Efficient Runtime Verification of Real-Time Systems under Parametric Communication Delays

MARTIN FRÄNZLE, Carl von Ossietzky Universitat Oldenburg, Oldenburg, Germany
THOMAS MØLLER GROSEN, Aalborg University, Aalborg, Denmark
KIM GULDSTRAND LARSEN, Aalborg University, Aalborg, Denmark
MARTIN ZIMMERMANN, Aalborg University, Aalborg, Denmark

Timed Büchi automata provide a very expressive formalism for expressing requirements of real-time systems. Online monitoring and active testing of embedded real-time systems can then be achieved by symbolic execution of such automata on the trace observed from the system. However, this direct construction is only faithful if the observation of the trace is immediate in the sense that the monitor (or test harness, respectively) can assign exact timestamps to the actions it observes. This is rarely true in practice due to the substantial and fluctuating parametric delays introduced by the circuitry connecting the observed system to its monitoring or testing device.

We present purely zone-based online monitoring and testing algorithms, which handle such parametric delays exactly without recurrence to costly verification procedures for parametric timed automata. We have implemented our algorithms on top of the real-time model checking tool Uppaal, and report on encouraging initial results.

CCS Concepts: • **Theory of computation** → *Modal and temporal logics*; **Logic and verification**; **Timed and hybrid models;**

Additional Key Words and Phrases: Monitoring, timing uncertainty, timed büchi Automata

ACM Reference Format:
Martin Fränzle, Thomas Møller Grosen, Kim Guldstrand Larsen, and Martin Zimmermann. 2026. Efficient Runtime Verification of Real-Time Systems under Parametric Communication Delays. *Form. Asp. Comput.* 38, 2, Article 17 (June 2026), 32 pages. https://doi.org/10.1145/3749848

1 Introduction

Online monitoring and testing are two important tools to achieve functional correctness of safety-critical systems. They analyse the execution traces observed from a system during its runtime by determining in real-time whether the observed traces satisfy the system's specification. Online monitoring passively observes an (execution) trace of the system. A typical application is to ensure

Associate Editor: Nikolai Kosmatov and Laura Kovács

M. Fränzle has been funded by the State of Lower Saxony, Zukunftslabor Mobilität, and by Deutsche Forschungsgemeinschaft, grants FR 2715/5-1 and FR 2715/6-1. T.M. Grosen and K.G. Larsen have been funded by the VILLUM Investigator grant S4OS, and together with M. Zimmermann they have been supported by DIREC - Digital Research Centre Denmark.
Authors' Contact Information: Martin Fränzle, Carl von Ossietzky Universitat Oldenburg, Oldenburg, Germany; e-mail: martin.fraenzle@informatik.uni-oldenburg.de; Thomas Møller Grosen, Aalborg University, Aalborg, Denmark; e-mail: tmgr@cs.aau.dk; Kim Guldstrand Larsen, Aalborg University, Aalborg, Denmark; e-mail: kgl@cs.aau.dk; Martin Zimmermann, Aalborg University, Aalborg, Denmark; e-mail: mzi@cs.aau.dk.

ACM 1433-299X/2026/06-ART17
https://doi.org/10.1145/3749848

that an **unmanned aerial vehicle** (**UAV**) stays within its safety envelope. In testing, the system is actively subjected to stimuli, which allows to cover a wider range of traces which might not be observed by passively monitoring the system. A typical application is a UAV on a test stand that allows to control wind speed and direction.

Both continuous online monitoring and testing are therefore concerned with unbounded time horizons, unlike offline monitoring where a fixed finite trace is analysed after the execution has terminated. Hence, specifications for online monitoring and testing are typically defined over infinite traces, with the most significant approach being temporal logics. As specifications often include real-time requirements, e.g., "every request is answered within 10 **milliseconds** (**ms**)", we focus here on metric-time temporal logics over timed words. More precisely, we consider **Metric Interval Temporal Logic** (**MITL**) [2], which offers a good balance between expressiveness and algorithmic properties. For example, the request-response specification above is expressed by the MITL formula $G_{\geq 0}(\texttt{req} \rightarrow F_{\leq 10}\texttt{resp})$.

While the specifications classify infinite traces, the traces observed online and to be checked against the specification remain finite. Nevertheless, one can still return verdicts [7]: for example, *every* infinite extension of a finite trace with some request that is not answered within 10 ms violates the request-response specification above. Hence, violation of the specification is already witnessed by such a finite trace. Dually, consider the specification "system calibration is completed within 500 ms", expressed by the formula $F_{\leq 500}\texttt{cc}$ with the proposition cc representing the completion of calibration. Every infinite extension of a finite trace on which the calibration is completed within 400 ms satisfies the specification. Hence, satisfaction of the specification is already witnessed by such a finite trace. However, there are also traces and specifications for which no verdict can currently be drawn, like in the situation where no calibration has been observed yet at current time of 350 ms. As usual, we capture these three situations with the three verdicts $\top$ (satisfaction for every extension), $\bot$ (violation for every extension), and *?* (inconclusive).

Online monitoring and testing can be achieved by compiling the MITL specification into an equivalent **timed Büchi automaton** (**TBA**) and then symbolically executing the automaton on the observed trace of the system [7, 20]. However, this approach is correct only if the actions of the system can be observed immediately by the monitor or test harness, respectively. In practice, there is usually a communication delay between the system and the monitor or testing harness. This delay is induced by various types of circuitry at their interfaces, like technical sensors, conversion between analog and digital signals, and communication networks forwarding signals to the monitor. We follow the approach described in McGraw–Hill's Encyclopedia of Networking and Telecommunications [32] where a communication delay consists of a constant part (latency) and varying part (jitter). Here, we consider a monolithic system that is monitored or tested, i.e., there is a single communication channel from the system to the monitor respectively one from the system to the test harness and one from the test harness to the system.

Due to the delay, the system and the symbolic execution are no longer synchronized but deviate by a delay, for which only bounds, yet not exact values tend to be known. But even then, one can still provide meaningful verdicts, see Figure 1: Consider monitoring of the specification $F_{\leq 10}a \wedge G_{\leq 20}\neg b$, which expresses that an a occurs within 10 ms and no b occurs within 20 ms, under delayed observations. The observed trace shows the first a at 17.3 ms and the first b at 27.1 ms. This observation is only consistent with satisfaction of the constraint $F_{\leq 10}a$ if a's observation delay exceeds 7.3 ms, while satisfaction of $G_{\leq 20}\neg b$ requires a delay of at most 7.1 ms for b. Thus, if the jitter is strictly smaller than 0.2 ms, the specification is definitely violated. Note that the verdict "violated" is true independently of the actual value of the unknown, parametric communication latency.

On the other hand, if the parametric latency is known to be in the range $[4.5, 8]$ ms and the jitter is in $[0, 0.3]$ ms, then we cannot give a definitive verdict: The a may have occurred at 10 ms and

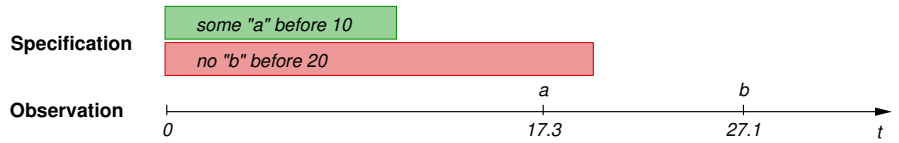

Fig. 1. Monitoring under observation delay: At time $t = 27.1$ we can conclusively decide that the MITL property $F_{[0,10]}a \wedge G_{[0,20]}\neg b$ is violated irrespective of the latency of the observation channel, provided the jitter is less than 0.2.

then has been observed with 7 ms latency plus 0.3 ms jitter at 17.3 = 10 + 7 + 0.3 ms, and the b may have occurred at 20.1 ms and then observed with the same latency (yet independent jitter) at 27.1 = 20.1 + 7 + 0 ms. In this case, the property would be satisfied. But the a may also have occurred at 10.3 ms, violating the property, and still be observed with the same latency at 17.3 = 10.3 + 7 + 0 ms. From the observations, we can nevertheless derive bounds on the parametric latency, as the property definitely is violated irrespective of the actual (unknown) value of the jitter whenever the actual latency is smaller than 7 ms or larger than 7.1 ms. It however cannot be guaranteed to be satisfied when the latency is in the range of $[7, 7.1]$ ms, as satisfaction then depends on the exact value of the jitter, which is not detectable. Thus, one can determine information beyond the verdicts $\top$, $\bot$, and *?* in terms of bounds on the delay that imply definitive verdicts.

1.1 Our Contribution

Based on previous work by Grosen et al. [20] on online monitoring of MITL specifications without delay via TBA, we present a symbolic MITL monitoring algorithm and a symbolic MITL testing algorithm that provide exact verdicts under unknown delay consisting of parametric (i.e., unknown within bounds) latency and jitter. While an unknown delay is a timing parameter, our constructions avoid the semidecidability [3] of analysis for parameterized timed automata, and instead uses only classical clock zones [8].

In addition, our approach has the advantage that it is even more informative than typical monitoring and testing algorithms, which only return a verdict in $\{\top, \bot, ?\}$. Recall the example specification $F_{\leq 10}a \wedge G_{\leq 20}\neg b$ in the case where the jitter is constrained to [0,0.3] ms. As argued above, this specification can, given this bound on the jitter, only be satisfied if $7 \leq \ell \leq 7.1$, where ℓ denotes the actual latency. Our algorithms, for which we also provide a prototype implementation and experimental evaluation, compute such parametric constraints on the set of potential latencies under which the specification can be satisfied as well as on the set of potential latencies under which the specification can be violated.

The implementation is built on top of the real-time model checking tool Uppaal [26] using the **difference-bounded matrix** (**DBM**) data structure allowing for representation of convex polytopes called zones. Most importantly, the DBM data structure can be used for efficient implementation of various geometrical operations over zones needed for the symbolic analysis of timed automata, such as testing for emptiness, inclusion, equality, and computing projection and intersection of zones [8]. Our experiments show encouraging initial results on an industrial gear controller model from [27].

This article is an revised and extended version of an article published at IFM 2024 [18], which contains all proofs omitted in the conference version, a new section on testing under delay, and additional experiments.

1.2 Related Work

Our automata-based monitoring of finite traces against specifications over infinite words using the three verdicts $\{\top, \bot, ?\}$ follows the seminal work of Bauer et al. [7], who presented monitoring

algorithms for the qualitative-time linear temporal logic LTL and its metric-time variant Timed LTL. Their algorithm for Timed LTL is based on clock regions [1], while we follow the approach of Grosen et al. [20] and leverage the performance advantages of clock zones [8], which account for roughly an order of magnitude of improvement in runtime (see, e.g., [25]). Furthermore, Bauer et al. in [7] translated Timed LTL into event-clock automata, which are less expressive than the TBA used both by Grosen et al. [20] and here. More recently, the same approach has been used to monitor real-time properties under assumptions [11].

As our algorithms work with TBA, we also support MITL specifications, as these can be compiled into TBA. The monitoring problem for MITL under perfect observability, i.e. without delay, has been investigated before. Baldor et al. in [5] showed how to construct a monitor for dense-time MITL formulas by constructing a tree of timed transducers. Ho et al. split unbounded and bounded parts of MITL formulas for monitoring, using traditional LTL monitoring for the unbounded parts and permitting a simpler construction for the (finite-word) bounded parts [21]. Bulychev et al. in [10] apply a technique of rewriting a given **Weighted Metric Temporal Logic** (**WMTL**) formula during monitoring as part of performing statistical model checking. None of the above works makes use of the efficient DBM data structure or extends to the setting of TBA that provides the basis of our approach. To this end, we note that as a specification formalism, TBA exceeds the expressive power of MITL, with the additional expressive power clearly being useful in certain application contexts (e.g., in the presence of counting properties).

There is also a large body of work on monitoring with finite-word semantics. Roşu et al. focussed on discrete-time finite-word MTL [34], while Basin et al. proposed algorithms for monitoring real-time finite-word properties [6] and elucidated the differences between different time models. André et al. consider monitoring finite logs of parameterized timed and hybrid systems [38]. Finally, Ulus et al. described algorithms for monitoring finite timed words against specifications given as timed regular expressions. Their monitoring procedures exploit unions of two-dimensional zones [35, 36].

Common to the aforementioned technical developments is the assumption of perfect observability of the system being monitored, where the monitor can rely on always exact measurements of states and/or events of the system as well as of their occurrence times. As this is an unrealistic assumption, the problem of monitoring trace properties under uncertain observation has been addressed before [15, 16, 23, 30, 37], most notably based on **Signal Temporal Logic** (**STL**) and exploiting STL's quantitative semantics [29] that characterizes robustness against variation in state variables. These approaches are mostly orthogonal to ours, as they tend to address uncertainty in the state observed at a time instant rather than uncertainty in the timestamps associated to state observations. It would consequently be interesting to combine the two approaches, thus permitting both state uncertainty due to inexact measurements and time uncertainty due to inexact clocks and fluctuating communication latencies. It should also be noted that robust STL monitoring comes in diverse variants representing different error models and different levels of exactness in representing them. Early and efficient procedures for monitoring under state uncertainty implement the compositional real-valued robustness semantics [15, 30]. This semantics however underapproximates the factual robustness of the verdict against state shifts in the observed trace such that monitoring algorithms based on this compositional semantics are sound and computationally efficient, yet incomplete. Due to the safe approximation, they may yield inconclusive verdicts in actually determined situations. Complete and thus optimally informed STL monitoring under uncertainty, which guarantees a verdict whenever the property is determined, has only recently been investigated. Visconti et al. in [37] developed sound and complete monitoring wrt. an interval model of state measurement error, where each single measurement features an independent displacement ranging over a bounded interval. Finkbeiner et al. in [16, 17] address a refined model distinguishing between a constant,

yet unknown up to bounds, offset and a time-varying, interval-bounded noise, as suggested by the pertinent ISO norm 5725 on measurement accuracy (there called "trueness" and "precision" of a measurement). We here and previously in [18] adopt the latter, more refined model of measurement error and transfer it into the time domain, thus implementing sound and complete monitoring for the case when timestamps are affected by a parametric (unknown, yet constant) observation latency plus a fluctuating jitter that differs between observations. Closest to our approach is [24], which addresses a more confined model of observation delay comprising a fixed known (non-parametric) latency plus a varying jitter. It also covers clock drift, which is an additional source of (relative) jitter that we have excluded to simplify the exposition.

Closely related to (passive) system monitoring is (active) testing, where pre-defined stimuli are supplied to the **system under test** (**SUT**) and the SUT's response is monitored in order to acquire a test verdict. In this journal article, we expand on our previous work [18] to also cover active testing, where the stimulus sequence, its timing, and the expected responses are specified together by TBA. The work thus belongs to the large set of formal approaches to test-case generation and evaluation, which is a long-standing and lively topic with enormous economic impact especially in the hardware domain, but also in the automotive and railway industries [22, 33]. Testing under timing uncertainties has been investigated from the perspective of distributed systems that exhibit asynchronous and complexly interleaved behavior due to variance in component timing [12, 13, 19]. But asynchrony between the test harness and the SUT due to such timing variance in the communication infrastructure connecting the test harness and SUT has not hitherto been covered by formal approaches to testing, despite technical control of their latency and jitter having always been concerns in the design of hardware-in-the-loop testing facilities due to their fundamental impact on the feasibility of test conduct and evaluation [31].

2 Preliminaries

The set of natural numbers (excluding zero) is $\mathbb{N}$, we define $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$, the set of rational numbers is $\mathbb{Q}$, the set of non-negative rational numbers is $\mathbb{Q}_{\geq 0}$. The set of real numbers is $\mathbb{R}$, and the set of non-negative real numbers is $\mathbb{R}_{\geq 0}$. The powerset of a set S is denoted by 2^S.

2.1 Timed Words

A timed word over a finite alphabet Σ is a pair $\rho = (\sigma, \tau)$ where σ is a word over Σ and τ is a sequence of non-decreasing non-negative real numbers of the same length as σ. Timed words may be finite or infinite; in the latter case, we require $\limsup \tau = \infty$, i.e., time diverges. The set of finite timed words is denoted by $T\Sigma^*$ and the set of infinite timed words by $T\Sigma^\omega$. We also represent a timed word as a sequence of pairs $(\sigma_1, \tau_1), (\sigma_2, \tau_2), \ldots$. If $\rho = (\sigma_1, \tau_1), (\sigma_2, \tau_2), \ldots, (\sigma_n, \tau_n)$ is a finite timed word, we denote by $\tau(\rho)$ the total time duration of ρ, i.e., τ_n (with the convention that the duration of the empty word is 0).

If $\rho_1 = (\sigma_1^1, \tau_1^1), \ldots, (\sigma_n^1, \tau_n^1)$ is a finite timed word, $\rho_2 = (\sigma_1^2, \tau_1^2), (\sigma_2^2, \tau_2^2), \ldots$ is a finite or infinite timed word, and $t \in \mathbb{Q}_{\geq 0}$ then the timed word concatenation $\rho_1 \cdot_t \rho_2$ is defined if and only if $\tau(\rho_1) \leq t$. Then, $\rho_1 \cdot_t \rho_2 = (\sigma_1, \tau_1), (\sigma_2, \tau_2), \ldots$ such that

$$\sigma_i = \begin{cases} \sigma_i^1 & \text{if and only if } i \leq n \\ \sigma_{i-n}^2 & \text{else} \end{cases} \quad \text{and} \quad \tau_i = \begin{cases} \tau_i^1 & \text{if and only if } i \leq n \\ \tau_{i-n}^2 + t & \text{else.} \end{cases}$$

In the following, we often need to "shift" a timed word in the sense that we add or subtract a $t \in \mathbb{R}_{\geq 0}$ to each time point of ρ. In the latter case, we need to be careful to ensure that the time points in the shifted word are still nonnegative. For the sake of readability, let us introduce some notation for these operations. Given a (finite or infinite) timed word $\rho = (\sigma_1, \tau_1), (\sigma_2, \tau_2), \ldots$ and such a $t \in \mathbb{R}_{\geq 0}$,

— let $\rho + t$ denote the timed word $(\sigma_1, \tau_1 + t), (\sigma_2, \tau_2 + t), \ldots$, and
— if $t \leq \tau_1$, let $\rho - t$ denote the timed word $(\sigma_1, \tau_1 - t), (\sigma_2, \tau_2 - t), \ldots$. This is well-defined, as we require that τ_1 (and therefore each τ_i) is at least t, so we never obtain negative time points in $\rho - t$.

The following properties follow directly from the definition of timed concatenation and will be applied in the proofs below.

Remark 2.1. Let $\rho = (\sigma_1, \tau_1), \ldots, (\sigma_n, \tau_n) \in T\Sigma^*$, let $\mu = (\sigma'_1, \tau'_1), (\sigma'_2, \tau'_2), \ldots \in T\Sigma^\omega$, and let $\tau(\rho) \leq t$.

