Synthesizing Safe Bounded Timing Models Through Simulation*

Extended Abstract†

Ben Trovato‡
Institute for Clarity in Documentation
Dublin, Ohio
trovato@corporation.com

ABSTRACT

CCS CONCEPTS

• **Computer systems organization → Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability;

KEYWORDS

ACM proceedings, LaTeX, text tagging

ACM Reference Format:
Ben Trovato. 1997. Synthesizing Safe Bounded Timing Models Through Simulation: Extended Abstract. In *Proceedings of ACM Woodstock conference (WOODSTOCK'97).* ACM, New York, NY, USA, Article 4, 3 pages. https://doi.org/10.475/123_4

1 INTRODUCTION

Given a CPS platform and application SW, how can we find timing bounds which are extensible i.e. evolution or modification of the system doesn't effect the timing bounds.

A key assumption in verifying time constraints on embedded cyber physical systems is the that the given timing model is correct. The time for each step between nodes of the automata are given upper and lower timing bounds - usually given by the manufacturer of the device. This expands the trusted base to include the manufacturer.

While manufacturer guarantees can often be taken as safe assumptions, such models are not available in many other situations. For example, when embedding platform independent software into a particular system, the timing model may change based on this hardware. Furthermore, as cyber physical system component development becomes more accessible to individuals, there may not be a central manufacturer that can provide a bounded timing model.

2 BACKGROUND

As CPS are resource-constrained systems, understanding the impact of any security solutions on control performance, timing, and resources of the system is important. Furthermore, ensuring the solutions respect the semantic gap between design and implementation is crucial for its correct operation. Consequently, Lin *et al.* [3], Pasqualetti and Zhu [4] and Zheng *et al.* [5] proposed frameworks that analyses the impact of security solutions and consider the gap between controller design and implementation. Lin *et al.* [3] analysed the impact of message authentication mechanism with time-delayed release of keys on real-time constraints of a ground-vehicle. Such a security solution was developed to protect Time Division Multiple Access (TDMA)-based protocol, which is used in many safety-critical systems such as automobile and avionics electronic systems because of their more predictable timing behavior. To ensure the increased latencies (due to delayed key release) did not violate timing requirements, an algorithm to optimize task allocation, priority assignment, network scheduling, and key-release interval length during the mapping process from the functional model to the architectural model, with consideration of the overhead was developed. This algorithm combined simulated annealing with a set of efficient optimization heuristics. However, their approach did not consider the impact of their security solution on sampling periods and control performance. Furthermore, they didn't consider presence of a software platform between the security solution and hardware.

Pasqualetti and Zhu's [4] method could analyse control performance, security, and end-to-end timing of a resource-constrained CPS under network (cyber) attack that can compromise systems privacy (confidentiality). They have also quantified interdependency between the three system objectives by means of a minimal set of interface variables and relations. In their work, they have considered an adversary that has complete knowledge of the system model and can reconstruct system states from measurements. As a first step, the physical plant was modeled as a continuous time LTI system. The control input was determined using an output-based control law. A relationship was established to show that the control performance improved with reduced sampling time. Next, resiliency of the encryption method, protecting messages transmitted by sensor to controller was evaluated. It was observed that the encryption method increased the sampling period thereby degrading control performance. While implementing the control function on a CPS platform, the end-to-end delay was calculated by incorporating time incurred during sensing, computation, and communication. During development of the scheduling algorithm for the system, it was ensured that the measured delay was within the sampling period. Based on their analysis, they concluded that the control and the security algorithms should be designed based on the implementation platform so as to optimize performance and robustness.

*Produces the permission block, and copyright information
†The full version of the author's guide is available as `acmart.pdf` document
‡Dr. Trovato insisted his name be first.

WOODSTOCK'97, July 1997, El Paso, Texas USA

ACM ISBN 123-4567-24-567/08/06…$15.00
https://doi.org/10.475/123_4

Zheng *et al.* [5] quantify the impact of their security solution on control performance and schedulability. They also analyzed the tradeoffs between security level and control performance while ensuring the resource and real-time constraints were satisfied. For demonstration, a CPS with multiple controllers that share computation and communication resources was considered. A controller, which was modelled as a control task, processed information collected from sensors on a shared resource and commanded actuators to carry out task. To prevent attackers from eavesdropping on the communication medium for obtaining system's internal information, messages from sensors were encrypted. The decryption of these messages were modeled as task. Each of these tasks were given an activation period and worst case execution time. In the system, the control tasks competed for computation resources whereas as messages competed for communication resources. Incorporating the message encryption mechanism introduced resource overhead that impacted schedulability and control performance. To avoid this issue, they framed an optimization problem where control performance (a function of control task period) was the objective function and security level, computation resource, communication resource, and end-to-end latency were constraints. By varying the security level (function of messages to be encrypted), they ensured that the system achieved optimal control performance and platform schedulability.

