Similarities between Timing Constraint Sets: Towards Interchangeable Constraint Models for Real-World Software Systems

Yue Yu and Shangping Ren
Department of Computer Science
Illinois Institute of Technology
Chicago, IL 60616
{yyu8, ren}@iit.edu

Abstract—Traditionally, given two timing constraint sets, their relationship is defined by their timed trace inclusions. This approach only gives a boolean answer to if one set of constraints is contained within the other. In this paper, we first introduce a quantitative measure to describe the closeness or the similarity between two timing constraint sets. We intend to study the satisfaction bounds of similar timing constraint sets by similar timed systems. Such bounds will help improve the predictability of real-time systems in real-world applications and provide guidance for self-tuning systems.

I. INTRODUCTION

Software for real-world systems inevitably has to operate in an unpredictable environment and interacts with physical machineries. Hence, for most of these software systems, it is difficult and unrealistic to design and implement them in such a way that can be guaranteed to behave precisely as specified due to the following facts:

- **System Complexity** The ever increasing complexities of software systems have made guarantees of exact system behavior impractically expensive, if not impossible. For example, as pointed out by Lee [1], advances in computer architecture and software have made it difficult or impossible to estimate or predict the execution time of software. Moreover, networking techniques introduce variability and stochastic behavior.

- **Operating Environment** The intrinsically unpredictable nature of the environments in which software systems operate determines that even though software operates precisely as designed, its interactions with the outer world may not be totally expected. For example, [2] shows that several aircraft accidents have been attributed to "mode confusion", where the software operated as designed but not as expected by the pilots.

- **Computational Intractability** From a theoretical point of view, achieving exactness in the verification of system properties is sometimes intractable. For example, [3] has shown that the satisfiability of a very simple class of real-time properties such as "every p-state is followed by a q-state precisely 5 time units later" turns out to be undecidable in a continuous model of time. On the other hand, several real-time logics are decidable under discrete approximations to the real time [4] or under interval timing constraints that prohibit infinite accuracy [5].

Therefore, basing our reasoning about systems’ timing properties and timing constraint satisfactions on precise information of real-world software systems is unpractical. Moreover, the traditional view of equivalence between timed systems and inclusion of timed trace set is hardly obtainable, neither sufficient. In fact, it is more practical and more accurate to allow impreciseness when modeling real-world systems.

Similarity metrics have been studied recently [6], [7], [8], [9]. In parallel to the researches on similarities between timed systems, our ongoing studies focus on the similarities between timing constraint sets and their impacts on constraint satisfactions. The constraint similarity theories can be further applied to our non-intrusive, event-based feedback loop control system for enhancing legacy systems with self-protection features.

II. SATISFACTIONS OF SIMILAR CONSTRAINT SETS BY SIMILAR TIMED SYSTEMS

A. Similarities between Timed Systems

Timed systems model the sequence of system events and timing information of those events. However, since timed system models are approximations of the real world, achieving exactness in these models are unrealistic [10], [11].

Huang et al. [7] investigate the real-time property preservation between two similar timed state sequences (execution traces of timed systems), and extend the results to timed systems (a timed system is described and modeled by a set of timed state sequences). More specifically, the authors define the distance metric over timed state sequences $d_{sup}$ and the weakening function over real-time properties $R^\mu(\mu \in \mathbb{R}^+)$, such that

- **Relaxation Property of $R^\mu$:** For real-time property $\varphi$, $R^\mu(\varphi)$ is weaker than $\varphi$; and the larger $\mu$ is, the weaker is the real-time property $R^\mu(\varphi)$.

- **Robustness Property of $d_{sup}$:** Given two timed state sequences $\bar{\tau}$ and $\bar{\tau}'$ such that $d_{sup}(\bar{\tau}, \bar{\tau}') \leq \epsilon$, if $\bar{\tau}$ satisfies formula $\varphi$, the real-time property $R^{2\epsilon}(\varphi)$ is preserved for $\bar{\tau}'$.
The authors extend these results to concurrent real-time systems (with interleaving semantics) in [8]. However, in both papers, they do not provide algorithms to compute distances between systems, relying on system execution to estimate the bound.