(1) Let $t' \in \mathbb{R}_{\geq 0}$ be such that $t - t' \geq \tau(\rho)$. Then
$$\rho \cdot_t \mu = \rho \cdot_{t-t'} (\mu + t').$$
(2) Let $0 \leq t' \leq t$, let $n' \in \{0, 1, \ldots, n\}$ be such that $\tau_{n'} \leq t' \leq \tau_{n'+1}$ (where we use $\tau_0 = -\infty$ to allow $n' = 0$ and $\tau_{n+1} = \infty$ to allow $n' = n$) and define $\rho_1 = (\sigma_1, \tau_1), \ldots, (\sigma_{n'}, \tau_{n'})$ as well as $\rho_2 = (\sigma_{n'+1}, \tau_{n'+1}), \ldots, (\sigma_n, \tau_n)$. Then
$$\rho \cdot_t \mu = \rho_1 \cdot_{t'} ((\rho_2 - t') \cdot_{t-t'} \mu).$$
(3) Let $t' \geq 0$ be such that $\tau(\rho) \leq t - t'$, let $n' \geq 0$ be such that $\tau'_{n'} \leq t' \leq \tau'_{n'+1}$ (where we use $\tau_0 = -\infty$ to allow $n' = 0$), and define $\rho' = (\sigma'_1, \tau'_1), \ldots, (\sigma'_{n'}, \tau'_{n'})$ as well as $\mu' = (\sigma'_{n'+1}, \tau'_{n'+1}), (\sigma'_{n'+2}, \tau'_{n'+2}), \ldots$. Then
$$\rho \cdot_{t-t'} \mu = (\rho \cdot_{t-t'} \rho') \cdot_t (\mu' - t').$$
Note that n' is well-defined due to time-divergence of μ.

2.2 Timed Automata

A Timed Büchi automaton (TBA) $\mathcal{A} = (Q, Q_0, \Sigma, C, \Delta, \mathcal{F})$ consists of a finite alphabet Σ, a finite set Q of locations, a set $Q_0 \subseteq Q$ of initial locations, a finite set C of clocks, a finite set $\Delta \subseteq Q \times Q \times \Sigma \times 2^C \times G(C)$ of transitions with $G(C)$ being the set of clock constraints over C, and a set $\mathcal{F} \subseteq Q$ of accepting locations. A transition (q, q', a, λ, g) is an edge from q to q' on input symbol a, where λ is the set of clocks to reset and g is a clock constraint over C. A clock constraint is a conjunction of atomic constraints of the form $x \sim n$, where x is a clock, $n \in \mathbb{N}_0$, and $\sim \in \{<, \leq, =, \geq, >\}$. A state of $\mathcal{A}$ is a pair (q, v) where q is a location in Q and $v \colon C \to \mathbb{R}_{\geq 0}$ is a valuation mapping clocks to their values. For any $d \in \mathbb{R}_{\geq 0}$, $v + d$ is the valuation $x \mapsto v(x) + d$.

A run of $\mathcal{A}$ from a state (q_0, v_0) over a timed word $(\sigma_1, \tau_1)(\sigma_2, \tau_2) \cdots$ is a sequence of steps $(q_0, v_0) \xrightarrow{(\sigma_1, \tau_1)} (q_1, v_1) \xrightarrow{(\sigma_2, \tau_2)} (q_2, v_2) \xrightarrow{(\sigma_3, \tau_3)} \cdots$ where for all $i \geq 1$ there is a transition $(q_{i-1}, q_i, \sigma_i, \lambda_i, g_i)$ such that $v_i(x) = 0$ for all x in λ_i and $v_i(x) = v_{i-1}(x) + (\tau_i - \tau_{i-1})$ otherwise, and g_i is satisfied by the valuation $v_{i-1} + (\tau_i - \tau_{i-1})$. Here, we use $\tau_0 = 0$. Given a run r, we denote the set of locations visited infinitely many times by r as $\mathrm{Inf}(r)$. A run r of $\mathcal{A}$ is accepting if $\mathrm{Inf}(r) \cap \mathcal{F} \neq \emptyset$. The language of $\mathcal{A}$ from a starting state (q, v), denoted $L(\mathcal{A}, (q, v))$, is the set of all infinite timed words with an accepting run in $\mathcal{A}$ starting from (q, v). We define the language of $\mathcal{A}$, written $L(\mathcal{A})$, to be $\bigcup_q L(\mathcal{A}, (q, v_0))$, where q ranges over Q_0 and where $v_0(x) = 0$ for all $x \in C$.

2.3 Logic

We use Metric Interval Temporal Logic (MITL) to express properties to be monitored; these are subsequently translated into equivalent TBA which we use in our monitoring algorithm. The syntax of MITL formulas over a finite alphabet Σ is defined as
$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid X_I \varphi \mid \varphi \, U_I \varphi$$

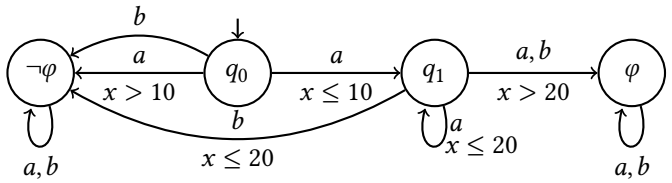

Fig. 2. An automaton for the language of the property $\varphi = F_{[0,10]}a \wedge G_{[0,20]}\neg b$ and its negation: If location φ is accepting then it accepts $L(\varphi)$, if location $\neg\varphi$ is accepting then it accepts $L(\neg\varphi)$.

where $p \in \Sigma$ and I ranges over non-singular intervals over $\mathbb{R}_{\geq 0}$ with endpoints in $\mathbb{N}_0 \cup \{\infty\}$. We write $\sim n$ for $I = \{d \in \mathbb{R}_{\geq 0} \mid d \sim n\}$ for $\sim \in \{<, \leq, \geq, >\}$ and $n \in \mathbb{N}$. We also define the standard syntactic sugar: $\mathtt{true} = p \vee \neg p$, $\varphi \wedge \psi = \neg(\neg\varphi \vee \neg\psi)$, $F_I\varphi = \mathtt{true}\, U_I\varphi$, and $G_I\varphi = \neg F_I \neg\varphi$.

The satisfaction relation $\rho, i \vDash \varphi$ is defined for infinite timed words $\rho = (\sigma_1, \tau_1), (\sigma_2, \tau_2), \ldots$, positions $i \geq 1$, and an MITL formula φ:

- $\rho, i \vDash p$ if $p = \sigma_i$.
- $\rho, i \vDash \neg\varphi$ if $\rho, i \nvDash \varphi$.
- $\rho, i \vDash \varphi \vee \psi$ if $\rho, i \vDash \varphi$ or $\rho, i \vDash \psi$.
- $\rho, i \vDash X_I\varphi$ if $\rho, (i+1) \vDash \varphi$ and $\tau_{i+1} - \tau_i \in I$.
- $\rho, i \vDash \varphi\, U_I \psi$ if there exists $k \geq i$ s.t. $\rho, k \vDash \psi$, $\tau_k - \tau_i \in I$, and $\rho, j \vDash \varphi$ for all $i \leq j < k$.

We write $\rho \vDash \varphi$ whenever $\rho, 1 \vDash \varphi$, and say that ρ satisfies φ. The language $L(\varphi)$ of an MITL formula φ is the set of all $\rho \in T\Sigma^\omega$ such that $\rho \vDash \varphi$.

Theorem 2.2 ([2, 9]). *For each MITL formula φ there exists a TBA $\mathcal{A}$ with $L(\varphi) = L(\mathcal{A})$.*

Figure 2 illustrates Theorem 2.2 by providing TBA for the formula $F_{[0,10]}a \wedge G_{[0,20]}\neg b$ from the introduction and its negation.

3 Monitoring under Delayed Observation

According to McGraw–Hill's Encyclopedia of Networking and Telecommunications [32], a communication delay consists of a constant part (latency) and varying part (jitter). We describe the delay as a pair $(\delta, \varepsilon) \in \mathbb{R}^2_{\geq 0}$ where δ is the constant latency for all signals and ε is the bound on the jitter. Thus, all signals from the system are delayed within $[\delta, \delta + \varepsilon]$ before they arrive at the monitor.

In the simplest case, our obligation is to monitor violation of an MITL specification φ by a system while observing the events through a channel *Chan* featuring a constant, yet unknown (up to a given, but maybe trivial, lower bound $l \in \mathbb{R}_{\geq 0}$ and upper bound $u \in \mathbb{R}_{\geq 0}$) transportation latency $\delta \in [\ell, u]$ and a varying jitter bounded by $\varepsilon \in \mathbb{R}_{\geq 0}$. Figure 1 shows an example of a property and an observation that conclusively violates the specification at time 27.1, even if the channel latency $\delta \in [0, \infty[$ is unknown, as long as the jitter is bounded by 0.2.

Thus, we need to distinguish between observations (the timed word corresponding to the events as they are observed by the monitoring device, subject to delay) and the possible ground-truths, as they may have been emitted by the monitored system. We begin by formalizing the concept of observation, where the occurrence of observed events is constrained by a set $\mathcal{D}$ capturing known bounds on the delay. Obviously, under latency δ (and arbitrary jitter), the first observation can only be made after at least δ units of time.

Definition 3.1. A delay set $\mathcal{D}$ is a nonempty subset of $\mathbb{R}^2_{\geq 0}$ containing pairs of latencies and jitters. A $\mathcal{D}$-observation, i.e. an observation that can in principle be made under delay in $\mathcal{D}$, is a finite timed word $\rho^* = (\sigma_1^*, \tau_1^*), \ldots, (\sigma_m^*, \tau_m^*)$ with $\tau_1^* \geq \delta$ for some $(\delta, \varepsilon) \in \mathcal{D}$.

As the ground-truth occurrence times of events in the system cannot be determined exactly from their delayed copies that the monitor receives through the communication channel, we have to consider all ground-truth timed words that the particular observation is consistent with, as follows.[1]

Definition 3.2 (Consistency). Let $\rho^* = (\sigma_1^*, \tau_1^*), \dots, (\sigma_m^*, \tau_m^*)$ be a $\{(\delta, \varepsilon)\}$-observation and let $\rho = (\sigma_1, \tau_1), \dots, (\sigma_n, \tau_n)$ be a finite timed word. We say that ρ is *consistent* with ρ^* at observation time $t \in \mathbb{R}_{\geq 0}$ under latency δ and jitter ε if and only if

(1) $\tau_n \leq t$ and $\tau_m^* \leq t$,
(2) $n \geq m$, and $\sigma_i = \sigma_i^*$ and $\tau_i^* - \tau_i \in [\delta, \delta + \varepsilon]$ for all $i \in \{1, \dots, m\}$,[2] and
(3) if $n > m$ then $\tau_{m+1} \geq t - (\delta + \varepsilon)$.

We denote the set of timed words ρ that are consistent with a $\{(\delta, \varepsilon)\}$-observation ρ^* at observation time t under latency δ and jitter ε by $GT_{\delta,\varepsilon}(\rho^*, t)$. Then, we define $GT_{\mathcal{D}}(\rho^*, t) = \bigcup_{(\delta,\varepsilon)\in\mathcal{D}} GT_{\delta,\varepsilon}(\rho^*, t)$.

$GT_{\mathcal{D}}(\rho^*, t)$ thus collects the possible ground-truths that are consistent with the observation ρ^* when the time elapsed since the system has started is t, and the delay (δ, ε) is within the set $\mathcal{D}$.

Example 3.3. Figure 3 shows a $\{(\delta, \varepsilon)\}$-observation and a consistent ground-truth and illustrates how the delay shifts the timestamps of the events. The length of ρ is $n = 9$ and the length of ρ^* is $m = 4$. Recall that t is the time of observation.

In particular, notice the following:

- No event can occur in the observation ρ^* with a timestamp smaller than δ, as it takes at least δ units of time for an event to be send from the system through the communication channel to the monitor. Obviously, at the system side (i.e., in the ground-truth ρ) events can happen at any timestamp, also before δ (e.g., the first a).
- The difference $\tau_i^* - \tau_i$ for $i \leq 4$ (i.e., the difference between the time the event is observed and the time the event was emitted) must be in the interval $[\delta, \delta + \varepsilon]$.
- The time elapsed between the b and the last a in the observation ρ^* is larger than the time elapsed between the corresponding events in the ground-truth ρ. This means the jitter for the a is larger than the jitter for the b.
- The last five events in ρ have not yet been observed in ρ^*. Such events can only have timestamps in the interval $[t - (\delta + \varepsilon), t]$, as all earlier events must necessarily have been observed. Said differently, there cannot be any events between timestamp τ_4 (corresponding to the last observed event in ρ^* with timestamp τ_4^*) and timestamp $t - (\delta + \varepsilon)$, as any such event would have arrived at the monitor, even under the maximal possible delay of $\delta + \varepsilon$. However, there can be an arbitrary number of events in the ground-truth ρ between timestamps $t - (\delta + \varepsilon)$ and t.

Remark 3.4. Note that $GT_{\mathcal{D}}(\rho^*, t)$ is always nonempty, if ρ^* is a $\mathcal{D}$-observation and $t \geq \tau(\rho^*)$, as it contains, e.g., $\rho^* - \delta$ for a $(\delta, \varepsilon) \in \mathcal{D}$ such that ρ^* is a (δ, ε)-observation.

The following property about consistent words will be useful in the proofs below. Here, we use an extension relation over observations with time points: Let $\rho = (\sigma_1, \tau_1), \dots, (\sigma_n, \tau_n)$ and $\rho' = (\sigma_1', \tau_1'), \dots, (\sigma_{n'}', \tau_{n'}')$ be two finite timed words. Also let $t, t' \in \mathbb{R}_{\geq 0}$ with $\tau(\rho) \leq t$ and $\tau(\rho') \leq t'$. Then, we define $(\rho, t) \sqsubseteq (\rho', t')$, if $n \leq n'$, $\sigma_i = \sigma_i'$ and $\tau_i = \tau_i'$ for all $i \leq n$, and either $n = n'$ and $t \leq t'$ or $n < n'$ and $t \leq \tau_{n+1}'$.

[1] Note that we simplify our definitions by assuming that jitter does not change the order of observations. Under the additional assumption that only a (uniformly) bounded number of events can be generated by the system in each unit of time, it is possible to take "overtaking" of events into account, by looking at all consistent permutations. However, this would lead to a severe overhead in the implementation.

[2] Note that the conference version [18] contained a typo in this condition.

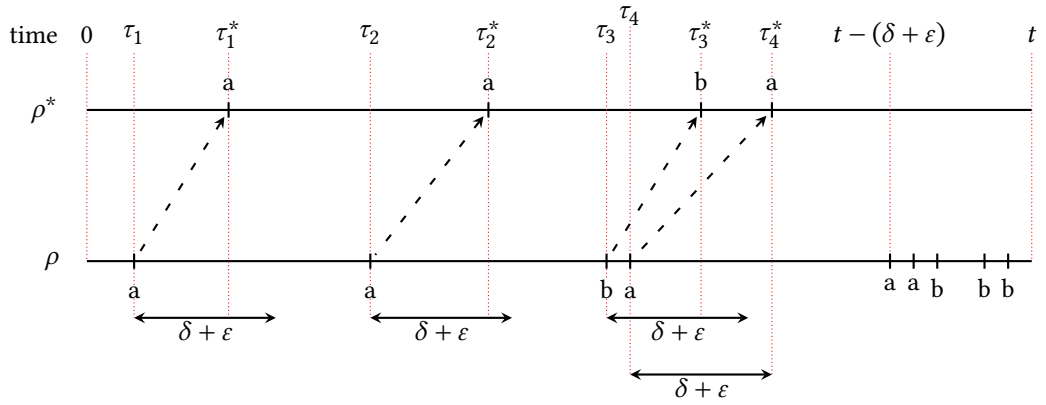

Fig. 3. A $\{(\delta,\varepsilon)\}$-observation ρ^* and a consistent ground-truth ρ.

Lemma 3.5. *Let $(\rho_1^*,t_1) \sqsubseteq (\rho_2^*,t_2)$ with $\tau(\rho_1^*) \le t_1$ and $\tau(\rho_2^*) \le t_2$, let $\rho_2 = (\sigma_1,\tau_1),\dots,(\sigma_n,\tau_n) \in GT_{\delta,\varepsilon}(\rho_2^*,t_2)$, let $n' \in \{0,1,\dots,n\}$ be such that $\tau_{n'} \le t_1 \le \tau_{n'+1}$ (where we use $\tau_0 = -\infty$ to allow $n'=0$ and $\tau_{n+1} = \infty$ to allow $n'=n$), and define $\rho_1 = (\sigma_1,\tau_1),\dots,(\sigma_{n'},\tau_{n'})$. Then $\rho_1 \in GT_{\delta,\varepsilon}(\rho_1^*,t_1)$.*

Proof. We need to show that ρ_1 is consistent with ρ_1^* at t_1 under δ and ε. The first requirement of the definition of consistency follows from $\tau(\rho_1^*) \le t_1$ and the choice of n' (which implies $\tau(\rho_1) \le t_1$). The second requirement follows from the fact that ρ_2 is consistent with ρ_2^* at t_2 under δ and ε and the fact that ρ_1 is a prefix of ρ_2 and ρ_1^* is a prefix of ρ_2^*.

Finally, consider the third requirement and assume it is violated, i.e., let ρ_1^* have m letters and assume ρ_1 has at least $m+1$ letters, i.e., the $(m+1)$-th letter of ρ_1 has not yet been observed before time t_1 in (ρ_1^*). Let τ_{m+1} denote the time point of the $(m+1)$-th letter of ρ_1, which is also the time-point of the $(m+1)$-th letter of ρ_2. We consider two cases.

If ρ_1^* and ρ_2^* have the same length, then the $(m+1)$-th letter of ρ_1 (which is also the $(m+1)$-th letter of ρ_2) has also not been observed before time t_2 in (ρ_2^*). Hence, the third requirement of consistency (between ρ_2^* and ρ_2) implies $t_{m+1} \ge t_2 - (\delta+\varepsilon)$. Hence, we obtain the desired bound $t_{m+1} \ge t_1 - (\delta+\varepsilon)$ from the inequality $t_2 \ge t_1$.

The only other option is that ρ_1^* is strictly shorter than ρ_2^*. This implies that the $(m+1)$-th letter of ρ_2 (which is also the $(m+1)$-th letter of ρ_1) has been observed before time t_2 in ρ_2^*. Then, the second requirement of consistency (between ρ_2^* and ρ_2) implies $\tau_{m+1}^* - \tau_{m+1} \in [\delta,\delta+\varepsilon]$, where τ_{m+1}^* is the time-point at which the $(m+1)$-th letter of ρ_2 is observed in ρ_2^*. Furthermore, the definition of $\sqsubseteq$ yields $\tau_{m+1}^* \ge t_1$. Combining these two yields the desired bound $t_{m+1} \ge t_1 - (\delta+\varepsilon)$. □

Next, we introduce monitoring under delay. A monitor obviously ought to supply a verdict if and only if that verdict applies across *all possible* ground-truth timed words that the observed word explains. For our definition of monitor, we use the set $\mathbb{B}_3 = \{\top, ?, \bot\}$ of verdicts, as usual.

