquad (10.26)$$

where $E_r = R_h - R$, $E_v = V - V_p$, with $R_h$ and $R$ the desired and actual headway, $V_p$ and $V$ the preceding and host vehicle velocities, and $a$ the host velocity acceleration. The controller is implemented with a triggering coefficient $\alpha = 0.05$. In the abstraction we select $m = 10$, the number of subdivisions for each angular coordinate in the interval $[-\frac{\pi}{2}, \frac{\pi}{2})$, which results in $q = 2 \times m^{(n-1)} = 2 \times 10^2 = 200$ states of the abstraction, i.e., regions in which the state space is divided. The rest of the parameters are selected as $\bar{\sigma} = 2$ s, $N_{conv} = 5$, and $l = 100$.

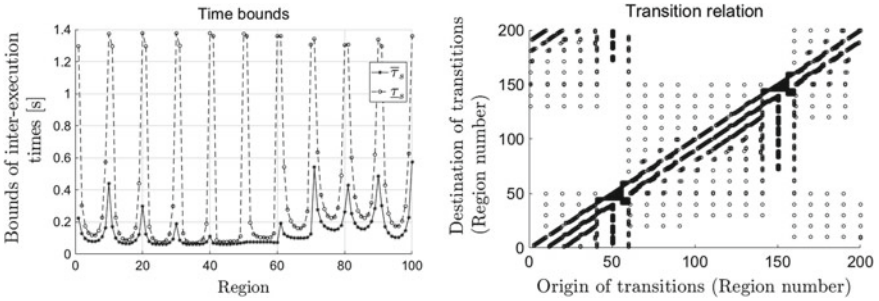


**Fig. 10.6** System (10.25) lower and upper bounds of inter-sample times depicted by solid and dashed curves, respectively (left panel). Graphic representation of the transition relation (right panel)

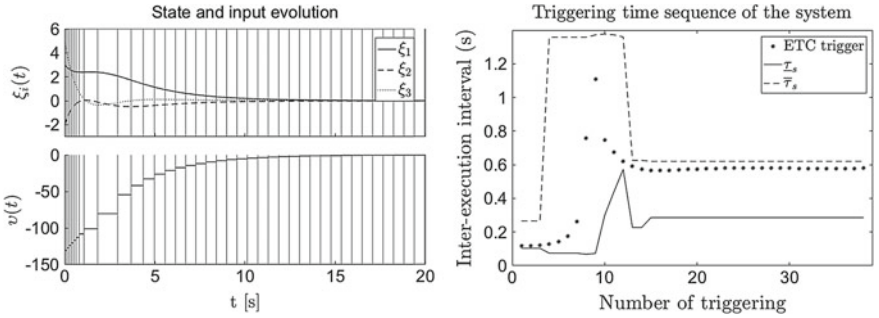


**Fig. 10.7** System (10.25) states and input trajectories (left panel) and triggering times (right panel) of a simulation of the ETC system. Time between triggering (asterisks), predicted lower bound (solid line) and upper bound (dashed line)

Figure 10.6 shows the resulting abstraction of the closed-loop system (10.25). Note that times for only half of the state space (100 regions) are plotted, as the symmetric half of the state space results in identical bounds. The precision of the constructed abstractions is $\varepsilon = 1.3$ s. The validity of the theoretical bounds that we found on a simulation, with initial condition $x_0 = [3 \quad -2 \quad 5]^T$, is visualized in Fig. 10.7.

## 10.6 Conclusion

We have presented a methodology to construct models describing the timing patterns of updates in Event-Triggered control systems. The resulting models can be recast as timed automata, for which a large body of literature and tools are available. In particular, one can employ techniques from the literature on timed automata to automatically synthesize schedulers arbitrating the access to shared resources between ETC loops and possibly other (real-time) tasks.

An apparent drawback of the proposed approach is the amount of computation required to construct the models. In the construction of each abstract state one needs to solve several LMI feasibility problems to construct the output set. Then a reachability analysis must be run for each of these states. Fortunately, this is a procedure that can be easily parallelized and that is only run offline. However, the amount of abstract states that is required (assuming a uniform partitioning) scales exponentially with the dimensionality of the system. A couple of promising approaches to address the challenge of scalability are the use of compositional ideas, as in e.g., [25], and the use of model order reduction techniques, as in e.g., Chap. 1 of this book.

The versatility of timed automata to model the traffic of model-based aperiodic controllers has been also demonstrated in e.g., [26], or Chap. 6 of this book. Extensions to other types of event-based controller implementations, like periodic ETC with dynamic controllers [27], or to the non-linear context [20] can be constructed similarly, provided that: (i) the reachability of the considered systems is possible, and (ii) one can construct computable triggering checks dependent solely on the last sampled state. Many approaches are available for the reachability of non-linear systems, see e.g., [28–30]. The second condition is closely related to the idea of Self-Triggered control, for which large classes of non-linear systems have been studied in e.g., [21, 31]

Future work shall investigate if other applications of these sort of abstractions can be found in the real-time control context. An interesting possibility is the study of security, where timed automata may serve to characterize resilient traffic flows (or conversely attacker patterns) as in the context of Chap. 11 in this book.