Henzinger et al. [6] define quantitative notions of timed similarity and bisimilarity which generalize timed similarity and bisimilarity relations [12] to metrics over timed systems. The authors show that both the timed computation tree logic (TCTL) [13] and the discounted computation tree logic (DCTL) [14] are robust under the bisimilarity metrics, i.e., states similar under the metric satisfy specifications with similar timing requirements. They also give algorithms to compute the similarity distance between two timed systems modeled as timed automata to within any given precision.

B. Similarities between Timing Constraint Sets

In parallel to the works on similarity concepts for timed state sequences and timed systems, we study similarities between linear timing constraint sets which can be used to express timing requirements for timed systems. Traditional ways of comparing timing constraint sets search for exactness. Consider the following problem of timed trace inclusion.

Example 1: A timed trace of a set of real-time constraints can be represented as a timed data stream\(^1\) [15]. The set of all timed data streams satisfying a given set of real-time constraints is often infinite. However, it can be represented as a convex polyhedron in the affine space \(\mathbb{R}^n\) where \(n\) is the number of constrained event types. For example, Fig. 1 shows the trace polyhedron of the constraint set (1).

\[
\begin{align*}
    t(e_1) - t(e_2) &\leq 6, & t(e_2) - t(e_1) &\leq 6, \\
    t(e_1) - t(e_3) &\leq 7, & t(e_3) - t(e_1) &\leq 3, \\
    t(e_2) - t(e_3) &\leq 9, & t(e_3) - t(e_2) &\leq 14
\end{align*}
\]

From Fig. 1, it is not hard to see that each plane representing a constraint is parallel to the vector \(z = (-1)x_1 + (-1)x_2 + (-1)x_3\), where vectors \(x_1, x_2,\) and \(x_3\) indicate time axes of independent events \(e_1, e_2,\) and \(e_3\), respectively. Thus the circumscribed polyhedron is in fact a prism. In the figure, the pentagonal prism circumscribed by all but the plane representing the constraint \(t(e_3) - t(e_2) \leq 14\) characterizes the set of allowed timed data streams, i.e., each point \((t(e_1), t(e_2), t(e_3))\) in the prism uniquely maps to a timed data stream satisfying the set of constraints.

Now, consider another set of timing constraints:

\[
\begin{align*}
    t(e_1) - t(e_2) &\leq 5, & t(e_2) - t(e_1) &\leq 3, \\
    t(e_1) - t(e_3) &\leq 5, & t(e_3) - t(e_1) &\leq 2, \\
    t(e_2) - t(e_3) &\leq 15
\end{align*}
\]

To facilitate the discussion of trace inclusion, we show the planes of the constraints in the same affine space \(\mathbb{R}^3\) as in (1), and view the space axonomatically in the direction \(z = (-1)x_1 + (-1)x_2 + (-1)x_3\), as shown in Fig. 2.

![Fig. 1. The set of timed data streams satisfying (1) can be represented as a convex polyhedron (a pentagonal prism in this case) in affine space \(\mathbb{R}^3\).](image)

\(^1\)A timed data stream over an event set \(E\) is a pair \((\alpha, \alpha)\) where \(\alpha\) is a sequence with elements from \(E\) and \(\alpha\) is a monotonically increasing sequence with elements from \(\mathbb{R}^+ \cup \{+\infty\}\).
and (3) show more similarity than computations constrained by (1) and (2).

The proposed work on quantifying similarities between timing constraint sets is thus to

- define similarity metrics (e.g., percentage of intersection or maximum distance) that reflect observations and their geometric interpretations; and
- give efficient algorithms for calculating similarities under such metrics. Under some metrics, the problem can be intractable, e.g., calculating the percentage of intersection could have exponential cost. In these cases, approximation algorithms are needed.

Our previous study has shown that the set of timed data streams allowed by a set of real-time constraints does not change when constraints between all event pairs are replaced by implicit constraints derived by applying all-pairs shortest path algorithms on the corresponding timing constraint graph. Based on this property, and the fact that the intersection of convex sets is still convex, an intersection between two sets of constrained timed data streams can be derived by forming the union of the constraint sets and applying all-pairs shortest path algorithms. Such intersections can be used for deriving a constraint set that satisfies both sets of constraints, and facilitating similarity comparisons of timing constraint sets. For example, in Fig. 3, the intersection of trace polyhedra of constraint sets (1) (bold lines) and (3) (light lines) is the hexagonal prism in dark gray; and similarity between (1) and (3) can be defined based on their closenesses to the intersection.