Definition 3.6 (Monitor Verdicts under Delay). Given a language $L \subseteq T\Sigma^\omega$, a set of possible observation delays $\mathcal{D}$, a $\mathcal{D}$-observation $\rho^* \in T\Sigma^*$, and an observation time $t \ge \tau(\rho^*)$, the function $\mathcal{V}_\mathcal{D} : 2^{T\Sigma^\omega} \to T\Sigma^* \times \mathbb{R}_{\ge 0} \to \mathbb{B}_3$ evaluates to the verdict

$$\mathcal{V}_\mathcal{D}(L)(\rho^*,t) = \begin{cases} \top & \text{if } \rho \cdot_t \mu \in L \text{ for all } \rho \in GT_\mathcal{D}(\rho^*,t) \text{ and all } \mu \in T\Sigma^\omega, \\ \bot & \text{if } \rho \cdot_t \mu \notin L \text{ for all } \rho \in GT_\mathcal{D}(\rho^*,t) \text{ and all } \mu \in T\Sigma^\omega, \\ ? & \text{otherwise.} \end{cases}$$

$\mathcal{V}_\mathcal{D}(L)(\rho^*,t)$ is undefined when $t < \tau(\rho^*)$.

Example 3.7. Consider the property $\varphi = F_{[0,10]}a \wedge G_{[0,20]}\neg b$ and observed word $\rho^* = (a, 17.3), (b, 27.1)$ shown in Figure 1, time point $t = 27.1$, and set of delays $\mathcal{D} = \{(\delta, 0.2) \mid \delta \in \mathbb{R}_{\geq 0}\}$. As the jitter is bounded by 0.2, in all ground-truths either a occurred after time point 10, or b occurred before time point 20. Thus, all extensions of all possible ground-truths satisfy $\neg\varphi$, i.e., $\mathcal{V}_{\mathcal{D}}(L(\varphi)))(\rho^*, t) = \bot$.

Note that for the special case of $\mathcal{D} = \{(0, 0)\}$ we cover classical (i.e., delay-free) monitoring [20]. Before we turn our attention to computing $\mathcal{V}$ in Sections 4 and 5, we study some properties of our definition. First, let us note that the ability to make firm verdicts increases with increased certainty of the observation channel delay.

Lemma 3.8. *Let $L \subseteq T\Sigma^\omega$, $\rho^* \in T\Sigma^*$, let $\mathcal{D} \subseteq \mathcal{D}'$ be delay sets, let ρ^* be a $\mathcal{D}$-observation, and let $t \geq \tau(\rho^*)$. Then, $\mathcal{V}_{\mathcal{D}'}(L)(\rho^*, t) = \top$ implies $\mathcal{V}_{\mathcal{D}}(L)(\rho^*, t) = \top$ and $\mathcal{V}_{\mathcal{D}'}(L)(\rho^*, t) = \bot$ implies $\mathcal{V}_{\mathcal{D}}(L)(\rho^*, t) = \bot$.*

Proof. Note that $\mathcal{D} \subseteq \mathcal{D}'$ implies $GT_{\mathcal{D}}(\rho^*, t) \subseteq GT_{\mathcal{D}'}(\rho^*, t)$. Thus, the universal quantification over possible ground-truths ρ in the first two cases of the definition of $\mathcal{V}_{\mathcal{D}}(L)(\rho^*, t)$ ranges over a subset of the possible ground-truths that are considered for $\mathcal{V}_{\mathcal{D}'}(L)(\rho^*, t)$. □

As a refinement of the verdict function in Definition 3.6, one may provide information about the delay parameters (δ, ε) that can explain an observation. Given $L \subseteq T\Sigma^\omega$, a finite timed word $\rho^* \in T\Sigma^*$, and $t \geq \tau(\rho^*)$, the set $\Delta(L, \rho^*, t)$ of delays that are consistent with the observation ρ^* at t is defined as

$$\Delta(L, \rho^*, t) = \{(\delta, \varepsilon) \mid \exists \rho \in GT_{\delta,\varepsilon}(\rho^*, t)\, \exists \mu \in T\Sigma^\omega \text{ s.t. } \rho \cdot_t \mu \in L\}.$$

We denote by $\Delta_{\mathcal{D}}(L, \rho^*, t)$ the set $\Delta(L, \rho^*, t) \cap \mathcal{D}$.

The following fact follows directly from the nonemptiness of ground-truths and is later useful in proofs.

Lemma 3.9. *Let ρ^* be a (δ, ε)-observation, $t \geq \tau(\rho^*)$, and $L \subseteq T\Sigma^\omega$. Then, $(\delta, \varepsilon) \in \Delta(L, \rho^*, t)$ or $(\delta, \varepsilon) \in \Delta(\overline{L}, \rho^*, t)$ (note that it may be in both).*

Proof. Fix some $\rho \in GT_{\delta,\varepsilon}(\rho^*, t)$ (which is always possible due to Remark 3.4) and some $\mu \in T\Sigma^\omega$. Then, $\rho \cdot_t \mu$ is either in L or in $\overline{L}$. In the former case, we have $(\delta, \varepsilon) \in \Delta(L, \rho^*, t)$, in the latter case, we have $(\delta, \varepsilon) \in \Delta(\overline{L}, \rho^*, t)$. □

In the following, we present an example of a delay that is in both sets of consistent delays.

Example 3.10. Let $L = L(F_{\leq 10}a)$, consider the observation $\rho^* = (b, 3)$, and let $\mathcal{D} = \{(3, 7)\}$. Then, we have $(3, 7) \in \Delta(L, \rho^*, t)$ (as we can extend the ground-truth $(b, 0)$ so that it is in L) and $(3, 7) \in \Delta(\overline{L}, \rho^*, t)$ (as we can extend the ground-truth $(b, 0)$ so that it is in $\overline{L}$).

Our next result shows that conclusive monitoring verdicts can be characterized via the sets of consistent delays, i.e., computing the sets of consistent delays generalizes monitoring under delay.

Lemma 3.11. *Given $L \subseteq T\Sigma^\omega$, a set $\mathcal{D}$ of delays, a $\mathcal{D}$-observation $\rho^* \in T\Sigma^*$, and $t \geq \tau(\rho^*)$, we have*

(1) *$\Delta_{\mathcal{D}}(L, \rho^*, t) = \emptyset$ if and only if $\mathcal{V}_{\mathcal{D}}(L)(\rho^*, t) = \bot$, and*
(2) *$\Delta_{\mathcal{D}}(\overline{L}, \rho^*, t) = \emptyset$ if and only if $\mathcal{V}_{\mathcal{D}}(L)(\rho^*, t) = \top$.*

Proof. We have

$$\begin{aligned}
&\Delta_{\mathcal{D}}(L, \rho^*, t) = \emptyset \\
\Leftrightarrow\ &\rho \cdot_t \mu \in \overline{L} \text{ for all } (\delta, \varepsilon) \in \mathcal{D}, \text{ all } \rho \in GT_{\delta,\varepsilon}(\rho^*, t), \text{ and all } \mu \in T\Sigma^\omega \\
\Leftrightarrow\ &\rho \cdot_t \mu \in \overline{L} \text{ for all } \rho \in GT_{\mathcal{D}}(\rho^*, t) \text{ and all } \mu \in T\Sigma^\omega \\
\Leftrightarrow\ &\mathcal{V}_{\mathcal{D}}(L)(\rho^*, t) = \bot.
\end{aligned}$$

The second claim is obtained by a dual argument (swapping $\bot$ with $\top$ and L with $\overline{L}$). □

But even in the case when both delay-sets are nonempty (i.e., the verdict is *?*), we can still provide useful information in terms of the sets $\Delta(L, \rho^*, t)$ and $\Delta(\overline{L}, \rho^*, t)$ of consistent delays. In particular, the set of consistent delays is non-increasing during observations: By extending the observations, we (potentially) reduce the set of consistent delays.

Lemma 3.12. *Let $(\rho_1^*, t_1) \sqsubseteq (\rho_2^*, t_2)$ for finite timed words ρ_1^* and ρ_2^* with $t_1 \geq \tau(\rho_1^*)$ and $t_2 \geq \tau(\rho_2^*)$ and $t_2 \geq t_1$. Then, $\Delta(L, \rho_1^*, t_1) \supseteq \Delta(L, \rho_2^*, t_2)$.*

Proof. Let $(\delta, \varepsilon) \in \Delta(L, \rho_2^*, t_2)$, i.e., there exists $\rho_2 \in GT_{\delta,\varepsilon}(\rho_2^*, t_2)$ and a $\mu_2 \in T\Sigma^\omega$ such that $\rho_2 \cdot_{t_2} \mu_2 \in L$.

Let $\rho_2 = (\sigma_1, \tau_1), \dots, (\sigma_n, \tau_n)$ and let n' be maximal with $\tau_{n'} \leq t_1$ (where we use $\tau_0 = -\infty$ to allow $n' = 0$ and $\tau_{n+1} = \infty$ to allow $n' = n$). Then, Lemma 3.5 yields

$$\rho_1 = (\sigma_1, \tau_1), \dots, (\sigma_{n'}, \tau_{n'}) \in GT_{\delta,\varepsilon}(\rho_1^*, t_1)$$

and an application of Remark 2.1. Remark 2 yields

$$\rho_1 \cdot_{t_1} \Big(\big[((\sigma_{n'+1}, \tau_{n'+1}), \dots, (\sigma_n, \tau_n)) - t_1 \big] \cdot_{t_2 - t_1} \mu \Big) = \rho_2 \cdot_{t_2} \mu_2 \in L.$$

This implies $(\delta, \varepsilon) \in \Delta(L, \rho_1^*, t_1)$. □

Another interesting point is that in some cases, no extension of the observed word will provide a definitive verdict.

Example 3.13. Consider the language $L(F_{\leq 10} a)$, the observation $\rho^* = (a, 15)$, and the set $\mathcal{D} = \{(\delta, 0) \mid \delta \in [0, 10]\}$ of delays. For any given $t \geq \tau(\rho^*)$ the sets of consistent delays are $\Delta_{\mathcal{D}}(L, \rho^*, t) = \{(\delta, 0) \mid \delta \in [5, 10]\}$ and $\Delta_{\mathcal{D}}(\overline{L}, \rho^*, t) = \{(\delta, 0) \mid \delta \in [0, 5)\}$, i.e., both sets of consistent delays are a strict subset of $\mathcal{D}$. Further, due to Lemma 3.12, this will be the case, no matter what observations occur in the future, as the set of consistent delays can only shrink when further observations are made.

As the sets of consistent verdicts can only shrink, but must contain every possible $(\delta, 0) \in \mathcal{D}$ (due to the fact that ρ^* is a $(\delta, 0)$-observation for each such $(\delta, 0)$ and due to Lemma 3.9), we can conclude that neither of the sets can become empty. So, Lemma 3.11 implies that the verdict is *?*, even if additional observations occur.

The following lemma formalizes this: as soon as the set of consistent delays w.r.t. L ($\overline{L}$) is no longer equal to $\mathcal{D}$, then the verdict can never become $\top$ ($\bot$).

Lemma 3.14. *Let $L \subseteq T\Sigma^\omega$, $\mathcal{D}$ be a set of delays, and $\rho^* = (\sigma_1^*, \tau_1^*), \dots, (\sigma_m^*, \tau_m^*)$ a nonempty $\mathcal{D}$-observation. Then, for all $t \geq \tau(\rho^*)$*

1. *$\Delta_{\mathcal{D}}(L, \rho^*, t) \subsetneq \mathcal{D} \cap \{(\delta, \varepsilon) \mid \delta \leq \tau_1^*\}$ implies there is no $\rho_1^* \in T\Sigma^*$ such that $\mathcal{V}_{\mathcal{D}}(L)(\rho^* \cdot_t \rho_1^*, t') = \top$ for any $t' \geq t + \tau(\rho_1^*)$, and*
2. *$\Delta_{\mathcal{D}}(\overline{L}, \rho^*, t) \subsetneq \mathcal{D} \cap \{(\delta, \varepsilon) \mid \delta \leq \tau_1^*\}$ implies there is no $\rho_1^* \in T\Sigma^*$ such that $\mathcal{V}_{\mathcal{D}}(L)(\rho^* \cdot_t \rho_1^*, t') = \bot$ for any $t' \geq t + \tau(\rho_1^*)$.*

Proof. Let $\Delta_{\mathcal{D}}(L,\rho^*,t) \subsetneq \mathcal{D} \cap \{(\delta,\varepsilon) \mid \delta \leq \tau_1^*\}$, i.e., there is a $(\delta,\varepsilon) \in \mathcal{D}$ with $\delta \leq \tau_1^*$ and $(\delta,\varepsilon) \notin \Delta_{\mathcal{D}}(L,\rho^*,t)$. Thus, Lemmas 3.9 and 3.12 imply that $(\delta,\varepsilon) \in \Delta_{\mathcal{D}}(\overline{L},\rho^* \cdot_t \rho_1^*,t')$ for all ρ_1^* with $t \geq \tau(\rho^*)$ and $t' \geq t + \tau(\rho_1^*)$. Hence, $\Delta_{\mathcal{D}}(\overline{L},\rho^* \cdot_t \rho_1^*,t') \neq \emptyset$ for all such ρ_1^* and all such t'. Finally, Lemma 3.11 yields $\mathcal{V}_{\mathcal{D}}(L)(\rho^* \cdot_t \rho_1^*,t') \neq \top$ for all such ρ_1^* and all such t'.

The second claim is proven by a dual argument (swapping L with $\overline{L}$ and $\top$ with $\bot$). □

Note that $\Delta_{\mathcal{D}}(L,\rho^*,t) \subsetneq \mathcal{D} \cap \{(\delta,\varepsilon) \mid \delta \leq \tau_1^*\}$ and $\Delta_{\mathcal{D}}(\overline{L},\rho^*,t) \subsetneq \mathcal{D} \cap \{(\delta,\varepsilon) \mid \delta \leq \tau_1^*\}$ can both be true simultaneously (as in Example 3.13). In this situation, we will under no future observation reach a conclusive verdict.

4 Towards an Algorithm

Typically, monitoring algorithms rely on automata-based techniques. To this end, first the specification and its complement are translated into suitable automata. Then one computes the set of states reachable by processing the observation and then checks whether from one of these states the automaton can still accept an infinite continuation. If this is the case for both automata, then the verdict is *?*, if it is only the case for the automaton for the specification, then the verdict is $\top$, and vice versa for the complement automaton and $\bot$.

We want to follow the same blueprint, but we need to make adjustments to handle delay. Intuitively, we need to compute all states that are reachable by possible ground-truths of a given observation. However, a ground-truth may contain more events than the observation, as some events may not yet have been observed due to delay. This complicates the construction of the set of reachable states, as an unbounded number of events may not yet have been observed.

In the definition of $GT_{\mathcal{D}}$ (Definition 3.2) there is an implicit universal quantification over all possible sequences of such events that have not yet been observed (e.g., the last five events in ρ in Figure 3). We exploit the fact that the verdicts are defined with respect to all possible extensions μ of a possible ground-truth (i.e., also a universal quantification over the μ's) to "merge" the universal quantification over events that have not yet been observed into the universal quantification of the extension μ. Then, a possible ground-truth has exactly the same number of events as the observation (i.e., ground-truth and observation have **equal length** (**EL**)). We begin defining this restricted notion of possible ground-truth by strengthening Definition 3.2.

Definition 4.1 (EL-Consistency). Let $\rho^* = (\sigma_1^*,\tau_1^*),\dots,(\sigma_m^*,\tau_m^*)$ be a $\{(\delta,\varepsilon)\}$-observation and $\rho = (\sigma_1,\tau_1),\dots,(\sigma_n,\tau_n)$ be a finite timed word. We say that ρ is *EL-consistent* with ρ^* at observation time $t \in \mathbb{R}_{\geq 0}$ under latency δ and jitter ε if and only if ρ is consistent with ρ^* at t under δ and ε and $m = n$. We denote the set of timed words ρ that are EL-consistent with an $\{(\delta,\varepsilon)\}$-observation ρ^* at observation time t under latency δ and jitter ε by $GT^{\mathrm{el}}_{\delta,\varepsilon}(\rho^*,t)$ and define $GT^{\mathrm{el}}_{\mathcal{D}}(\rho^*,t) = \bigcup_{(\delta,\varepsilon)\in\mathcal{D}} GT^{\mathrm{el}}_{\delta,\varepsilon}(\rho^*,t)$.

Example 4.2. Continuing Example 3.3, an EL-consistent ground-truth of the observation ρ^* in Figure 3 has exactly four events corresponding to the four events in the observation. Thus, there cannot be any unobserved events between $t-(\delta+\varepsilon)$ and t in an EL-consistent ground-truth (e.g., the last five events of ρ in Figure 3).

The following lemma relates the original definition of consistency with EL-consistency.

Lemma 4.3.

(1) *Let ρ^* be a $\{(\delta,\varepsilon)\}$-observation, let $t \geq \tau(\rho^*)$, let $\rho \in GT^{\mathrm{el}}_{\delta,\varepsilon}(\rho^*,t)$, and let ρ' be a finite timed word with $\tau(\rho') \leq t - \max(\tau(\rho), t-(\delta+\varepsilon))$. Then, $\rho \cdot_{\max(\tau(\rho),t-(\delta+\varepsilon))} \rho' \in GT_{\delta,\varepsilon}(\rho^*,t)$.*

(2) *Let ρ^* be a $\{(\delta,\varepsilon)\}$-observation (say with m letters), let $t \geq \tau(\rho^*)$, let $\rho \in GT_{\delta,\varepsilon}(\rho^*,t)$, and let ρ' be the prefix of ρ with m letters. Then, $\rho' \in GT^{\mathrm{el}}_{\delta,\varepsilon}(\rho^*,t)$.*

Proof. (1) We have to show that $\rho \cdot_{\max(\tau(\rho),t-(\delta+\varepsilon))} \rho'$ is consistent with ρ^* at t under δ and ε. This follows directly from the fact that all events in ρ' have time points (in $\rho \cdot_{\max(\tau(\rho),t-(\delta+\varepsilon))} \rho'$) in the interval $[t-(\delta+\varepsilon), t]$ and are therefore covered by the third requirement of the definition of consistency.

(2) We need to show that ρ' is EL-consistent with ρ^* at t under δ and ε. By definition, ρ' has the same length as ρ^* and the first two requirements of the definition of consistency are satisfied, as ρ is consistent with ρ^* at t under δ and ε and ρ' is a prefix of ρ. Hence, it is EL-consistent, as the third requirement only refers to ground-truths that have more letters than the observation. □

Now, we present the revised verdict function using only EL ground-truths. Note that merging the unobserved events from the possible ground-truth ρ into the extension μ requires changing the time instant at which we concatenate the ground-truth and the extension: $t-(\delta+\varepsilon)$ is the earliest time point at which an event can occur that may not yet have been observed at time t. Due to jitter however, there might also be events after $t-(\delta+\varepsilon)$ that have been observed, which are in the possible ground-truth ρ: the last such event happened at time $\tau(\rho)$. Hence, we need to concatenate at time point $\max(\tau(\rho), t-(\delta+\varepsilon))$.