## References

1. Araújo, J., Mazo Jr., M., Anta, A., Tabuada, P., Johansson, K.H.: System architectures, protocols and algorithms for aperiodic wireless control systems. IEEE Trans. Ind. Inf. **10**(1), 175–184 (2014)
2. Caccamo, M., Buttazzo, G., Sha, L.: Elastic feedback control. In: Proceedings 12th Euromicro Conference on Real-Time Systems, pp. 121–128. (2000)
3. Bhattacharya, R., Balas, G.: Anytime control algorithm: model reduction approach. J. Guid. Control Dyn. **27**(5), 767–776 (2004)
4. Al-Areqi, S., Gorges, D., Reimann, S., Liu, S.: Event-based control and scheduling codesign of networked embedded control systems, pp. 5299–5304. (2013)
5. Kolarijani, A.S., Mazo Jr., M.: A formal traffic characterization of LTI event-triggered control systems. IEEE Trans. Control Netw. Syst. **5**(1), 274–283. (2018)
6. Abdeddaïm, Y., Asarin, E., Maler, O.: Scheduling with timed automata. Theor. Comput. Sci. **354**(2), 272–300 (2006)
7. Fiter, C., Hetel, L., Perruquetti, W., Richard, J.-P.: A state dependent sampling for linear state feedback. Automatica **48**(8), 1860–1867 (2012)
8. Hetel, L., Daafouz, J., Lung, C.: LMI control design for a class of exponential uncertain systems with application to network controlled switched systems. In: Proceedings of the American Control Conference, pp. 1401–1406. (2007)
9. Chutinan, A., Krogh, B.: Computing polyhedral approximations to flow pipes for dynamic systems. In: Proceedings 37th IEEE Conference on Decision and Control, vol. 2, pp. 2089–2094. (1998)

10. Alur, R., Dill, D.: A theory of timed automata. Theor. Comput. Sci. **126**(2), 183–235 (1994)
11. Maler, O., Pnueli, A., Sifakis, J.: On the synthesis of discrete controllers for timed systems. In: Mayr E., Puech C. (eds.) Proceedings of the 12th Symposium on Theoretical Aspects of Computer Science, vol. 900, pp.229–242 (1995)
12. Uppaal tiga. [Online]. Available: http://people.cs.aau.dk/~adavid/tiga/
13. Tabuada, P.: Verification and Control of Hybrid Systems: A Symbolic Approach. Springer, London, Limited (2009)
14. Ewald, G.: Graduate texts in mathematics. In: Combinatorial Convexity And Algebraic Geometry. Springer, New York (1996)
15. Henzinger, T., Nicollin, X., Sifakis, J., Yovine, S.: Symbolic model checking for real-time systems. Inf. Comput. **111**(2), 193–244 (1994)
16. Bengtsson, J., Yi, W.: Timed automata: semantics, algorithms and tools. Lect. Concurr. Petri Nets **3098**, 87–124 (2004)
17. Behrmann, G., David, A., Larsen, K.: A tutorial on Uppaal. Form. Methods Des. Real-Time Syst. **3185**, 200–236 (2004)
18. Chatain, T., David, A., Larsen, K.G.: Playing games with timed games. IFAC Proc. Vol. **42**(17), 238–243 (2009)
19. Adzkiya, D., Mazo Jr., M.: Scheduling of event-triggered networked control systems using timed game automata. arXiv:1610.03729 (2016)
20. Tabuada, P.: Event-triggered real-time scheduling of stabilizing control tasks. IEEE Trans. Autom. Control **52**(9), 1680–1685 (2007)
21. Anta, A., Tabuada, P.: To sample or not to sample: self-triggered control for nonlinear systems. IEEE Trans. Autom. Control **55**, 2030–2042 (2010)
22. Kvasnica, M., Grieder, P., Baotic, M., Morari, M.: Multi-parametric toolbox (MPT). In: HSCC, pp. 448–462. Springer, Berlin (2004)
23. Bagnara, R., Hill, P.M., Zaffanella, E.: The parma polyhedra library: toward a complete set of numerical abstractions for the analysis and verification of hardware and software systems. Sci. Comput. Program. **72**(1), 3–21 (2008)
24. Bin, L., Rongben, W., Jiangwei, C.: A new optimal controller for intelligent vehicle headway distance. In: IEEE Intelligent Vehicle Symposium, vol. 2, pp. 387–392 (2002)
25. Wang, X., Lemmon, M.D.: Event-triggering in distributed networked control systems. IEEE Trans. Autom. Control **56**(3), 586–601 (2011)
26. Zamani, M., Dey, S., Mohamed, S., Dasgupta, P., Mazo Jr., M.: Scheduling of controllers' update-rates for residual bandwidth utilization. In: 14th International Conference on Formal Modeling and Analysis of Timed Systems, LNCS, vol. 9884, pp. 85–101 (2016)
27. Heemels, W.H., Donkers, M., Teel, A.R.: Periodic event-triggered control for linear systems. IEEE Trans. Autom. Control **58**(4), 847–861 (2013)
28. Balluchi, A., Casagrande, A., Collins, P., Ferrari, A., Villa, T., Sangiovanni-Vincentelli, A.L.: Ariadne: a framework for reachability analysis of hybrid automata. In: Proceedings of the International Syposium on Mathematical Theory of Networks and Systems (2006)
29. Frehse, G., Le Guernic, C., Donzé, A., Cotton, S., Ray, R., Lebeltel, O., Ripado, R., Girard, A., Dang, T., Maler, O.: Spaceex: scalable verification of hybrid systems. Comput. Aided Verif. 379–395 (2011)
30. Althoff, M.: An introduction to CORA 2015. In: ARCH@ CPSWeek, pp. 120–151 (2015)
31. Di Benedetto, M.D., Di Gennaro, S., D'Innocenzo, A.: Digital self-triggered robust control of nonlinear systems. Int. J. Control **86**(9), 1664–1672 (2013)