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, **??** pages. https://doi.org/10.475/123_4

1 INTRODUCTION

The *bounded integer parameter synthesis problem* from **??**, is defeined as follows: Given a parametric timed automaton $\mathcal{M}$, a labelling function $\mathcal{L}$, an LTL property ϕ, a lower bound function lb and an upper bound function ub, compute the set of all parameter valuations v such that $(M, v, L) \models \phi$ and $lb(p) < v(p) < ub(p)$. However, in the context of embedded systems, the bounds lb and ub must be taken from the manufacturer. If the mafucatureer 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 *integer bound parameter synthesis problem.* Here we will try to synthesis the parameters Given a parametric timed automaton $\mathcal{M}$, a labelling function $\mathcal{L}$, an LTL property ϕ, find a lower bound function lb and an upper bound up such that $\forall v.\, lb(p) < v(p) < ub(p).\ (M, v, L) \models \phi$ and $min(\Sigma_{0->p}.\, ubp-lb(p)$.

If p is prime and $g, h \in \mathbb{Z}_p^*$, we write $\log_g(h) = n$ if $n \in \mathbb{Z}$ satisfies $g^n = h$. The problem of finding such an integer n for a given $g, h \in \mathbb{Z}_p^*$ (with $g \neq 1$) is the *Discrete Log Problem.*

Conjecture 1.1. *There does not exist a polynomial time algorithm to solve the Discrete Log Problem.*

[*]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.

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.* [?], Pasqualetti and Zhu [?] and Zheng *et al.* [?] proposed frameworks that analyses the impact of security solutions and consider the gap between controller design and implementation. Lin *et al.* [?] 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 [?] 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.

Zheng *et al.* [?] 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.

1.1 Modeling Networked CPS

The continuous-time plant in presence of noise and attack on output:

$$\dot{x}_p = f_p(x_p, \hat{u}) + w_p \qquad y = g_p(x_p, \eta(t)) + v_p \tag{1}$$

where, $x_p \in \mathbb{R}^{n_p}$ represents the state of the plant, $\hat{u} \in \mathbb{R}^{n_u}$ represents the control values implemented at the plant, $w_p \sim N(0, Q)$ is the additive zero mean Gaussian noise, $y \in \mathbb{R}^{n_y}$ is the output of the plant, $v_p \sim N(0, R)$ is the additive zero mean Gaussian measurement noise, and $\eta(t)$ is the variable for attack on sensor. During attack, $\eta(t)$ corrupts the output of the plant (y), which is measured by sensors.

The plant is controlled by a controller over the shared communication network, whose equations are:

$$\dot{x}_c = f_c(x_c, \hat{y}) \qquad u = g_c(x_c, \hat{y}, \mu(t)) \tag{2}$$

where, $x_c \in \mathbb{R}^{n_c}$ represents the state of the controller, $\hat{y} \in \mathbb{R}^{n_y}$ is the most recent output (either good or corrupted by attack) of the plant available to the controller, $u \in \mathbb{R}^{n_u}$ represents the controller output, and $\mu(t)$ is the variable for attack on actuator. The controller output (u) can be manipulated by cyber or physical attack, which is represented by $\mu(t)$. Due to the presence of communication network, $u \neq \hat{u}$ and $y \neq \hat{y}$. In this setup, sensors are time-driven and both actuator and controller are event-driven. As the controllers, sensors, and actuators are connected via a shared network, they are subject to varying transmission intervals and varying delays. Due to varying transmission intervals, the instants ($t_k \in \mathbb{R}_{\geq 0},\ k \in \mathbb{N}$) at which the plant outputs and control values are sampled and transmitted over network are non-equidistantly spaced in time. These transmitted values are received by other units in the network after a delay of $\tau_k \in \mathbb{R}_{\geq 0}$, with $\tau_k \in [\tau_{min}, \tau_{max}]$, for all $k \in \mathbb{N}$. This delay is due to the speed at which data travels in the network and it should be less than transmission interval to ensure correct operation of the controllers. In NCS, a scheduling protocol is used in the network to ensure data from all sensors and actuators are not transmitted at the same time. Moreover, while updating the values of $\hat{y}$ and $\hat{u}$, the network is assumed to operate in a zero-order-hold (ZOH) manner i.e. these values remain constant while being updated.

TODO - find a formal model to embed the above equations into some logical formula that can be checked (with wieghted max-smt?). We want to know, given a single scheduler (instance of the model), how much delay must be introduced to inalidate the saftey/stability specifications of the system. If the attacker can only introduce delay at one point (step) in the system, how much delay must be introduced. If the attacker can introduce a global delay, how much is needed? What about subsets of modules (steps in system).

Goal of Project

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 MOTIVATIONS

1) cant trust the manufacture

2) the manufacturer things could be hacked, and we want to know 'how can the timing guarantees from the manufacture be hacked' so that we still have a safe/stable system.

Outline of approach

(1) Derive average timing bounds for each atomic step of the CPS through simulation.
(2) To safely generalize the simulation Assume a probabilistic distribution over the simulation time to get bounds.
(3) Using a model checker, find the minimal distribution that implies bounds on the timing model such that the system still satisfies the safety conditions.

REFERENCES

[] 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

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

[] 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.