Definition 4.4 (Monitor Verdicts under Delay – EL Version). Given $L \subseteq T\Sigma^\omega$, a set $\mathcal{D}$ of delays, a $\mathcal{D}$-observation $\rho^* \in T\Sigma^*$, and $t \geq \tau(\rho^*)$, the function $\mathcal{V}^{\text{el}}_{\mathcal{D}} : 2^{T\Sigma^\omega} \to T\Sigma^* \times \mathbb{R}_{\geq 0} \to \mathbb{B}_3$ evaluates to the verdict

$$\mathcal{V}^{\text{el}}_{\mathcal{D}}(L)(\rho^*,t) = \begin{cases} \top & \text{if } \rho \cdot_{\max(\tau(\rho),t-(\delta+\varepsilon))} \mu \in L \text{ for all } (\delta,\varepsilon) \in \mathcal{D}, \text{ all } \rho \in GT^{\text{el}}_{\delta,\varepsilon}(\rho^*,t) \text{ and all } \mu \in T\Sigma^\omega, \\ \bot & \text{if } \rho \cdot_{\max(\tau(\rho),t-(\delta+\varepsilon))} \mu \notin L \text{ for all } (\delta,\varepsilon) \in \mathcal{D}, \text{ all } \rho \in GT^{\text{el}}_{\delta,\varepsilon}(\rho^*,t) \text{ and all } \mu \in T\Sigma^\omega, \\ ? & \text{otherwise.} \end{cases}$$

$\mathcal{V}^{\text{el}}_{\mathcal{D}}(L)(\rho^*,t)$ is undefined when $t < \tau(\rho^*)$.

Next, we show that both verdict functions coincide.

Lemma 4.5. *$\mathcal{V}^{\text{el}}_{\mathcal{D}}(L)(\rho^*,t) = \mathcal{V}_{\mathcal{D}}(L)(\rho^*,t)$ for all $L \subseteq T\Sigma^\omega$, all sets $\mathcal{D}$ of delays, all $\mathcal{D}$-observations ρ^*, and all $t \geq \tau(\rho^*)$.*

Proof. Let $\mathcal{V}_{\mathcal{D}}(L)(\rho^*,t) = \top$. We show $\mathcal{V}^{\text{el}}_{\mathcal{D}}(L)(\rho^*,t) = \top$ by proving that we have $\rho' \cdot_{\max(\tau(\rho'),t-(\delta+\varepsilon))} \mu' \in L$ for all $\rho' \in GT^{\text{el}}_{\delta,\varepsilon}(\rho^*,t)$ for some $(\delta,\varepsilon) \in \mathcal{D}$ and all $\mu' = (\sigma_1,\tau_1),(\sigma_2,\tau_2),\ldots \in T\Sigma^\omega$.

First, consider the case where $\tau(\rho') < t-(\delta+\varepsilon)$. Let n be maximal with $\tau_n \leq \delta+\varepsilon$ (this is well-defined due to time-divergence), let $\rho'_1 = (\sigma_1,\tau_1),\ldots,(\sigma_n,\tau_n)$ and $\mu'_2 = ((\sigma_{n+1},\tau_{n+1}),(\sigma_{n+2},\tau_{n+2}),\ldots) - (\delta+\varepsilon)$. Note that μ'_2 is well-defined as τ_{n+1} is, by the choice of n, greater than $\delta+\varepsilon$. Then, Lemma 4.3.Lemma 1 yields that $\rho' \cdot_{t-(\delta+\varepsilon)} \rho'_1$ is in $GT_{\delta,\varepsilon}(\rho^*,t)$ and Remark 2.1. Remark 3 yields

$$\rho' \cdot_{\max(\tau(\rho'),t-(\delta+\varepsilon))} \mu' = \rho' \cdot_{t-(\delta+\varepsilon)} \mu' = (\rho' \cdot_{t-(\delta+\varepsilon)} \rho'_1) \cdot_t \mu'_2.$$

Therefore, $\rho' \cdot_{\max(\tau(\rho'),t-(\delta+\varepsilon))} \mu'$ is the concatenation of the possible ground-truth $(\rho' \cdot_{t-(\delta+\varepsilon)} \rho'_1)$ of ρ^* and the suffix μ'_2. As we have $\rho \cdot_t \mu \in L$ for all $\rho \in GT_{\mathcal{D}}(\rho^*,t)$ and all $\mu \in T\Sigma^\omega$ (due to $\mathcal{V}_{\mathcal{D}}(L)(\rho^*,t) = \top$), we conclude $\rho' \cdot_{\max(\tau(\rho'),t-(\delta+\varepsilon))} \mu' \in L$ as required.

Now, consider the case where $\tau(\rho') \geq t-(\delta+\varepsilon)$. Note that we have $t-\tau(\rho') \geq 0$ due to $\rho' \in GT^{\text{el}}_{\delta,\varepsilon}(\rho^*,t)$. Hence, let n be maximal with $\tau_n \leq t-\tau(\rho')$ (again, this is well-defined due to time-divergence), let $\rho'_1 = (\sigma_1,\tau_1)\cdots(\sigma_n,\tau_n)$ and $\mu'_2 = ((\sigma_{n+1},\tau_{n+1})(\sigma_{n+2},\tau_{n+2})\cdots) - (t-\tau(\rho'))$. Again, μ'_2 is well-defined due to the choice of n. Then, Lemma 4.3. Lemma 1 yields that $\rho' \cdot_{\tau(\rho')} \rho'_1$ is in $GT_{\delta,\varepsilon}(\rho^*,t)$ and Remark 2.1. Remark 3 yields

$$\rho' \cdot_{\max(\tau(\rho'),t-(\delta+\varepsilon))} \mu' = \rho' \cdot_{\tau(\rho')} \mu' = (\rho' \cdot_{\tau(\rho')} \rho'_1) \cdot_t \mu'_2.$$

As $\rho' \cdot_{\max(\tau(\rho'),t-(\delta+\varepsilon))} \mu'$ is the concatenation of a possible ground-truth of ρ^* and an arbitrary suffix, it is again, as required, in L.

Using a dual argument (i.e., swapping $\top$ with $\bot$ and L with $\overline{L}$), we can show that $\mathcal{V}_{\mathcal{D}}(L)(\rho^*,t) = \bot$ implies $\mathcal{V}^{\mathrm{el}}_{\mathcal{D}}(L)(\rho^*,t) = \bot$.

Now, we show that $\mathcal{V}^{\mathrm{el}}_{\mathcal{D}}(L)(\rho^*,t) = \top$ implies $\mathcal{V}_{\mathcal{D}}(L)(\rho^*,t) = \top$. A dual argument again shows that $\mathcal{V}^{\mathrm{el}}_{\mathcal{D}}(L)(\rho^*,t) = \bot$ implies $\mathcal{V}_{\mathcal{D}}(L)(\rho^*,t) = \bot$. This will then complete our proof, as both functions only have three elements in their codomain and we have shown that two of them have the same preimage w.r.t. both functions.

So, let $\mathcal{V}^{\mathrm{el}}_{\mathcal{D}}(L)(\rho^*,t) = \top$. We show $\mathcal{V}_{\mathcal{D}}(L)(\rho^*,t) = \top$ by showing $\rho \cdot_t \mu \in L$ for all $\rho = (\sigma_1,\tau_1),\dots,(\sigma_n,\tau_n) \in GT_{\mathcal{D}}(\rho^*,t)$ and all $\mu = (\sigma_1',\tau_1'),(\sigma_2',\tau_2'),\dots \in T\Sigma^\omega$. By definition, there is a $(\delta,\varepsilon) \in \mathcal{D}$ such that $\rho \in GT_{\delta,\varepsilon}(\rho^*,t)$.

Let ρ^* have m letters. If $m = n$, then we also have $\rho \in GT^{\mathrm{el}}_{\mathcal{D}}(\rho^*,t)$. We consider two cases: If $\tau(\rho) < t-(\delta+\varepsilon)$, then an application of Remark 2.1. Remark 1 yields

$$\rho \cdot_t \mu = \rho \cdot_{t-(\delta+\varepsilon)} (\mu + (\delta+\varepsilon)) = \rho \cdot_{\max(\tau(\rho),t-(\delta+\varepsilon))} (\mu + (\delta+\varepsilon)),$$

and if $\tau(\rho) \geq t-(\delta+\varepsilon)$, then an application of Remark 2.1. Remark 1 yields

$$\rho \cdot_t \mu = \rho \cdot_{\tau(\rho)} (\mu + (t-\tau(\rho))) = \rho \cdot_{\max(\tau(\rho),t-(\delta+\varepsilon))} (\mu + (t-\tau(\rho))),$$

where $t-\tau(\rho)$ is nonnegative by definition of consistency. Hence, in both cases, $\rho \cdot_t \mu$ is the concatenation of a possible EL ground-truth of ρ^* and an arbitrary suffix. Due to $\mathcal{V}^{\mathrm{el}}_{\mathcal{D}}(L)(\rho^*,t) = \top$, all concatenations $\rho \cdot_{\max(\tau(\rho),t-(\delta+\varepsilon))} \mu$ for $(\delta,\varepsilon) \in \mathcal{D}$, $\rho \in GT^{\mathrm{el}}_{\delta,\varepsilon}(\rho^*,t)$, and $\mu \in T\Sigma^\omega$ are in L, which yields $\rho \cdot_t \mu \in L$.

It remains to consider the case where $n > m$, which we again split into two subcases. But first let us define $\rho_1 = (\sigma_1,\tau_1)\cdots(\sigma_m,\tau_m)$ as well as $\rho_2 = (\sigma_{m+1},\tau_{m+1})\cdots(\sigma_n,\tau_n)$. Lemma 4.3. Lemma 2 yields $\rho_1 \in GT^{\mathrm{el}}_{\mathcal{D}}(\rho^*,t)$.

First, consider the subcase where $\tau(\rho_1) < t-(\delta+\varepsilon)$. By definition of consistency, $n > m$ implies $\tau_{m+1} + \delta + \varepsilon \geq t$, and thus $\tau_{m+1} \geq t-(\delta+\varepsilon)$ (†). Hence, an application of Remark 2.1. Remark 2 yields

$$\rho \cdot_t \mu = \rho_1 \cdot_{t-(\delta+\varepsilon)} [(\rho_2 - (t-(\delta+\varepsilon))) \cdot_{\delta+\varepsilon} \mu] = \rho_1 \cdot_{\max(\tau(\rho_1),t-(\delta+\varepsilon))} [(\rho_2 - (t-(\delta+\varepsilon))) \cdot_{\delta+\varepsilon} \mu].$$

Note that the first time point of ρ_2, τ_{m+1}, is at least $(t-(\delta+\varepsilon))$ as required, as $\tau_{m+1} \geq t-(\delta+\varepsilon)$ (see †). Hence, $\rho \cdot_t \mu$ is the concatenation of a possible EL ground-truth of ρ^* and an arbitrary suffix and therefore in L.

Finally, consider the subcase where $\tau(\rho_1) \geq t-(\delta+\varepsilon)$. An application of Remark 2.1. Remark 2 yields

$$\rho \cdot_t \mu = \rho_1 \cdot_{\tau(\rho_1)} [(\rho_2 - \tau(\rho_1)) \cdot_{t-\tau(\rho_1)} \mu] = \rho_1 \cdot_{\max(\tau(\rho_1),t-(\delta+\varepsilon))} [(\rho_2 - \tau(\rho_1)) \cdot_{t-\tau(\rho_1)} \mu].$$

Again, this is well-defined as we have $\tau_{m+1} \geq \tau(\rho_1)$ (as τ_{m+1} is the next time instant after $\tau_m = \tau(\rho_1)$ in ρ) and as $t \geq \tau(\rho_1)$ by the definition of consistency. Hence, $\rho \cdot_t \mu$ is again the concatenation of a possible EL ground-truth of ρ^* and an arbitrary suffix and therefore in L. □

Next, we show that we can indeed make the definition of $\mathcal{V}^{\mathrm{el}}$ effective using automata-theoretic constructions. First, we formally capture the set of states that can be reached by processing the possible EL ground-truths of an observation. Let $\mathcal{A}$ be a TBA. We write $(q_0,v_0) \xrightarrow{\rho}_{\mathcal{A}} (q_n,v_n)$ for a finite timed word $\rho = (\sigma,\tau) \in T\Sigma^*$ to denote the existence of a finite sequence of states $(q_0,v_0) \xrightarrow{(\sigma_1,\tau_1)} (q_1,v_1) \xrightarrow{(\sigma_2,\tau_2)} \cdots \xrightarrow{(\sigma_n,\tau_n)} (q_n,v_n)$ of $\mathcal{A}$ where for all $1 \leq i \leq n$ there is a transition $(q_{i-1},q_i,\sigma_i,\lambda_i,g_i)$ of $\mathcal{A}$ such that $v_i(x) = 0$ for all x in λ_i and $v_{i-1}(x) + (t_i - t_{i-1})$ otherwise,

and g is satisfied by the valuation $v_{i-1} + (t_i - t_{i-1})$, where we use $t_0 = 0$. Given a TBA $\mathcal{A}$, a set $\mathcal{D}$ of delays, a finite observed timed word $\rho^* \in T\Sigma^*$, and $t \geq \tau(\rho^*)$, we define

$$\mathcal{R}^{\mathcal{D}}_{\mathcal{A}}(\rho^*, t) = \{(q, v + \max(0, (t - (\tau(\rho) + \delta + \varepsilon)))) \mid (q_0, v_0) \xrightarrow{\rho}_{\mathcal{A}} (q, v) \text{ where}$$
$$(q_0, v_0) \text{ with } q_0 \in Q_0,\ v_0(x) = 0 \text{ for all } x \in C, \text{ and } \rho \in GT^{\text{el}}_{\delta,\varepsilon}(\rho^*, t) \text{ for some } (\delta, \varepsilon) \in \mathcal{D}\}.$$

We call this the reach-set of ρ^* in $\mathcal{A}$ at t w.r.t. $\mathcal{D}$.

Next, we define the set of states of a TBA from where it is possible to reach an accepting location infinitely many times in the future, i.e., those states from which an accepting run is possible. This is useful, because if processing a finite timed word leads to such a state, then the timed word can be extended to an infinite one in the language of the automaton, a notion that underlies the definitions of the verdict functions. Given a TBA $\mathcal{A} = (Q, Q_0, \Sigma, C, \Delta, \mathcal{F})$, the set of states with nonempty language is

$$S^{ne}_{\mathcal{A}} = \{(q, v) \mid q \in Q, v \in C \to \mathbb{R}_{\geq 0} \text{ s.t. } L(\mathcal{A}, (q, v)) \neq \emptyset\}.$$

The set $S^{ne}_{\mathcal{A}}$ can be computed using a zone-based fixpoint algorithm [20]. Using these definitions, we can give an *effective* definition of the verdict functions, which we show to be equivalent to the previous definitions and implementable.

In the following definition, $\mathbb{A}$ denotes the set of all TBA.

Definition 4.6 (Monitoring TBA). Given a TBA $\mathcal{A}$, a complement automaton $\overline{\mathcal{A}}$ (i.e., with $L(\overline{\mathcal{A}}) = T\Sigma^\omega \setminus L(\mathcal{A})$), a set $\mathcal{D}$ of delays, a $\mathcal{D}$-observation $\rho^* \in T\Sigma^*$, and $t \geq \tau(\rho)$, $\mathcal{M}_{\mathcal{D}} : \mathbb{A} \times \mathbb{A} \to T\Sigma^* \times \mathbb{R}_{\geq 0} \to \mathbb{B}_3$ computes the verdict

$$\mathcal{M}_{\mathcal{D}}(\mathcal{A}, \overline{\mathcal{A}})(\rho^*, t) = \begin{cases} \top & \text{if } \mathcal{R}^{\mathcal{D}}_{\overline{\mathcal{A}}}(\rho^*, t) \cap S^{ne}_{\overline{\mathcal{A}}} = \emptyset, \\ \bot & \text{if } \mathcal{R}^{\mathcal{D}}_{\mathcal{A}}(\rho^*, t) \cap S^{ne}_{\mathcal{A}} = \emptyset, \\ ? & \text{otherwise.} \end{cases}$$

$\mathcal{M}_{\mathcal{D}}(\mathcal{A}, \overline{\mathcal{A}})(\rho^*, t)$ is undefined if $t < \tau(\rho)$.

Next we show that this automata-based definition of monitoring is equal to the verdict functions defined above.

Theorem 4.7. *$\mathcal{M}_{\mathcal{D}}(\mathcal{A}, \overline{\mathcal{A}})(\rho^*, t) = \mathcal{V}^{\text{el}}_{\mathcal{D}}(L(\mathcal{A}))(\rho^*, t)$ for all sets $\mathcal{D}$ of delays, all TBA $\mathcal{A}$ (and complement automata $\overline{\mathcal{A}}$), all $\mathcal{D}$-observations ρ^*, and all $t \geq \tau(\rho^*)$.*

Proof. We will show that $\mathcal{R}^{\mathcal{D}}_{\mathcal{A}'}(\rho^*, t) \cap S^{ne}_{\mathcal{A}'}$ is nonempty if and only if there exists a $\rho \in GT^{\text{el}}_{\delta,\varepsilon}(\rho^*, t)$ and a $\mu \in T\Sigma^\omega$ with $\rho \cdot_{\max(\tau(\rho), (t-(\delta+\varepsilon)))} \mu \in L(\mathcal{A}')$ for any TBA $\mathcal{A}'$. Then we obtain

- $\mathcal{M}_{\mathcal{D}}(\mathcal{A}, \overline{\mathcal{A}})(\rho^*, t) = \top$ if and only if $\mathcal{V}^{\text{el}}_{\mathcal{D}}(L(\mathcal{A}))(\rho^*, t) = \top$ by instantiating the equivalence for $\mathcal{A}' = \overline{\mathcal{A}}$, and
- $\mathcal{M}_{\mathcal{D}}(\mathcal{A}, \overline{\mathcal{A}})(\rho^*, t) = \bot$ if and only if $\mathcal{V}^{\text{el}}_{\mathcal{D}}(L(\mathcal{A}))(\rho^*, t) = \bot$ by instantiating the equivalence for $\mathcal{A}' = \mathcal{A}$.

This completes the proof, as both functions only have three elements in their codomain and we have shown that two of them have the same preimage w.r.t. both functions.

So, let $\mathcal{R}^{\mathcal{D}}_{\mathcal{A}'}(\rho^*, t) \cap S^{ne}_{\mathcal{A}'} \neq \emptyset$. Then, by definition, there is a state (q, v') of $\mathcal{A}'$ such that

- $(q_0, v_0) \xrightarrow{\rho}_{\mathcal{A}'} (q, v)$ for some initial state (q_0, v_0) of $\mathcal{A}'$, some $\rho \in GT^{\text{el}}_{\delta,\varepsilon}(\rho^*, t)$ for some $(\delta, \varepsilon) \in \mathcal{D}$, and $v' = v + \max(0, (t - (\tau(\rho) + \delta + \varepsilon)))$, and
- there is an accepting infinite run of $\mathcal{A}'$ starting in (q, v') that processes some $\mu \in T\Sigma^\omega$.