3 PRELIMINARIES

Here we give a full formal defintion of parametic timed automata and all the stuff we need in order to eventuallyl define the *parameter bound synthesis* problem.

4 PARAMETER BOUND SYNTHESIS

The *bounded integer parameter synthesis problem* from [2], is defined as follows: Given a parametric timed automaton $\mathcal{M}$, a labeling function $\mathcal{L}$, an LTL property (verification condition) ϕ, a lower bound function $lb : P \to \mathbb{Z}$ and an upper bound function $ub : P \to \mathbb{Z}$, compute the set of all parameter valuations $v : P \to \mathbb{Z}$ such that

$$lb(p) < v(p) < ub(p) \wedge$$
$$(M, v, L) \models \phi$$

However, in the context of embedded systems, the bounds lb and ub must be taken from the manufacturer. If the manufacturer is not able to provide such bounds, or we do not want to take these as an assumptions, we need to synthesize these bounds instead.

For this reason, we formulate the *parameter bound synthesis problem*, where the goal is to find upper and lower bounds such that all valuations within those bounds satisfy the verification condition. In the CPS domain, this corresponds to the question "what are the most flexible bounds on the delay my system can induce, such that the overall system is safe?" Formally, given a parametric timed automaton $\mathcal{M}$, a labeling function $\mathcal{L}$, an LTL property ϕ, and a target valuation function $t(p) : P \to \mathbb{R}$, find the lower bound $lb : P \to \mathbb{R}$ and upper bound $ub : P \to \mathbb{R}$ functions such that

$$lb(p) \leq t(p) \leq ub(p) \wedge$$
$$\forall v.\ lb(p) < v(p) < ub(p).\ (M, v, L) \models \phi$$

subject to the linear optimization/maximization condition

$$\max(\mathcal{OP}(lb, ub, t))$$

We leave the exact optimization condition (definition of flexible bounds) up to the user, as certain use cases may call for slightly different system needs. For example, if the implementation of our system induces a large range for every parameter, we may seek to maximize the minimum range of the bounds. This finds gives us the most relaxed condition on the bounds so that, if our implementation can meet those bounds, the system is still safe.

$$\mathcal{OP}(lb, ub, t) = \min(ub(p) - lb(p))$$

It may be the case that we find that the bounds are too tight for the system implementation to guarantee. In this case, we may seek another another optimization condition, that satisfies the verification condition and is possible for the implementation to guarantee. For example, our implementation may be able to guarantee that sum of the upper bounds will be less than some value.

$$\mathcal{OP}(lb, ub, t) = \Sigma_p\ ub(p)$$

In a sense, the optimization function $\mathcal{OP}$ is an analog to finding the weakest precondition. The larger the value of the $\mathcal{OP}$, the more system implementations will satisfy the verification condition. We enforce the restriction here that it is always a maximization problem, and the $\mathcal{OP}$ is always linear (TODO is this necessary?).

4.1 Decidability

This problem can be framed as a decision problem in two different ways. First, as a general optimization, "Does there exist a global maximum value for which all valuations within the bounds satisfy ϕ?". This problem is undecidable, as we can reduce this to the unbounded integer parameter synthesis problem, which is known to be undecidable [1]. TODO proof - actually im not sure about this, even over integers.

Second, we can frame the problem as "Does there exist a global maximum value less than k for which all valuations within the bounds satisfy ϕ?" This seems decidable for integers, as we could enumerate every $lb$and ub, and check each valuation within those bounds. Th complexity of LTL model checking is exponential in the size of the spec. There are only finitely many options for $lb$and ub, since $lb$is finitely bounded by $\forall p.0 \leq lb(p) \leq t(p)$.

5 EVALUATION

To test our algorithm, we built a tool, ToolX, and tested it on real-world case studies from the CPS domain.

REFERENCES

[1] Nikola Beneš, Peter Bezděk, Kim G Larsen, and Jiří Srba. 2015. Language emptiness of continuous-time parametric timed automata. In *International Colloquium on Automata, Languages, and Programming*. Springer, 69–81.

[2] Peter Bezdek, Nikola Benes, Vojtech Havel, Jiri Barnat, and Ivana Cerná. 2014. LTL Model Checking of Parametric Timed Automata. *CoRR* abs/1409.3696 (2014). http://arxiv.org/abs/1409.3696

[3] C. W. Lin, Q. Zhu, and A. Sangiovanni-Vincentelli. 2014. Security-aware mapping for TDMA-based real-time distributed systems. In *2014 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. 24–31. https://doi.org/10.1109/ICCAD.2014.7001325

[4] Fabio Pasqualetti and Qi Zhu. 2015. Design and operation of secure cyber-physical systems. *IEEE Embedded Systems Letters* 7, 1 (2015), 3–6.

[5] Bowen Zheng, Peng Deng, Rajasekhar Anguluri, Qi Zhu, and Fabio Pasqualetti. 2016. Cross-layer codesign for secure cyber-physical systems. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 35, 5 (2016), 699–711.