C. How Similarity Relations Commute

In Section II-A and II-B, we discuss similarities between timed systems and between timing constraint sets, respectively. These results, together with the existing results on satisfiability of timing constraints by timed systems, can be integrated to study the satisfaction bounds of similar timing constraint sets by similar timed systems. More specifically, assuming that timed systems $S_1$ and $S_2$ can be shown to differ by $\epsilon$ (by results in Section II-A), timing constraint sets $C_1$ and $C_2$ can be shown to differ by $\epsilon^*$ (by results in Section II-B), and $S_1$ satisfies $C_1$ with weakening function $R^{\mu}$ as mentioned in Section II-A, some interesting questions would be: (1) how does a replacement timed system ($S_2$) satisfy the original constraint set ($C_1$); (2) how does the original timed system ($S_1$) satisfy a replacement constraint set ($C_2$); and (3) how does a replacement timed system ($S_2$) satisfy a replacement constraint set ($C_2$)?

III. APPLICATION: A NON-INTRUSIVE APPROACH TO ENHANCE LEGACY EMBEDDED CONTROL SYSTEMS WITH CYBER PROTECTION FEATURES

We plan to apply the timing constraint similarity theories to the event-based feedback loop framework proposed in [16]. The framework is designed to externalize the cyber attack-tolerant logic out of the controlled system to allow for easier conception, maintenance, and extension of attack-tolerant behaviors. Under this architecture, a controlled system is monitored and compared with a system model that represents the essential components and their relationship with the controlled system to determine the health of the system. Fig. 5 depicts the high-level view of our proposed architecture.

As shown in the figure, the newly added protection logic is separated from the existing controlled system and its activation is only through event observations. In addition, the observation, reasoning, and action schemes are separated into independent modules. Such architecture allows us to change and incorporate different observation interests, reasoning schemes, and action strategies without much modification to the controlled systems or other modules.

More specifically, the external layer contains three modules, i.e., Observation, Evaluation, and Protection modules. These three modules communicate with each other through standard interfaces. The Observation module observes events generated by the controlled system and maps them into a high level abstraction so that the Evaluation model does not have to

2The interpretation of relaxing satisfactions of linear timing constraint sets by timed systems can be slightly different from relaxing satisfactions of temporal logics. The weakening function $R^{\mu}$ for a set of linear timing constraint can be defined as incrementing each timing constraint in the set by $\mu$. 

Fig. 3. The trace polyhedra of constraint sets (1) (bold lines) and (3) (light lines), and their intersection (the dark gray region).

Fig. 4. Satisfaction of similar timing constraint sets by similar timed systems.
be tied with a specific system or system specific events; instead, the information will be provided to the **Evaluation** module with high level abstractions to promote the separation of reasoning logics from individual systems. The **Evaluation** module is responsible for reasoning about the controlled system from the information provided by the **Observation** module and decides if the controlled system is behaving normally. The **Protection** module interfaces with the controlled system and imposes protective constraints on the physical units to prevent potential catastrophe.

Now, assuming that the **Evaluation** module carries a static set of timing constraints $C_1$ to be satisfied by the controlled system, and the **Protection** module carries a dynamic set of timing constraints $C_2$ that constantly changes in order to adjust the timing behavior of the controlled system. The consistency between $C_1$ and $C_2$ can be guaranteed by ensuring that the two sets do not differ by more than $\epsilon$. The satisfaction bounds mentioned in Section II-C can be used to improve the predictability of the system: the controlled system and the adaptive constraints in the **Protection** module may change, the system’s timing behavior always stays within acceptable ranges (bounded by the bounds) from the desired behavior specified in the **Evaluation** module.

### IV. Conclusion

Quantifications of similarities between timed state sequences and between timed systems have been studied. This paper presents our ongoing researches on similarities between timing constraint sets. Our preliminary results show the following:

- inclusion and intersection relations of timing constrained trace sets can be derived by applying all-pairs shortest paths algorithms on the corresponding constraint graphs; and
- intersections of timing constrained trace sets can be used to derive similarity metrics between timing constraint sets.

Our future research aims at:

- define similarity metrics between timing constraint sets;
- give efficient algorithms for calculating similarities under the metrics;
- study the satisfaction bounds of similar timing constraint sets by similar timed systems; and
- apply the theoretical bounds to our event-based feedback loop control system so that the predictability of the system can be improved.

### References