These two runs can be combined into an accepting run of $\mathcal{A}'$ that starts in (q_0, v_0) and processes

$$\rho \cdot_{\tau(\rho)+\max(0,(t-(\tau(\rho)+\delta+\varepsilon)))} \mu = \rho \cdot_{\max(\tau(\rho),(t-(\delta+\varepsilon)))} \mu,$$

which implies that it is in $L(\mathcal{A}')$ as required.

Conversely, let there be a $\rho \in GT^{\mathrm{el}}_{\delta,\varepsilon}(\rho^*, t)$ and a $\mu \in T\Sigma^\omega$ with

$$\mu \cdot_{\max(\tau(\rho),(t-(\delta+\varepsilon)))} \mu \in L(\mathcal{A}').$$

Then, there exists an accepting run of $\mathcal{A}'$ starting in some initial state (q_0, v_0) that processes $\rho \cdot_{\max(\tau(\rho),(t-(\delta+\varepsilon)))} \mu$. This run can be split into

- $(q_0, v_0) \xrightarrow{\rho}_{\mathcal{A}'} (q, v)$ for some state (q, v) of $\mathcal{A}'$ and
- an accepting infinite run of $\mathcal{A}'$ starting in (q, v') that processes μ, where

$$v' = v + \max(0, (t - (\tau(\rho) + \delta + \varepsilon))).$$

Hence, $(q, v') \in \mathcal{R}^{\mathcal{D}}_{\mathcal{A}'}(\rho^*, t) \cap S^{ne}_{\mathcal{A}'}$, which is therefore, as required, nonempty. □

Recall that $S^{ne}_{\mathcal{A}}$ can be computed for any given TBA $\mathcal{A}$. Therefore, in the next section, we show how to calculate $\mathcal{R}^{\mathcal{D}}_{\mathcal{A}}(\rho^*, t)$ for a given TBA $\mathcal{A}$, set $\mathcal{D}$ of delays, observation ρ^*, and time point t using a zone-based algorithm. This will then allow us to compute verdicts effectively.

5 A Zone-Based Online Monitoring Algorithm

In this section, we demonstrate how to compute the reach-set of ρ^* in $\mathcal{A}$ at t w.r.t. $\mathcal{D}$. So far we have developed the theory with observations, latency, and jitter being reals. Now, we are concerned with algorithms and thus assume all these quantities to be rationals. For the monitoring algorithm, we use—as standard in analysing timed automata models—symbolic states being pairs (q, Z) of locations and zones. A zone is a finite conjunction of constraints of the form $x \sim t$ and $x - x' \sim t$ for clocks x, x', constants $t \in \mathbb{Q}_{\geq 0}$, and $\sim \in \{<, \leq, =, \geq, >\}$. Given two zones Z and Z' over a set C of clocks, and a set of clocks $\lambda \subseteq C$, we define the following operations on zones (which can be efficiently implemented using the DBM data structure [8]):

- $Z[\lambda] = \{v \mid \exists v' \vDash Z \text{ s.t. } v(x) = 0 \text{ if } x \in \lambda, \text{otherwise } v(x) = v'(x)\}$
- $Z^{\nearrow} = \{v \mid \exists v' \vDash Z \text{ s.t. } v = v' + d \text{ for some } d \in \mathbb{R}_{\geq 0}\}$
- $Z \wedge Z' = \{v \mid v \vDash Z \textit{ and } v \vDash Z'\}$

We can use these functions to compute the successor states after an input. Given a TBA $\mathcal{A} = (Q, Q_0, \Sigma, C, \Delta, \mathcal{F})$, a symbolic state (q, Z), and a letter $a \in \Sigma$, we define

$$\mathrm{Post}((q, Z), a) = \{(q', Z') \mid (q, q', a, \lambda, g) \in \Delta, Z' = (Z^{\nearrow} \wedge g)[\lambda]\},$$

as the set of states one can reach by taking an a-transition at some point in the future from (q, Z). Using Post we can compute the successor states of a timed input $(a, \tau) \in \Sigma \times \mathbb{Q}_{\geq 0}$ by extending the zones with an additional clock *time* just recording time since system start. The set of successors of a symbolic state is

$$\mathrm{Succ}((q, Z), (a, \tau)) = \{(q', Z') \mid (q', Z'') \in \mathrm{Post}((q, Z), a), Z' = Z'' \wedge \textit{time} = \tau\}$$

and the set of successors of a set of symbolic states S is

$$\mathrm{Succ}(S, (a, \tau)) = \bigcup_{(q', Z') \in S} \mathrm{Succ}((q', Z'), (a, \tau)).$$

In handling delayed observations, we assume that the delay set has the form

$$\mathcal{D} = \{(\delta, \varepsilon) \mid \delta \in [\ell, u]\}$$

for given $l, u, \varepsilon \in \mathbb{Q}_{\geq 0}$, i.e., the latency δ is bounded by an interval $[\ell, u] \subseteq \mathbb{R}_{\geq 0}$ and the jitter is bounded by $\varepsilon \in \mathbb{Q}_{\geq 0}$.

To represent the latency and thereby be able to reason about and indirectly store the latency bounds, we add a clock *etime* representing the "expected" real time that an event generated just now could be observed by the monitor after having been delayed according to the latency. This allows us

(1) to represent the actual latency as $\mathit{etime} - \mathit{time}$,
(2) to represent the initial knowledge about latencies by initializing $\mathit{etime} - \mathit{time}$ to the initially known bounds on latency, namely $\mathit{etime} - \mathit{time} \in [\ell, u]$ by setting *time* to 0 and constraining *etime* to $[\ell, u]$, and
(3) to refine our knowledge about the actual latency after having observed an event (σ^*, τ^*) by then setting *etime* to a value in $[\tau^* - \varepsilon, \tau^*]$.

Consequently, we change the initial zones to include the latency bounds ℓ and u as the differences between the clocks *etime* and *time*. This way, *etime* represents the expected time an event is observed at the monitor, given ℓ and u, and *time* represents the actual time the event happened (at the system). The aforementioned refinement (see Item 3 above and Figure 4) then permits to deduce actual latency ranges consistent with the specification (or its negation) from observation times of events.

In detail, this refinement of the $\mathit{etime} - \mathit{time}$ relation works as follows. Given a TBA $\mathcal{A}$ extended with the clocks *time* and *etime*, and an observation $(\sigma, \tau^*) \in \Sigma \times \mathbb{Q}_{\geq 0}$, the successors of (q, Z) are

$$\mathrm{Succ}_d((q,Z),(\sigma,\tau^*)) = \{(q', Z') \mid (q', Z'') \in \mathrm{Post}((q,Z),\sigma), Z' = Z'' \wedge \mathit{etime} \leq \tau^* \wedge \mathit{etime} \geq \tau^* - \varepsilon\}$$

and the successors $\mathrm{Succ}_d(S, (\sigma, \tau^*))$ of a set of symbolic states S is equal to $\bigcup_{(q,Z)\in S} \mathrm{Succ}_d((q,Z),(\sigma,\tau^*))$.

The online monitoring algorithm will essentially apply Succ_d repeatedly to update the reach-set, once for each new observation. Note that there is a slight mismatch, as Succ_d is computed with the two auxiliary clocks *time* and *etime*, which are not clocks of $\mathcal{A}$.

The initial reach-set is given by the following zone Z_0^d requiring all ordinary clocks of the TBA $\mathcal{A}$ to be zero and with *time* and *etime* satisfying $\mathit{etime} - \mathit{time} \in [\ell, u]$. That is

$$Z_0^d \equiv \underbrace{\mathit{etime} - \mathit{time} \leq u \wedge \mathit{time} - \mathit{etime} \leq -\ell}_{\mathit{etime}-\mathit{time}\in[\ell,u]} \wedge \underbrace{\bigwedge_{x\in C\cup\{\mathit{time}\}} x = 0}_{x_1,\ldots,x_{|C|}=0,\ \mathit{time}=0} .$$

Given a fixed jitter bound ε, we can now compute the reach-set after a sequence of observations under delay, where the latency is bounded in $[\ell, u]$.

Lemma 5.1. *Given a TBA $\mathcal{A}$, a delay set $\mathcal{D} = \{(\delta, \varepsilon) \mid \delta \in [\ell, u]\}$ with $\ell, u, \varepsilon \in \mathbb{Q}_{\geq 0}$, a $\mathcal{D}$-observation $\rho^* = (\sigma_1, \tau_1^*), \ldots, (\sigma_n, \tau_n^*)$, and $t \in \mathbb{Q}_{\geq 0}$ with $t \geq \tau_n^*$, let $S_0 = \{(q_0, Z_0^d) \mid q_0 \in Q_0\}$ and $S_i = \mathrm{Succ}_d(S_{i-1}, (\sigma_i, \tau_i^*))$ for $i \in \{1, \ldots, n\}$. Then, the reach-set $\mathcal{R}_{\mathcal{A}}^{\mathcal{D}}(\rho^*, t)$ is the projection of*

$$\{(q', Z') \mid (q', Z'') \in S_n, Z' = Z''^{\nearrow} \wedge \mathit{etime} = t - \varepsilon\}$$

to the clocks of $\mathcal{A}$ (obtained by removing all constraints on time and etime).

Proof. Given the form of $\mathcal{D}$ we can rewrite the definition of the reach-set to

$$\mathcal{R}_{\mathcal{A}}^{\mathcal{D}}(\rho^*, t) = \{(q, v + \max(0, (t - (\tau(\rho) + \delta + \varepsilon)))) \mid (q_0, v_0) \xrightarrow{\rho}_{\mathcal{A}} (q, v) \text{ where}$$
$$(q_0, v_0) \text{ with } q_0 \in Q_0,\ v_0(x) = 0 \text{ for all } c \in C, \text{ and } \rho \in GT_{\delta,\varepsilon}^{\mathrm{el}}(\rho^*, t) \text{ for some } \delta \in [\ell, u]\}.$$

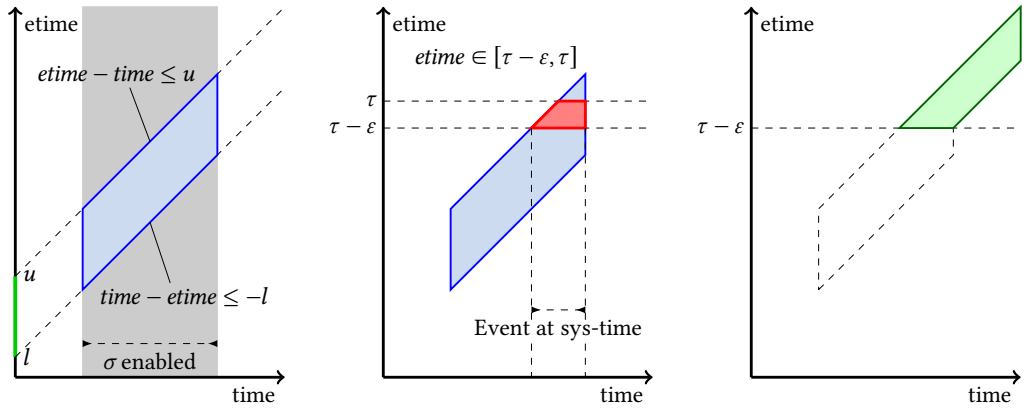

Fig. 4. Illustration of a single zone in the Succ_d computation (only *time*-*etime* plane depicted). Left: initial zone (in green) is diagonally extrapolated for time passage and then intersected with the guard of an edge. Middle: observing event σ at time τ. By restricting *etime* to $[\tau-\varepsilon,\tau]$, the clock *time* is restricted to when the event could have occurred at the system. Right: computing the future zone we see that the bound on *time* − *etime* is now stricter and thus the bounds for the consistent latencies are refined.

Now, extending the transition relation $\xrightarrow{\rho}_{\mathcal{A}}$ to clock valuations over the extended set of clocks $C \cup \{time, etime\}$, we may further reformulate the reach-set as follows:

$$\begin{aligned}\mathcal{R}^{\mathcal{D}}_{\mathcal{A}}(\rho^*,t) = \{(q_n, v^* + \max(0,(t-(v^*_n(etime)+\varepsilon)))) \mid (q_0,v^*_0) \xrightarrow{\rho}_{\mathcal{A}} (q_n,v^*_n) \text{ where}\\ (q_0,v^*_0) \text{ with } q_0 \in Q_0,\ v^*_0(x)=0 \text{ for all } c\in C, \text{ and}\\ v^*_0(etime) - v^*_0(time) \in [\ell,u] \text{ and } v^*_i(etime) \le \tau^*_i \wedge v^*_i(etime) \ge \tau^*_i - \varepsilon \wedge\\ \sigma_i = \sigma^*_i \text{ for } i \in \{0,\dots,n\}\}.\end{aligned}$$

A key observation for the correctness of the above reformulation is that the extended clocks *etime* and *time* are *not* modified by the TBA $\mathcal{A}$. That is $v^*_i(etime) - v^*_i(time) = v^*_0(etime) - v^*_0(time) \in [\ell,u]$ for all $i = \{0,\dots,n\}$.

Now let $\mathcal{R}^{\mathcal{D},j}_{\mathcal{A}}(\rho^*,t)$ for $j \in \{0,\dots,n\}$ be defined as follows:

$$\begin{aligned}\mathcal{R}^{\mathcal{D},j}_{\mathcal{A}}(\rho^*,t) = \{(q_j, v^*_j) \mid (q_0,v^*_0) \xrightarrow{\rho}_{\mathcal{A}} (q_j,v^*_j) \text{ where}\\ (q_0,v^*_0) \text{ with } q_0 \in Q_0,\ v^*_0(x)=0 \text{ for all } c\in C, \text{ and}\\ v^*_0(etime) - v^*_0(time) \in [\ell,u] \text{ and } v^*_i(etime) \le \tau^*_i \wedge v^*_i(etime) \ge \tau^*_i - \varepsilon \wedge\\ \sigma_i = \sigma^*_i \text{ for } i \in \{0,\dots,j\}\}.\end{aligned}$$

Then clearly $\mathcal{R}^{\mathcal{D},0}_{\mathcal{A}}(\rho^*,t) = \{(q_0, Z^d_0) \mid q_0 \in Q_0\} = S_0$ and $\mathcal{R}^{\mathcal{D},j}_{\mathcal{A}}(\rho^*,t) = \mathrm{Succ}_d(\mathcal{R}^{\mathcal{D},j-1}_{\mathcal{A}}(\rho^*,t), (\sigma_i,\tau_i))$ for $j \in \{1,\dots,n\}$, using standard arguments for the correctness of symbolic exploration of timed automata (see, e.g., [8]). Finally, as $(t-(v^*_n(etime)+\varepsilon))) = ((t-\varepsilon) - v^*_n(etime))$ it follows that $\mathcal{R}^{\mathcal{D}}_{\mathcal{A}}(\rho^*,t) = \mathcal{R}^{\mathcal{D},n}_{\mathcal{A}}(\rho^*,t)^{\nearrow} \wedge etime = t-\varepsilon$. □

This lemma allows us to implement a monitoring algorithm by computing the reach-sets and intersecting them with the set of nonempty language states as described in Definition 4.6 and proven correct in Theorem 4.7.

Theorem 5.2. *The function $\mathcal{V}_{\mathcal{D}}(L(\mathcal{A}))$ is effectively computable for specifications given by TBA $\mathcal{A}$ and $\overline{\mathcal{A}}$ (for the complement), and $\mathcal{D} = \{(\delta,\varepsilon) \mid \delta \in [\ell,u]\}$ for $\ell,u,\varepsilon \in \mathbb{Q}_{\ge 0}$.*

The observation of events may lead to refinement of the difference between *time* and *etime* as depicted in Figure 4. This captures the definition of the sets of consistent delays from Section 3.

Lemma 5.3. *Given $\mathcal{A}$, $\mathcal{D}$, ρ^*, t, and S_n as in Lemma 5.1, we can compute the set of consistent delays by looking at the bounds on etime − time:* $\Delta_{\mathcal{D}}(L(\mathcal{A}), \rho^*, t) = \{(\delta, \varepsilon) \in \mathcal{D} \mid S_n \models etime - time = \delta\}$.

This information can be used to decorate the *?* verdict, so that we can report a set of bounds on the latency for which we would provide a $\top$ or $\bot$ verdict.

Example 5.4. Let us show an example of our algorithm for monitoring under delayed observation. Note that, for the sake of readability, we use sets of clock constraints instead of conjunctions of clock constraints when specifying zones.

Consider the property $\varphi = F_{[0,10]}a \wedge G_{[0,20]}\neg b$ from Figure 1. The TBA accepting $L(\varphi)$ and $L(\neg\varphi)$ are shown in Figure 2. The nonempty language states for $\mathcal{A}_\varphi$ and $\mathcal{A}_{\neg\varphi}$ are $S^{ne}_{\mathcal{A}_\varphi} = \{(q_0, \{x \leq 10\}), (q_1, \texttt{true}), (\varphi, \texttt{true})\}$ and $S^{ne}_{\mathcal{A}_{\neg\varphi}} = \{(q_0, \texttt{true}), (q_1, \{x \leq 20\}), (\neg\varphi, \texttt{true})\}$. Let us assume the latency is between 0 and 10, and the jitter is bounded by 0.2. Now we compute the reach-sets S_0 (initial), S_1 (after $(a, 17.3)$), and S_2 (after $(b, 27.5)$) as

$$
\begin{aligned}
S_0 &= \{(q_0, \{x = 0,\ etime \leq 10,\ (etime - x) \in [0, 10]\})\},\\
S_1 &= \{(q_1, \{x \in [7.1, 10],\ etime \in [17.1, 17.3],\ (etime - x) \in [7.1, 10]\}),\\
&\qquad (\neg\varphi, \{x \in [10, 17.3],\ etime \in [17.1, 17.3],\ (etime - x) \leq 7.3\})\}, \text{ and}\\
S_2 &= \{(\varphi, \{x \in (20, 20.4], etime \in [27.3, 27.5),\ (etime - x) \in [7.1, 7.5]\}),\\
&\qquad (\neg\varphi, \{x \in [17.3, 27.5],\ etime \in [27.3, 27.5],\ (etime - x) \in [0, 10]\})\}.
\end{aligned}
$$

Note that we omit the clock *time* and only look at x and *etime* since *time* and x always have the same constraints.

All reach-sets intersect with both sets of nonempty language states; thus, the verdict is *?*. However, we can refine this verdict with knowledge about the consistent delays that change after each observation. The jitter bound is fixed at 0.2, but the bounds on the latency can be found in the clock constraints on the difference between *etime* and x. For $\bot$, the latency range remains $[0, 10]$ in all reach-sets. For $\top$, the consistent latency range is $[0, 10]$ in S_0, $[7.1, 10]$ in S_1, and it is $[7.1, 7.5)$ in S_2. This means that if the latency is outside $[7.1, 7.5)$, then the verdict is $\bot$.

On the other hand, for the observation $\rho^* = (a, 17.3), (b, 27.1)$ from Example 3.7 (and using the same latency and jitter bounds as above), we compute the reach-sets S_0, S_1, and S_2' where

$$S_2' = \{(\neg\varphi, \{x \in [16.9, 27.1], etime \in [26.9, 27.1], (etime - x) \in [0, 10]\})\}.$$

As S_2' has an empty intersection with $S^{ne}_{\mathcal{A}_\varphi}$, the verdict is $\bot$.

6 Testing under Delay

After having tackled the problem of monitoring under delay, we now consider the problem of testing under delay. Testing is distinguished from passive monitoring by providing the system under observation with stimuli that are actively generated by the test harness. We therefore assume that the event alphabet Σ of our system is partitioned into an input alphabet Σ_I and an output alphabet Σ_O (where we take the point of view of the system under observation, i.e., events in Σ_I are inputs to the system and events in Σ_O are outputs of the system). The output channel behaves as in the case of passive monitoring, namely delivering messages from the SUT to the monitor with a delay given by a latency δ_O and a jitter ε_O. The input channel behaves symmetrically in that it delivers messages from the monitor to the SUT with again a (possibly different) delay given by a latency δ_I and a jitter ε_I. Our obligation then is to check satisfaction or violation of a specification L

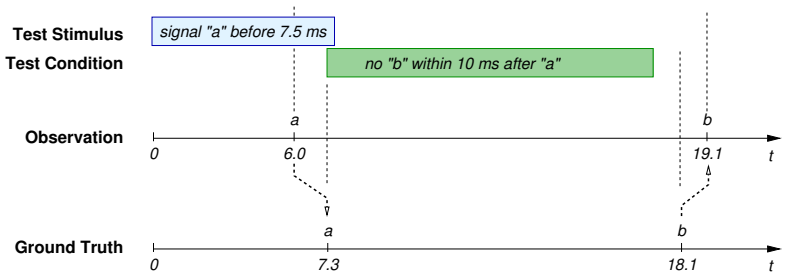

Fig. 5. Executing real-time tests via delayed channels: both the stimuli and the responses are mediated via delayed channels. Input and output delays to/from the test harness are unknown up to bounds. Assuming latency bounded by $\infty > \delta_I \geq \delta_O \geq 0$ and jitter of at most 0.05, a definite test verdict to the test specification $\big(F_{[0,7.5)}a\big) \rightarrow \big((\neg a)U(a \wedge G_{[0,10]}\neg b)\big)$, where a is an input to the SUT and b a response, is to be issued at time 19.1. Its value is $\top$, i.e., "passed", as either $\delta_I \geq 1.5$, in which case the antecedent of the implication is violated, or $1.5 > \delta_I \geq \delta_O$, in which case a was received strictly before $6.0 + 1.5 + 0.05 = 7.55$ while the observed b happened strictly after $19.1 - 1.5 - 0.05 = 17.55$.

over Σ by a system while providing stimuli (from Σ_I) over a delayed input channel and observing reactions (from Σ_O) over a delayed output channel. Both input delay and output delay are unknown up to given bounds.

Figure 5 illustrates the concept where inputs are observed by the test monitor before they actually arrive at the SUT, and outputs are observed by the monitor later than they are generated by the SUT. Due to the uncertainty about delays in the two directions, it is obvious that the observed order of events may be different from the order in which the events occurred on the system side. For example, in Figure 5, if $\delta_I = \delta_O = 10$ and $\varepsilon_I = \varepsilon_O = 1$ then the input event a would arrive at the SUT between 16.0 and 17.0, thus *after* the output event b observed at 19.1, which would have to be generated between 8.1 and 9.1 to be observed at 19.1. In our definitions, we will admit such overtaking between messages in the input channels and the output channels, but will —for simplicity of the exposition— assume in-order delivery of input messages, as well as in-order delivery of output messages (see Footnote 1), but will later restrict the setting when presenting our algorithm implementing testing under delay.

Due to overtaking only being allowed between inputs and outputs, the ground-truth corresponding to a monitor-side observation can be defined in terms of the two projections of the observed timed trace to the inputs and to the outputs, respectively. To simplify our notation, given a (finite or infinite) timed word ρ over Σ, we denote its Σ_I projection, i.e., the subsequence of pairs in $\Sigma_I \times \mathbb{R}_{\geq 0}$, as $\rho|_I$. Similarly, we denote ρ's Σ_O projection as $\rho|_O$. Both projections are timed words over the respective alphabet.

As for monitoring, we begin by formalizing delay sets and observations that can be made under delay.

Definition 6.1. An *IO* delay set $\mathcal{D}$ is a nonempty subset of $\mathbb{R}^4_{\geq 0}$ containing pairs of latencies and jitters for the input channel and pairs of latencies and jitters for the output channel. A $\mathcal{D}$-observation, i.e. an observation that can in principle be made under delay in $\mathcal{D}$, is a finite timed word $\rho^* = (\sigma_1^*, \tau_1^*), \ldots, (\sigma_m^*, \tau_m^*)$ such that there is a $(\delta_I, \varepsilon_I, \delta_O, \varepsilon_O,) \in \mathcal{D}$ with $\tau_i^* \geq \delta_O$, where i is minimal with $\sigma_i^* \in \Sigma_O$.

Note that we only need to constrain occurrences of outputs in the observation, as inputs can be send to the SUT at any time.

Next, we formalize the notion of ground-truth, i.e., executions of the SUT that are consistent with an observation.

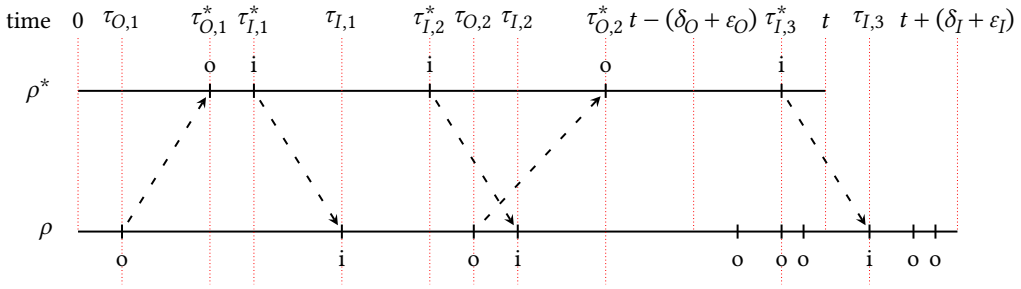

Fig. 6. A $\{(\delta,\varepsilon)\}$-observation ρ^* and a consistent ground-truth ρ.

Definition 6.2 (Testing Consistency). Let ρ^* be a $\{(\delta_I,\varepsilon_I,\delta_O,\varepsilon_O)\}$-observation with $\rho^*|_I = (\sigma^*_{I,1},\tau^*_{I,1}),\dots,(\sigma^*_{I,m_I},\tau^*_{I,m_I})$ and $\rho^*|_O = (\sigma^*_{O,1},\tau^*_{O,1}),\dots,(\sigma^*_{O,m_O},\tau^*_{O,m_O})$. Let ρ be a finite timed word with $\rho|_I = (\sigma_{I,1},\tau_{I,1}),\dots,(\sigma_{I,n_I},\tau_{I,n_I})$ and $\rho|_O = (\sigma_{O,1},\tau_{O,1}),\dots,(\sigma_{O,n_O},\tau_{O,n_O})$.

We say that ρ is *(testing) consistent* with ρ^* at observation time $t \in \mathbb{R}_{\geq 0}$ under $(\delta_I,\varepsilon_I,\delta_O,\varepsilon_O)$ if and only if

(1) $\tau(\rho^*) \leq t$ and $\tau(\rho) \leq \max(t, \tau(\rho^*|_I) + (\delta_I+\varepsilon_I))$,
(2I) $n_I = m_I$, and $\sigma_{I,i} = \sigma^*_{I,i}$ and $\tau_{I,i} - \tau^*_{I,i} \in [\delta_I, \delta_I+\varepsilon_I]$ for all $i \in \{1,\dots,m_I\}$,
(2O) $n_O \geq m_O$, and $\sigma_{O,i} = \sigma^*_{O,i}$ and $\tau^*_{O,i} - \tau_{O,i} \in [\delta_O, \delta_O+\varepsilon_O]$ for all $i \in \{1,\dots,m_O\}$, and
(3) if $n_O > m_O$ then $\tau_{m_O+1} \geq t - (\delta_O+\varepsilon_O)$.

We denote the set of timed words ρ that are consistent with a $\{(\delta_I,\varepsilon_I,\delta_O,\varepsilon_O)\}$-observation ρ^* at observation time t under $(\delta_I,\varepsilon_I,\delta_O,\varepsilon_O)$ by $\widehat{GT}_{(\delta_I,\varepsilon_I,\delta_O,\varepsilon_O)}(\rho^*,t)$. Then, we define $\widehat{GT}_{\mathcal{D}}(\rho^*,t) = \bigcup_{((\delta_I,\varepsilon_I,\delta_O,\varepsilon_O))\in\mathcal{D}} \widehat{GT}_{(\delta_I,\varepsilon_I,\delta_O,\varepsilon_O)}(\rho^*,t)$.

Before we continue, let us compare the previous definition to Definition 3.2, its analogue for monitoring. In Condition (1), we must allow the ground-truth to have a longer duration than the observation to account for inputs that arrive at the SUT after time t due to the delay. All such inputs must have arrived at time $\tau(\rho^*|_I) + (\delta_I+\varepsilon_I)$. Furthermore, we can take into account that we are at time point t, so we use $\max(t, \tau(\rho^*|_I) + (\delta_I+\varepsilon_I))$ as upper bound on the duration of the ground-truth. Further, condition (2) of definition 3.2 has to be stated for both channels independently. In Condition (2I), we need to consider the difference $\tau_{I,i} - \tau^*_{I,i}$ to account for the fact that inputs are sent from the test harness and arrive, after some delay, at the SUT. Also, we only consider ground-truths that have exactly the same number of inputs as the observation has (any additional input will be covered by extending the ground-truth to an infinite word). On the other hand, Condition (2O) and Condition (3) are similar to their counterparts in Definition 3.2.

Example 6.3. Figure 6 shows a $\{(\delta_I,\varepsilon_I,\delta_O,\varepsilon_O)\}$-observation and a consistent ground-truth and illustrates how the delay shifts the timestamps of the events. Here, i is an input in Σ_I and o is an output in Σ_O. The length of ρ is $n = 10$ and the length of ρ^* is $m = 5$. Recall that t is the time of observation.

In particular, notice the following:

— As in the case of testing, no output can occur in the observation ρ^* with a timestamp smaller than δ_O, as it takes at least δ_O units of time for an output to be send from the system through the output channel to the test harness. Obviously, at the system side (i.e., in the ground-truth ρ) events can happen at any timestamp, also before δ_O (e.g., the first o). Similarly, inputs can be send by the test harness at any time.

- The difference $\tau^*_{O,j} - \tau_{O,j}$ for $j \in \{1,2\}$ (i.e., the difference between the time an output is observed at the test harness and the time the event was emitted by the system) must be in the interval $[\delta_O, \delta_O + \varepsilon_O]$. Dually, the difference $\tau_{I,j} - \tau^*_{I,j}$ for $j \in \{1,2,3\}$ (i.e., the difference between the time an input is emitted by the test harness and the time the event was received by the system) must be in the interval $[\delta_I, \delta_I + \varepsilon_I]$.
  This can lead to overtaking between inputs and outputs, as evident by the second output being sent from the system before the second input arrives (i.e., $\tau_{O,2} < \tau_{I,2}$). At the test harness, the situation is reversed: Here, the second input is sent before the second output is received (i.e., $\tau^*_{O,2} > \tau^*_{I,2}$).
- As in the case of monitoring, there may be outputs in the ground-truth that have not been observed by the test harness, e.g., the last five outputs in ρ. As before, such events can only have timestamps in the interval $[t - (\delta_O + \varepsilon_O), \tau(\rho)]$, as all earlier events must necessarily have been observed at the test harness at time t. Dually, an input sent by the test harness before time t may arrive after t, e.g., the last input. However, all such events must have arrived at the system at time $t + (\delta_I + \varepsilon_I)$.

Remark 6.4. By extending the argument presented in Remark 3.4, one can show that $\widehat{GT}_{\mathcal{D}}(\rho^*, t)$ is always nonempty, if ρ^* is a $\mathcal{D}$-observation and $t \geq \tau(\rho^*)$.

Using these ground-truths, we can now define testing under delay.

Definition 6.5 (Testing Verdicts under Delay). Given a language $L \subseteq T\Sigma^\omega$, a set of possible observation delays $\mathcal{D}$, a $\mathcal{D}$-observation $\rho^* \in T\Sigma^*$, and an observation time $t \geq \tau(\rho^*)$, the function $\widehat{\mathcal{V}}_{\mathcal{D}} : 2^{T\Sigma^\omega} \to T\Sigma^* \times \mathbb{R}_{\geq 0} \to \mathbb{B}_3$ evaluates to the verdict

$$\widehat{\mathcal{V}}_{\mathcal{D}}(L)(\rho^*, t) = \begin{cases} \top & \text{if } \rho \cdot_{\max(t,\tau(\rho))} \mu \in L \text{ for all } \rho \in \widehat{GT}_{\mathcal{D}}(\rho^*, t), \text{ and all } \mu \in T\Sigma^\omega, \\ \bot & \text{if } \rho \cdot_{\max(t,\tau(\rho))} \mu \notin L \text{ for all } \rho \in \widehat{GT}_{\mathcal{D}}(\rho^*, t), \text{ and all } \mu \in T\Sigma^\omega, \\ ? & \text{otherwise.} \end{cases}$$

$\widehat{\mathcal{V}}_{\mathcal{D}}(L)(\rho^*, t)$ is undefined when $t < \tau(\rho^*)$.

Similarly to the corresponding result for monitoring (see Lemma 6.6), one can prove that our testing verdict function is monotone in the delays: decreasing the uncertainty about the possible delays preserves conclusive verdicts.

Lemma 6.6. *Let $L \subseteq T\Sigma^\omega$, $\rho^* \in T\Sigma^*$, let $\mathcal{D} \subseteq \mathcal{D}'$ be delay sets, let ρ^* be a $\mathcal{D}$-observation, and let $t \geq \tau(\rho^*)$. Then, $\widehat{\mathcal{V}}_{\mathcal{D}'}(L)(\rho^*, t) = \top$ implies $\widehat{\mathcal{V}}_{\mathcal{D}}(L)(\rho^*, t) = \top$ and $\widehat{\mathcal{V}}_{\mathcal{D}'}(L)(\rho^*, t) = \bot$ implies $\widehat{\mathcal{V}}_{\mathcal{D}}(L)(\rho^*, t) = \bot$.*

Also, as for monitoring, we can refine the testing verdict function from Definition 6.5 to provide information about the delay parameters $(\delta_I, \varepsilon_I, \delta_O, \varepsilon_O)$ that can explain an observation. Given $L \subseteq T\Sigma^\omega$, a finite timed word $\rho^* \in T\Sigma^*$, and $t \geq \tau(\rho^*)$, the set of delays $\widehat{\Delta}(L, \rho^*, t)$ that are consistent with the observation ρ^* at t is defined as

$$\widehat{\Delta}(L, \rho^*, t) = \{(\delta_I, \varepsilon_I, \delta_O, \varepsilon_O) \mid \exists \rho \in \widehat{GT}_{\delta_I,\varepsilon_I,\delta_O,\varepsilon_O}(\rho^*, t)\, \exists \mu \in T\Sigma^\omega \text{ s.t. } \rho \cdot_{\max(t,\tau(\rho))} \mu \in L\}.$$

We denote by $\widehat{\Delta}_{\mathcal{D}}(L, \rho^*, t)$ the set $\widehat{\Delta}(L, \rho^*, t) \cap \mathcal{D}$.

Conclusive testing verdicts can again be characterized via the sets of consistent delays, which can be proven by the same argument as for the special case of monitoring (see Lemma 3.11).

Lemma 6.7. *Given $L \subseteq T\Sigma^\omega$, a set $\mathcal{D}$ of delays, a $\mathcal{D}$-observation $\rho^* \in T\Sigma^*$, and $t \geq \tau(\rho^*)$, we have*
(1) $\widehat{\Delta}_{\mathcal{D}}(L, \rho^*, t) = \emptyset$ *if and only if* $\widehat{\mathcal{V}}_{\mathcal{D}}(L)(\rho^*, t) = \bot$, *and*
(2) $\widehat{\Delta}_{\mathcal{D}}(\overline{L}, \rho^*, t) = \emptyset$ *if and only if* $\widehat{\mathcal{V}}_{\mathcal{D}}(L)(\rho^*, t) = \top$.

Again, even in the case when both delay-sets are nonempty (i.e., the verdict is *?*), we can still provide useful information in terms of the sets $\Delta(L, \rho^*, t)$ and $\Delta(\overline{L}, \rho^*, t)$ of consistent delays. They are non-increasing during observations: By extending observations, we (potentially) reduce the set of consistent delays. This result is again proven along the lines of the proof of the similar result for monitoring (Lemma 3.12).

Lemma 6.8. *Let $(\rho_1^*, t_1) \sqsubseteq (\rho_2^*, t_2)$ for finite timed words ρ_1^* and ρ_2^* with $t_1 \geq \tau(\rho_1^*)$ and $t_2 \geq \tau(\rho_2^*)$ and $t_2 \geq t_1$. Then, $\widehat{\Delta}(L, \rho_1^*, t_1) \supseteq \widehat{\Delta}(L, \rho_2^*, t_2)$.*

Now, we show how to compute $\widehat{\mathcal{V}}_{\mathcal{D}}(L)$ following the blueprint developed for monitoring under delay: Given ρ^* and t, compute the reach-set for $\widehat{GT}_{\mathcal{D}}(\rho^*, t)$ in the two TBA for L and the complement of L and then intersect with the respective sets of nonempty language states. Note that we cannot work with EL ground-truths here, as we have done in the case of monitoring, since we need to allow for unobserved outputs that were sent before the last input from the observation has arrived at the system. In Figure 6, consider the three outputs in the interval $[t - (\delta_O + \varepsilon_O), t]$: As the input at time $\tau_{I,3}$ needs to be part of the ground-truth, we need to allow these outputs as well.

Hence, we need to update our definition of reach-set to consider arbitrary ground-truths, not just EL ones: Given a TBA $\mathcal{A}$, a set $\mathcal{D}$ of delays, a finite observed timed word $\rho^* \in T\Sigma^*$, and $t \geq \tau(\rho^*)$, we define

$$\widehat{\mathcal{R}}^{\mathcal{D}}_{\mathcal{A}}(\rho^*, t) = \{(q, v + \max(0, (t - \tau(\rho)))) \mid (q_0, v_0) \xrightarrow{\rho}_{\mathcal{A}} (q, v) \text{ where } (q_0, v_0) \text{ with } q_0 \in Q_0,\ v_0(x) = 0 \text{ for all } x \in C, \text{ and } \rho \in \widehat{GT}_{\mathcal{D}}(\rho^*, t)\}.$$

We call this the reach-set of ρ^* in $\mathcal{A}$ at t w.r.t. $\mathcal{D}$. Using this, we can define the automata-based verdict function.

Definition 6.9. Given a TBA $\mathcal{A}$, a complement automaton $\overline{\mathcal{A}}$ (i.e., with $L(\overline{\mathcal{A}}) = T\Sigma^\omega \setminus L(\mathcal{A})$), a set $\mathcal{D}$ of delays, a $\mathcal{D}$-observation $\rho^* \in T\Sigma^*$, and $t \geq \tau(\rho)$, $\widehat{\mathcal{M}}_{\mathcal{D}} : \mathbb{A} \times \mathbb{A} \to T\Sigma^* \times \mathbb{R}_{\geq 0} \to \mathbb{B}_3$ computes the verdict

$$\widehat{\mathcal{M}}_{\mathcal{D}}(\mathcal{A}, \overline{\mathcal{A}})(\rho^*, t) = \begin{cases} \top & \text{if } \widehat{\mathcal{R}}^{\mathcal{D}}_{\overline{\mathcal{A}}}(\rho^*, t) \cap S^{ne}_{\overline{\mathcal{A}}} = \emptyset, \\ \bot & \text{if } \widehat{\mathcal{R}}^{\mathcal{D}}_{\mathcal{A}}(\rho^*, t) \cap S^{ne}_{\mathcal{A}} = \emptyset, \\ ? & \text{otherwise.} \end{cases}$$

$\widehat{\mathcal{M}}_{\mathcal{D}}(\mathcal{A}, \overline{\mathcal{A}})(\rho^*, t)$ is undefined if $t < \tau(\rho)$.

Using arguments similar to those to prove the analogous result for monitoring under delay (see Theorem 4.7), one can show that this automata-based definition of testing is equal to the verdict function defined above.

Theorem 6.10. *$\widehat{\mathcal{M}}_{\mathcal{D}}(\mathcal{A}, \overline{\mathcal{A}})(\rho^*, t) = \widehat{\mathcal{V}}_{\mathcal{D}}(L(\mathcal{A}))(\rho^*, t)$ for all sets $\mathcal{D}$ of delays, all TBA $\mathcal{A}$ (and complement automata $\overline{\mathcal{A}}$), all $\mathcal{D}$-observations ρ^*, and all $t \geq \tau(\rho^*)$.*

This result, while theoretically appealing, is unfortunately hard to implement efficiently using our zone-based approach.

Example 6.11. Recall that the delay $(\delta_I, \varepsilon_I, \delta_O, \varepsilon_O) = (0, 2, 0, 0)$ expresses that the input channel has latency zero and jitter two while the output channel has latency and jitter zero. Fix some $n > 0$. Now, consider a $(0, 2, 0, 0)$-observation ρ^* with n inputs in the interval $[0, 1]$ and n outputs in the interval $[1, 2]$. The inputs arrive at the system in the interval $[0, 3]$ and all outputs are observed at the same time as they have been sent, i.e., also in the interval $[1, 2]$.

In the following, we are focusing only on ground-truths where all inputs arrive in the interval $[1, 2]$ as well. Each input can arrive at any time in the interval $[1, 2]$, as long as the order of the inputs is preserved. Thus, the n outputs induce $n + 1$ buckets in which the n inputs can be placed into in the ground-truth. There are $\binom{2n}{n}$ ways to do so, a quantity that grows exponentially in n. Hence, even when abstracting away exact time points (this is what our zone-based construction does), we still need to account for all these possible orderings (which we would need to handle explicitly), leading to a combinatorial explosion.

Hence, in the following, we consider a setting that does not exhibit this combinatorial explosion while still being expressive enough to express properties from the use case studied in Section 7. Intuitively, we disallow the overtaking of inputs and outputs by considering only words in which inputs and outputs strictly alternate. Said differently, after each input, we need to wait for a corresponding output before the next input can be sent, and vice versa.

Definition 6.12. Let Σ be the disjoint union of Σ_I and Σ_O. A word $(\sigma_1, \tau_1), (\sigma_2, \tau_2), \ldots$ is IO-alternating if $\sigma_j \in \Sigma_I$ for all odd j and $\sigma_j \in \Sigma_O$ for all even j (note that the word needs to start with an input). Given a language $L \subseteq T\Sigma^\omega$, let $IO(L)$ be the set of IO-alternating words in L.

Remark 6.13.

(1) Given a *TBA* $\mathcal{A}$, one can effectively test whether it accepts only IO-alternating words (i.e., whether $\mathrm{IO}(L(\mathcal{A})) = L(\mathcal{A})$) by testing the intersection of $L(\mathcal{A})$ and the language of words with two consecutive inputs or two consecutive outputs for emptiness. As TBA are closed under intersection and emptiness is decidable [1], this can indeed be done effectively.
(2) Given a TBA $\mathcal{A}$, one can effectively construct a TBA that accepts $\mathrm{IO}(L(\mathcal{A}))$ by taking the product of $\mathcal{A}$ and a TBA that accepts the language of all IO-alternating words to accept the intersection of both languages.

In the following, we restrict ourselves only to words that are IO-alternating. This requires to further restrict the observations that can be made: An observation $\rho^* = (\sigma_1, \tau_1) \cdots (\sigma_n, \tau_n)$ is an IO-observation (under $(\delta_I, \varepsilon_I, \delta_O, \varepsilon_O)$) if ρ^* is IO-alternating and $\tau_{j+1} - \tau_j \geq \delta_I + \delta_O$ for all odd j, i.e., there is enough time between an input sent and the next observed output to sent these events through the corresponding channels.

The following theorem shows that we can test MITL specifications w.r.t. IO-alternating words. Note that $\mathrm{IO}(L(\mathcal{A})$ and $\mathrm{IO}(L(\overline{\mathcal{A}}))$ partition the set of IO-alternating words, which implies that $\widehat{\mathcal{M}}_{\mathcal{D}}(\mathrm{IO}(L(\mathcal{A})), \mathrm{IO}(L(\overline{\mathcal{A}})))$ is well-defined. Furthermore, by considering these TBA, we restrict ourselves to extensions of ground-truths of the observation that are IO-alternating, all other words are ignored.

Theorem 6.14. *The restriction of* $\widehat{\mathcal{M}}_{\mathcal{D}}(\mathrm{IO}(L(\mathcal{A})), \mathrm{IO}(L(\overline{\mathcal{A}})))$ *to IO-observations is effectively computable for specifications given by TBA* $\mathcal{A}$ *and* $\overline{\mathcal{A}}$*, and* $\mathcal{D} = \{(\delta_I, \varepsilon_I, \delta_O, \varepsilon_O) \mid \delta_I \in [\ell_I, u_I]$ *and* $\delta_O \in [\ell_O, u_O]\}$ *for given* $\ell_I, u_I, \ell_O, u_O, \varepsilon_I, \varepsilon_O \in \mathbb{Q}_{\geq 0}$.

Proof. We generalize the construction presented above for monitoring to testing, i.e., we show how to compute reach-sets and then intersect them with the non-empty language states. Due to Theorem 6.10, this suffices to compute $\widehat{\mathcal{M}}_{\mathcal{D}}(\mathrm{IO}(L(\mathcal{A})), \mathrm{IO}(L(\overline{\mathcal{A}})))$.

Instead of just adding the clock *etime* as in the case of monitoring under delay, we now add two clocks $etime_I$, $etime_O$, and change the initial zones to include the latency bounds as the differences between the clocks $etime_O$, $etime_I$, and *time*. The idea here is that $etime_I \leq time \leq etime_O$ and that $time - etime_I$ reflects the input delay from the test harness to the SUT that applies to test stimuli, while $etime_O - time$ corresponds to the output delay affecting responses from the SUT. Given a symbolic state (q, Z) where Z is extended with clocks *time*, $etime_I$, and $etime_O$, and a pair $(\square, \tau) \in \{I, O\} \times \mathbb{Q}_{\geq 0}$ we define the following zone operation:

$$(q, Z) \wedge (\square, \tau) = \begin{cases} (q, Z \wedge etime_I \in [\tau, \tau + \varepsilon_I]) & \text{if } \square = I, \\ (q, Z \wedge etime_O \in [\tau - \varepsilon_O, \tau]) & \text{if } \square = O. \end{cases}$$

Given a TBA $\mathcal{A}$ extended with clocks *time*, $etime_I$ and $etime_O$, and an observation $(\sigma, \tau) \in \Sigma \times \mathbb{Q}_{\geq 0}$, the set of possible successors of a symbolic state (q, Z) is

$$\begin{aligned} Succ_{IO}((q, Z), (\sigma, \tau)) &= \{(q', Z') \wedge (direction(\sigma), \tau) \mid (q', Z') \in post((q, Z), \sigma)\}, \text{ where} \\ direction(\sigma) &= \begin{cases} I & \text{if } \sigma \in \Sigma_I, \\ O & \text{if } \sigma \in \Sigma_O, \end{cases} \end{aligned}$$

and

$$Succ_{IO}(S, (\sigma, \tau)) = \bigcup_{(q,Z) \in S} succ_{IO}((q, Z), (\sigma, \tau))$$

collects the successors of a set of symbolic states S. In order to use $Succ_{IO}$ to compute the reach-set, we need a different initial zone than Z_0. We define Z_0^{IO} to extend Z_0 with *time*, $etime_I$, and $etime_O$ and set the bounds on the differences of these clocks to be the input/output delay bounds. Given a TBA $\mathcal{A} = (Q, Q_0, \Sigma, C, \Delta, \mathcal{F})$ and latency bounds ℓ_I, u_I, ℓ_O, u_O then we have that:[3]

$$\begin{aligned} Z_0^{IO} \equiv \overbrace{etime_O - time \leq u_O \wedge time - etime_O \leq -\ell_O}^{etime_O - time \in [\ell_O, u_O]} \wedge \\ \forall x \in C \cup \{time\} : x = 0 \wedge \\ \underbrace{etime_I - time \leq -\ell_I \wedge time - etime_I \leq u_I}_{time - etime_I \in [\ell_I, u_I]}. \end{aligned}$$

We can now compute the reach-set after a sequence of observations. Given a TBA $\mathcal{A}$, a finite timed word $\rho = (\sigma_1, \tau_1), \ldots, (\sigma_n, \tau_n) \in T\Sigma^*$, a time point $t \geq \tau(\rho)$, and a set of delays $\mathcal{D} = \{(\delta_I, \varepsilon_I, \delta_O, \varepsilon_O) \mid \delta_I \in [\ell_I, u_I]$ and $\delta_O \in [\ell_O, u_O]\}$ for given $\ell_I, u_I, \ell_O, u_O, \varepsilon_I, \varepsilon_O \in \mathbb{Q}_{\geq 0}$, the reach-set can be computed as $\widehat{\mathcal{R}}_{\mathcal{A}}^{\mathcal{D}}(\rho^*, t) = S_{n+1}$ where $S_0 = \{(q_0, Z_0^{IO}) \mid q_0 \in Q_0\}$ and $S_i = Succ_{IO}(S_{i-1}, (\sigma_i, \tau_i))$ for $i \in [1, n]$, and

$$S_{n+1} = \begin{cases} S_n \wedge (I, t) & \text{if } direction(\sigma_n) = O \\ S_n \wedge (O, t) & \text{if } direction(\sigma_n) = I \text{ and } t > l_I + u_O + \varepsilon_O \\ S_n & \text{otherwise} \end{cases}$$

The computation of the set of non-empty language states requires us to enforce IO-alternation by taking the intersection with an automaton $\mathcal{A}_{IO}$ that accepts all IO-alternating words such that $\mathrm{IO}(L(\mathcal{A})) = L(\mathcal{A} \otimes \mathcal{A}_{IO})$ as described in Remark 6.13. Remark 2. This allows us to implement the testing function by computing the reach-sets and intersecting them with the set of non-empty language states. □

[3]Technically speaking, the constraints defining Z_0^{IO} require negative clock values for $etime_I$. This can be avoided by a shift in the bounds between *time* and $etime_I$, or by letting time pass.

```
1  Input: @173 a
2
3  Verdict: INCONCLUSIVE
4  Positive:
5  Consistent latencies: {[71,100]}
6  Jitter bound: 2
7  Negative:
8  Consistent latencies: {[0,100]}
9  Jitter bound: 2
10
11 Input: @271 b
12
13 Verdict: INCONCLUSIVE
14 Positive:
15 Consistent latencies: {[71,75)}
16 Jitter bound: 2
17 Negative:
18 Consistent latencies: {[0,100]}
19 Jitter bound: 2
```

Listing 1. Demonstration of MoniTAal over Example 5.4.

7 Implementation

In this section we demonstrate our tool implementation of the monitoring and testing procedures described in this article. We run experiments to demonstrate the efficiency and illustrate the evaluation of consistent latencies.

The tool MoniTAal[4] implements monitoring and testing under delay as described in this article. This includes the DBM data structure to handle clock zones, parsing property automata modeled in Uppaal, computing the set of nonempty language states, computing the reach-sets in an online fashion over an observed word based on latency and jitter bounds in $[0, \infty[$, providing verdicts $\top$, $\bot$ or *?*, and providing bounds on the latency values that are consistent with $\top$ and $\bot$ for both inputs and outputs.

MoniTAal monitors a property by utilizing an automaton for the property and its complement. A series of observations can then be provided to compute verdicts. Bounds on latency and jitter can be given when monitoring under delay, where the valid latency values are computed for each verdict.

We give a short demonstration going through Example 5.4 using MoniTAal to monitor the property $F_{[0,10]}a \wedge G_{[0,20]}\neg b$ over the timed word $(a, 17.3), (b, 27.5)$, where the latency is between 0 and 10, and the jitter bound is 0.2. Listing 1 shows the output of the tool on lines 3–9 and 13–19 after each observation $(a, 173)$ on line 1 and $(b, 271)$ on line 11. Note that we multiply all timing values by 10 in order to use integer, rather than rational, time points. After the first observation, the verdict is *?* (inconclusive) (line 3). Furthermore, we see that the consistent latencies for the $\top$ (positive) verdict are now tightened to $[71, 100]$ (line 5). The consistent latencies for the $\bot$ (negative) verdict are still in $[0, 100]$ (line 8). The second observation is $(b, 271)$ on line 11, after which the verdict is still *?* (inconclusive) (line 13), but the $\top$ consistent latencies are now tightened further to be within $[71, 75)$ (line 15), while the $\bot$ consistent latencies are still in $[0, 100]$ (line 18).

For realistic experiments, we took traces and properties from the gear controller model in [28]. The model, along with its formal requirements, was created by the company Mecel.

[4] https://github.com/DEIS-Tools/MoniTAal

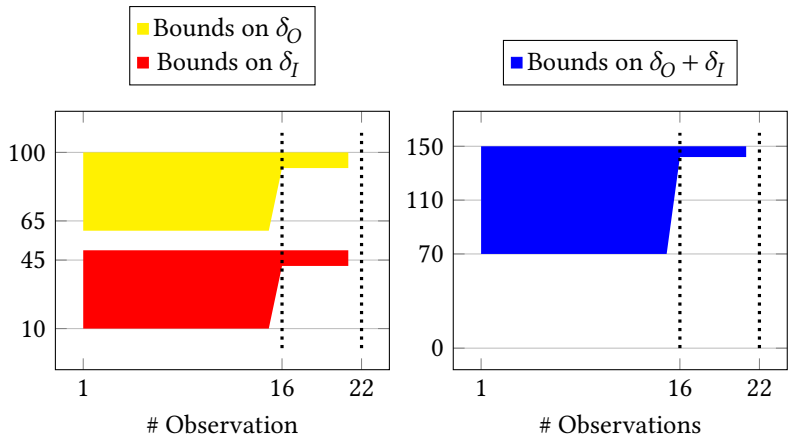

Fig. 7. Test run with 22 observations ending in a $\bot$ verdict where $\delta_I \in [10, 50], \delta_O \in [60, 100], \varepsilon = 10$ and the actual input latency is 65 and the actual output latency is 45. Errors occurred at the 16. and 22. observation. The maximal size of reach-sets is five states.

To demonstrate the tightening of bounds for the latency parameters, we monitor the response property

$$\varphi_{gear} = G_{[0,\infty]}(ReqNewGear \rightarrow F_{[150,1205]}NewGear)$$

that requires the gear controller to change gear within 150 ms to 1205 ms after a shift request. In this experiment the event *ReqNewGear* is an input from the test harness and *NewGear* is an output from the SUT, i.e., we are concerned with active testing where the inputs arrive after some delay at the system to be monitored and the outputs from the system are only observed after some delay.

To verify φ_{gear}, we start the monitor with the known bounds on the unknown latency and the jitter. We apply the monitor to original traces of the gearbox model, which have their timestamps modified by an assigned input-/output latency (which is unknown to the monitor) plus a random jitter value. The traces have a chance of providing an error, which means that the system changes the gear before 150 ms or after 1205 ms. Because of the unknown exact delay, the monitor is not guaranteed to catch the error and give a $\bot$ verdict immediately; however, it will tighten the latency bounds and may provide a verdict after further observations.

First we monitor φ_{gear} with latency parameters $\delta_I \in [10, 50]$ and $\delta_O \in [60, 100]$ and a jitter bound of 10. The actual input and output latency is 45 and 65. The trace provided had an error at observation 16 and 22, after which the monitor gave a $\bot$ verdict.

The changing bounds of $\top$-consistent latencies are illustrated in Figure 7 where the left side shows the upper and lower bounds on the input (red, at the top) and output (yellow, at the bottom) latency and the right side shows the bounds on the latencies added together (blue). When a gear is changed outside the bounds of the property, this is an error at the system, and is marked by a dashed vertical line at the index of that observation. As the first error occurs (at time 16), all lower bounds on latencies are tightened significantly. The actual latency of the output (which is 65) here already becomes inconsistent with the $\top$ verdict, but as the actual latency is unknown, this remains undetectable and the verdict stays inconclusive at this point of time. Yet another error occurs at the 22nd event, at which point the set of consistent delays for the $\top$ verdict becomes empty and the $\bot$ verdict is given.

We monitor φ_{gear} again, now with different latency bounds $\delta_I \in [0, 90]$ and $\delta_O \in [100, 200]$ and the actual input-/output latency is 60 and 120.

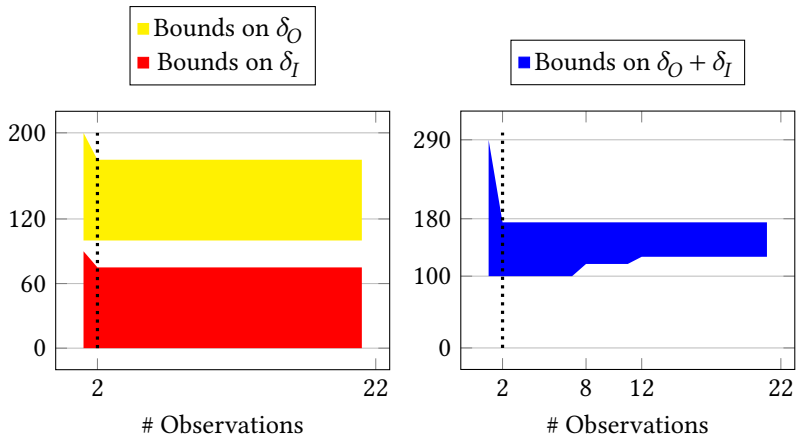

Fig. 8. Test run with 22 observations ending in a $\perp$ verdict where $\delta_I \in [0, 90], \delta_O \in [100, 200], \varepsilon = 10$ and the actual input latency is 60 and the actual output latency is 120. A single error occurred at observation # 2. The maximal size of reach-sets is five states.

Figure 8 shows the changing latency bounds where a single error occurs in the first response (observation 2), after which the upper bounds of the input-/output latency lower slightly, while the upper bound on the combined latency is tightened significantly. It drops slightly below the actual combined latency, which is 180, but this remains undetectable as the actual latency is unknown. Over the course of subsequent observations, the lower bound on the combined latency increases repeatedly, even though no further erroneous behavior is observed. After 22 observations, the set of consistent combined latencies finally becomes empty, and consequently the verdict $\perp$ is given. Note that this conclusive verdict is a consequence of a complex temporal matching process that tries to align the correct observations between time instants 3 and 22 with the possible observation delays and only thereby conclusively detects the single error that occurred at the very beginning of the observation.

To examine the performance of our monitoring algorithms, we again monitor the property φ_{gear}, but in this case over a trace that does not feature any errors such that monitoring will not terminate early with a conclusive verdict. As the monitor does not terminate, we can evaluate the monitor performance over arbitrarily long traces ranging from 1,000 to 10,000 events. In these experiments, we compare three scenarios: *classic* is regular monitoring with no delays, *delay* is where every observation is an output, and *testing* is where the request is a delayed input and the response is a delayed output. In the delay and testing cases the delay parameters are $\delta_I, \delta_O \in [0, 100]$ and $\varepsilon_I, \varepsilon_O = 10$.

The maximal response time and number of symbolic states in the reach-set of the three scenarios are plotted in Table 1. Response time is the time it takes from observing an event to providing a verdict. The size of the reach-set is the number of symbolic states stored in the representation of the reach-set. In the implementation, the reach-set is stored as a set of symbolic states which might contain redundant information. To minimize the size we check for zone inclusion, such that a symbolic state (q, Z) is only stored if it is not included in another stored state (q, Z') with $Z \subseteq Z'$. We also employ the inactive clock abstraction from [14].

The size of the reach-set obviously affects the response time negatively, as is evident from the differences between the classic, delay, and testing cases. However, all response times remain below 300 μs, demonstrating that the implementation can sustain real-time monitoring of the response property over an arbitrary number of observations and in all scenarios, even in the complex testing

Table 1. Results for Monitoring φ_{gear} under Delay and Test Over Traces with Varying Length

# Observ.	Max. response time (μs)			Max. # Symbolic States		
	Classic	Delay	Testing	Classic	Delay	Testing
1000	66	82	217	2	3	11
2000	42	76	153	2	3	11
3000	63	83	233	2	3	11
4000	34	84	230	2	3	11
5000	63	68	216	2	3	11
6000	67	101	216	2	3	11
7000	66	64	228	2	3	11
8000	59	73	215	2	3	11
9000	77	70	256	2	3	11
10000	47	77	216	2	3	11

scenario with both input and output delays. It is also interesting to note that in the testing case, the maximal number of states increased significantly. This is due to the fact that we add two clocks that are continuously changed but never reset. The property itself only requires a single clock, so the total number of clocks in the classic, delay, and testing scenarios are 1, 3, and 4. The stored symbolic states then have more constraints that can be different from each other. If resulting zones have slightly shifted bounds between $etime_I$ and $etime_O$, it might be beneficial to check if any zones can be merged, in order to make the reach-set representation more memory-efficient.

8 Conclusion

We have introduced zone-based algorithms realizing optimal (in the sense of being anticipating [7] as well as conclusive under uncertainty [16]) online operational monitoring and testing of embedded real-time systems when the communication between the monitor (or testing harness, respectively) and the system is subject to unknown (up to bounds) delay. This situation is rather typical in practice as observations are mediated by sensors, may involve conversion between analog and digital, or pass communication networks and consequently are indirect in general, leading to delays and inexact time-stamping. Our constructions thus fill a gap in the pre-existing theories for monitoring and testing hard real-time systems, which tend to assume full and exact temporal observability by immediate coupling or, equivalently, perfect synchrony between systems and their monitors or test harnesses.

We assume no knowledge of timing information of observations, except knowing when they are received. However, one could imagine a scenario where some output observations include timing information from the system. This information would give us the exact delay (latency + jitter) of a specific observation and, while not providing an exact synthesis, could be used to refine the latency parameter. Thus the parameter would not necessarily be fully determined, and is still needed in the gaps between future observations and for observations without timing information.

A notable point of our construction is that it applies a reduction to simple timed automata and is purely zone-based despite the unknown communication delay being a timing parameter. The construction thus not only avoids the complexities of property analysis for parameterized timed automata [4], but also provides an instance of monitoring and testing under uncertainty where the underlying arithmetic constraint systems remain of fixed dimensionality (namely the number of clocks in the property automata plus two for monitoring) despite their history dependence. This is in stark contrast to direct constraint encodings growing linearly over history length as in [16].

In further research, we study the question of monitorability [7]: some properties will never give definitive verdicts (e.g., "infinitely often a") and are therefore not useful for monitoring. We conjecture that our zone-based approach can be exploited to decide monitorability of real-time properties.

References

[1] Rajeev Alur and David L. Dill. 1994. A theory of timed automata. *Theor. Comput. Sci.* 126, 2 (1994), 183–235. DOI: https://doi.org/10.1016/0304-3975(94)90010-8

[2] Rajeev Alur, Tomás Feder, and Thomas A. Henzinger. 1996. The benefits of relaxing punctuality. *J. ACM* 43, 1 (1996), 116–146. DOI: https://doi.org/10.1145/227595.227602

[3] Rajeev Alur, Thomas A. Henzinger, and Moshe Y. Vardi. 1993. Parametric real-time reasoning. In *Proceedings of the 25th Annual ACM Symposium on Theory of Computing*, S. Rao Kosaraju, David S. Johnson, and Alok Aggarwal (Eds.). ACM, 592–601. DOI: https://doi.org/10.1145/167088.167242

[4] Étienne André, Didier Lime, and Olivier H. Roux. 2022. Reachability and liveness in parametric timed automata. *Log. Methods Comput. Sci.* 18, 1 (2022), 31:1–31:41. DOI: https://doi.org/10.46298/LMCS-18(1:31)2022

[5] Kevin Baldor and Jianwei Niu. 2013. Monitoring dense-time, continuous-semantics, metric temporal logic. In *Runtime Verification*. Springer Berlin, 245–259. DOI: https://doi.org/10.1007/978-3-642-35632-2_24

[6] David Basin, Felix Klaedtke, and Eugen Zălinescu. 2011. Algorithms for monitoring real-time properties. In *Proceedings of the 2nd International Conference on Runtime Verification, RV 2011, Revised Selected Papers*, Sarfraz Khurshid and Koushik Sen (Eds.). LNCS, Vol. 7186, Springer, 260–275. DOI: https://doi.org/10.1007/978-3-642-29860-8_20

[7] Andreas Bauer, Martin Leucker, and Christian Schallhart. 2006. Monitoring of real-time properties. In *FSTTCS 2006: Foundations of Software Technology and Theoretical Computer Science*, S. Arun-Kumar and Naveen Garg (Eds.). Springer, Berlin, 260–272. DOI: https://doi.org/10.1007/11944836_25

[8] Johan Bengtsson and Wang Yi. 2003. Timed automata: Semantics, algorithms and tools. In *Lectures on Concurrency and Petri Nets, Advances in Petri Nets* . LNCS, Vol. 3098, Springer, 87–124. DOI: https://doi.org/10.1007/978-3-540-27755-2_3

[9] Thomas Brihaye, Gilles Geeraerts, Hsi-Ming Ho, and Benjamin Monmege. 2017. MightyL: A compositional translation from MITL to timed automata. In *Computer Aided Verification*. Springer, 421–440. DOI: https://doi.org/10.1007/978-3-319-63387-9_21

[10] Peter E. Bulychev, Alexandre David, Kim G. Larsen, Axel Legay, Guangyuan Li, and Danny Bøgsted Poulsen. 2012. Rewrite-based statistical model checking of WMTL. In *Proceedings of the 3rd International Conference on Runtime Verification, RV 2012, Revised Selected Papers*, Shaz Qadeer and Serdar Tasiran (Eds.). LNCS, Vol. 7687, Springer, 260–275. DOI: https://doi.org/10.1007/978-3-642-35632-2_25

[11] Alessandro Cimatti, Thomas Møller Grosen, Kim G. Larsen, Stefano Tonetta, and Martin Zimmermann. 2024. Exploiting assumptions for effective monitoring of real-time properties under partial observability. In *SEFM 2024*, Alexandre Madeira and Alexander Knapp (Eds.). LNCS, Vol. 15280, Springer, 70–88. DOI: https://doi.org/10.1007/978-3-031-77382-2_5

[12] Werner Damm, Günter Ehmen, Kim Grüttner, Philipp Ittershagen, Björn Koopmann, Frank Poppen, and Ingo Stierand. 2019. Multi-layer time coherency in the development of ADAS/AD systems: Design approach and tooling. In *Proceedings of the Workshop on Design Automation for CPS and IoT (DESTION '19)*. ACM, New York, NY, USA, 20–30. DOI: https://doi.org/10.1145/3313151.3313167

[13] Alexandre David, Kim Guldstrand Larsen, Shuhao Li, Marius Mikucionis, and Brian Nielsen. 2010. Testing real-time systems under uncertainty. In *Proceedings of the 9th International Symposium on Formal Methods for Components and Objects, FMCO 2010. Revised Papers*, Bernhard K. Aichernig, Frank S. de Boer, and Marcello M. Bonsangue (Eds.). LNCS, Vol. 6957, Springer, 352–371. DOI: https://doi.org/10.1007/978-3-642-25271-6_19

[14] Conrado Daws and Sergio Yovine. 1996. Reducing the number of clock variables of timed automata. In *Proceedings of the 17th IEEE Real-Time Systems Symposium (RTSS 1996)*. IEEE Computer Society, 73–81. DOI: https://doi.org/10.1109/REAL.1996.563702

[15] Alexandre Donzé, Thomas Ferrère, and Oded Maler. 2013. Efficient robust monitoring for STL. In *Proceedings of the 25th International Conference on Computer Aided Verification, CAV 2013. Proceedings*, Natasha Sharygina and Helmut Veith (Eds.). LNCS, Vol. 8044, Springer, 264–279. DOI: https://doi.org/10.1007/978-3-642-39799-8_19

[16] Bernd Finkbeiner, Martin Fränzle, Florian Kohn, and Paul Kröger. 2022. A truly robust signal temporal logic: Monitoring safety properties of interacting cyber-physical systems under uncertain observation. *Algorithms* 15, 4 (2022), 126. DOI: https://doi.org/10.3390/a15040126

[17] Bernd Finkbeiner, Martin Fränzle, Florian Kohn, and Paul Kröger. 2025. Stream-based monitoring under measurement noise. In *Proceedings of the 24th International Conference on Runtime Verification, RV 2024*, Erika Ábrahám and Houssam Abbas (Eds.). LNCS, Vol. 15191, Springer Nature Switzerland, Cham, 22–39. DOI: https://doi.org/10.1007/978-3-031-74234-7_2

[18] Martin Fränzle, Thomas Møller Grosen, Kim G. Larsen, and Martin Zimmermann. 2024. Monitoring real-time systems under parametric delay. In *IFM 2024*, Nikolai Kosmatov and Laura Kovács (Eds.). LNCS, Vol. 15234, Springer, 194–213. DOI: https://doi.org/10.1007/978-3-031-76554-4_11

[19] Bence Graics, Milán Mondok, Vince Molnár, and István Majzik. 2025. Model-based testing of asynchronously communicating distributed controllers using validated mappings to formal representations. *Sci. Comput. Program.* 242 (2025), 103265. DOI: https://doi.org/10.1016/J.SCICO.2025.103265

[20] Thomas Møller Grosen, Sean Kauffman, Kim Guldstrand Larsen, and Martin Zimmermann. 2022. Monitoring timed properties (revisited). In *Proceedings of the 20th International Conference on Formal Modeling and Analysis of Timed Systems, FORMATS 2022*, Sergiy Bogomolov and David Parker (Eds.). LNCS, Vol. 13465, Springer, 43–62. DOI: https://doi.org/10.1007/978-3-031-15839-1_3

[21] Hsi-Ming Ho, Joël Ouaknine, and James Worrell. 2014. Online monitoring of metric temporal logic. In *Runtime Verification*. Springer, 178–192. DOI: https://doi.org/10.1007/978-3-319-11164-3_15

[22] Wen-ling Huang, Niklas Krafczyk, and Jan Peleska. 2024. Exhaustive property oriented model-based testing with symbolic finite state machines. *Sci. Comput. Program.* 231 (2024), 103005. DOI: https://doi.org/10.1016/J.SCICO.2023.103005

[23] Hannes Kallwies, Martin Leucker, and César Sánchez. 2022. Symbolic runtime verification for monitoring under uncertainties and assumptions. In *Proceedings of the 20th International Symposium on Automated Technology for Verification and Analysis, ATVA 2022*, Ahmed Bouajjani, Lukás Holík, and Zhilin Wu (Eds.). LNCS, Vol. 13505, Springer, 117–134. DOI: https://doi.org/10.1007/978-3-031-19992-9_8

[24] Maximilian A. Köhl and Holger Hermanns. 2023. Model-based diagnosis of real-time systems: Robustness against varying latency, clock drift, and out-of-order observations. *ACM Trans. Embed. Comput. Syst.* 22, 4 (2023), 68:1–68:48. DOI: https://doi.org/10.1145/3597209

[25] Kim Guldstrand Larsen, Paul Pettersson, and Wang Yi. 1995. Model-checking for real-time systems. In *FCT 1995*, Horst Reichel (Ed.). LNCS, Vol. 965, Springer, 62–88. DOI: https://doi.org/10.1007/3-540-60249-6_41

[26] Kim Guldstrand Larsen, Paul Pettersson, and Wang Yi. 1997. UPPAAL in a nutshell. *Int. J. Softw. Tools Technol. Transf.* 1, 1–2 (1997), 134–152. DOI: https://doi.org/10.1007/S100090050010

[27] Magnus Lindahl, Paul Pettersson, and Wang Yi. 1998. Formal design and analysis of a gear controller. In *Tools and Algorithms for Construction and Analysis of Systems (TACAS)*, Bernhard Steffen (Ed.). LNCS, Vol. 1384, Springer, 281–297. DOI: https://doi.org/10.1007/BFb0054178

[28] Magnus Lindahl, Paul Pettersson, and Wang Yi. 2001. Formal design and analysis of a gearbox controller. *Springer International Journal of Software Tools for Technology Transfer (STTT)* 3, 3 (2001), 353–368.

[29] Oded Maler and Dejan Nickovic. 2004. Monitoring temporal properties of continuous signals. In *Formal Techniques, Modelling and Analysis of Timed and Fault-Tolerant Systems, Joint International Conferences on Formal Modelling and Analysis of Timed Systems, FORMATS 2004 and Formal Techniques in Real-Time and Fault-Tolerant Systems, FTRTFT 2004*, Yassine Lakhnech and Sergio Yovine (Eds.). LNCS, Vol. 3253, Springer, 152–166. DOI: https://doi.org/10.1007/978-3-540-30206-3_12

[30] Dejan Ničković and Tomoya Yamaguchi. 2020. RTAMT: Online robustness monitors from STL. In *Automated Technology for Verification and Analysis*, Dang Van Hung and Oleg Sokolsky (Eds.). Springer International Publishing, Cham, 564–571.

[31] Caio R. D. Osório, Adrien Genic, and Sergio Costa. 2023. Introduction to typhoon HIL: Technology, functionalities, and applications. In *Real-Time Simulation and Hardware-in-the-Loop Testing Using Typhoon HIL*, Saurabh Mani Tripathi and Francisco M. Gonzalez-Longatt (Eds.). Springer Nature Singapore, Singapore, 1–28. DOI: https://doi.org/10.1007/978-981-99-0224-8_1

[32] Tom Sheldon. 2001. *McGraw-Hill's Encyclopedia of Networking and Telecommunications*. McGraw-Hill Professional.

[33] Tino Teige, Andreas Eggers, Karsten Scheibler, Matthias Stasch, Udo Brockmeyer, Hans Jürgen Holberg, and Tom Bienmüller. 2021. Two decades of formal methods in industrial products at BTC embedded systems. In *Proceedings of the 24th International Symposium on Formal Methods, FM 2021, Virtual Event*, Marieke Huisman, Corina S. Pasareanu, and Naijun Zhan (Eds.). LNCS, Vol. 13047, Springer, 725–729. DOI: https://doi.org/10.1007/978-3-030-90870-6_40

[34] Prasanna Thati and Grigore Rosu. 2004. Monitoring algorithms for metric temporal logic specifications. In *Proceedings of the 4th Workshop on Runtime Verification, RV@ETAPS 2004*, Klaus Havelund and Grigore Rosu (Eds.). Electronic Notes in Theoretical Computer Science, Vol. 113, Elsevier, 145–162. DOI: https://doi.org/10.1016/J.ENTCS.2004.01.029

[35] Dogan Ulus, Thomas Ferrère, Eugene Asarin, and Oded Maler. 2014. Timed pattern matching. In *Formal Modeling and Analysis of Timed Systems*. Springer, 222–236. DOI: https://doi.org/10.1007/978-3-319-10512-3_16

[36] Dogan Ulus, Thomas Ferrère, Eugene Asarin, and Oded Maler. 2016. Online timed pattern matching using derivatives. In *Tools and Algorithms for the Construction and Analysis of Systems*. Springer, 736–751. DOI: https://doi.org/10.1007/978-3-662-49674-9_47

[37] Ennio Visconti, Ezio Bartocci, Michele Loreti, and Laura Nenzi. 2021. Online monitoring of spatio-temporal properties for imprecise signals. In *MEMOCODE '21: Proceedings of the 19th ACM-IEEE International Conference on Formal Methods and Models for System Design, Virtual Event*, S. Arun-Kumar, Dominique Méry, Indranil Saha, and Lijun Zhang (Eds.). ACM, 78–88. DOI: https://doi.org/10.1145/3487212.3487344

[38] Masaki Waga, Étienne André, and Ichiro Hasuo. 2022. Model-bounded monitoring of hybrid systems. *ACM Trans. Cyber Phys. Syst.* 6, 4 (2022), 30:1–30:26. DOI: https://doi.org/10.1145/3529095

Received 14 February 2025; revised 12 June 2025; accepted 16 July 2025