Toward Online Hybrid Systems Model Checking of Cyber-Physical Systems' Time-Bounded Short-Run Behavior *

Lei Bu[1,2]
bulei@nju.edu.cn

Qixin Wang[3]
csqwang@comp.polyu.edu.hk

Xin Chen[1,2]
chenxin@nju.edu.cn

Linzhang Wang[1,2]
lzwang@nju.edu.cn

Tian Zhang[1,2]
ztluck@nju.edu.cn

Jianhua Zhao[1,2]
zhaojh@nju.edu.cn

Xuandong Li[1,2]
lxd@nju.edu.cn

[1]State Key Laboratory of Novel Software Technology, Nanjing University, Nanjing, Jiangsu, P.R.China
[2]Department of Computer Science and Technology, Nanjing University, Nanjing, Jiangsu, P.R.China
[3]Department of Computing, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong

ABSTRACT

Many *Cyber-Physical Systems* (CPS) are highly nondeterministic. This often makes it impractical to model and predict the complete system behavior. To address this problem, we propose that instead of offline modeling and verification, many CPS systems should be modeled and verified online, and we shall focus on the system's *time-bounded behavior in short-run future*, which is more describable and predictable. Meanwhile, as the system model is generated/updated online, the verification has to be fast. It is meaningless to tell an online model is unsafe when it is already outdated. To demonstrate the feasibility of our proposal, we study two cases of our ongoing projects, one on the modeling and verification of a train control system, and the other on a *Medical Device Plug-and-Play* (MDPnP) application. Both cases are about safety-critical CPS systems. Through these two cases, we exemplify how to build online models that describe the time-bounded short-run behavior of CPS systems; and we show that fast online modeling and verification is possible.

1. INTRODUCTION

By combining communication, computation, and control (3C), Cyber-Physical Systems (CPS)[1] aim to more tightly couple the physical world with the cyber-world, to enable more applications, enhance performance, increase dependability/safety etc.. Among these goals, however, guaranteeing the basic dependability/safety is after all the prerequisite and often the top concern.

If we reflect upon the cyber-world, to guarantee dependability/safety, classic cyber-system engineering carries out model checking before the system is put online. Specifically, model checking involves building a formal model of the system, and verifying the dependability/safety properties of the model. If the formal model built is accurate and it passes verification, then the cyber-system's dependability/safety is guaranteed.

Model checking has been studied extensively[2], but the convention is to build a complete static formal model of the cyber-system first, and then verify the model offline before the system runs. This practice is a proven success for many categories of cyber-systems applications, e.g., in computer aided design of digital hardware. This fosters people to develop *hybrid systems model checking* [3][4] to extend model checking into the realm of CPS. Existing hybrid systems model checking tools mainly focus on combining the cyber-world logic models (mainly automata) with the physical world control theory models (mainly differential equations); and the model checking practice is still assumed to be offline (i.e., before the system runs) and hence must cover the system's long-run behavior.

However, people widely believe that such existing hybrid systems model checking practices are insufficient to meet CPS applications' needs [1, 5]. Specifically, we face two challenges:

Challenge 1: The verification state space can easily explode due to the numerous combination possibilities of sub-components [6]. Take medical CPS for example, a patient's numerous tissues, organs, and biochemical processes can directly or indirectly affect each other.

Challenge 2: For many physical-world systems, there is *no* good offline models at all. Again take the human body for example, it is a complex biochemical system. For most medical applications, there is *no* good offline control theory models that can accurately describe the behavior of human's reactions to treatments.

Due to the above challenges, the current convention of building and verifying a comprehensive hybrid systems model *before* the

*The authors in Nanjing University are supported by the National Natural Science Foundation of China (No.90818022, No.91018006, No.61021062), the National Grand Fundamental Research 973 Program of China (No.2009CB320702), and by the National S&T Major Project (2009z01036-001-001-3). The author in Hong Kong Polytechnic University (HK PolyU) is supported by Hong Kong RGC General Research Fund (GRF) PolyU 5245/09E, The HK PolyU Internal Competitive Research Grant (DA) A-PJ68, HK PolyU Newly Recruited Junior Academic Staff Grant A-PJ80, HK PolyU Fund for CERG Project Rated 3.5 (DA) grant A-PK46, and Department of Computing start up fund.

CPS system runs (i.e. *offline* model checking) is difficult and often impractical.

In contrast to the conventional model checking, runtime verification[7] is performed while the system is running. It logs program execution traces and then analyzes the traces to check whether the program implementation complies with the program specifications. Therefore, runtime verification is not for *predicting* faults *before* they ever happen.

So now we are facing a dilemma: for many (if not all) safety-critical CPS systems, on the one hand, we are unable to exhaustively prove the system's safety before the system is put online; on the other hand, the cost of possible faults once the system is put online is hard to forbear.

To address this dilemma, we propose to carry out *online* hybrid systems model checking of CPS system's *time-bounded behavior in short-run future* (simplified as *time-bounded short-run behavior* in the following). The basic idea is as follows. For a given online CPS system, we carry out periodical sampling. Without loss of generality, let us suppose the period is T. Then at every time instance of kT ($k = 0, 1, 2, \ldots$), we sample the numeric value of each observable system state parameter, and build a hybrid system model $M(kT)$ based on these numeric values. We then only check the reachable state space for the next T seconds for $M(kT)$. If any unsafe state is reachable, we stop the online system by immediately switching to an application dependent fall-back plan. Otherwise, the online system can continue for another interval of T.

This approach retains model checking's original purpose of discovering/preventing faults *before* they happen; meanwhile responds to the two aforementioned challenges: *i*) by focusing on *online* fixed parameter readings and looking into just *time-bounded* short-run future, the verification state space is greatly reduced; *ii*) within *time-bounded* short-run future, many previously hard-to-model physical systems become quite predictable and describable.

On the other hand, the online modeling and verification of time-bounded short-run behavior must be fast, so as to guarantee discovering fault/failure *before* they happens, and to provide as much time as possible for the online systems to react. In this paper, we show the feasibility of our proposed approach through two case studies of our ongoing projects: one on a *Communications Based Train Control* (CBTC) system and the other on a *Medical Device Plug-and-Play* (MDPnP) system for laser tracheotomy.

2. COMMUNICATIONS BASED TRAIN CONTROL SYSTEM

2.1 System Description

Train control system plays the key role in the safe and efficient operations of a railway system. The cutting edge of modern train control systems is the *Communications Based Train Control* (CBTC) systems. Conceptually a CBTC system has two parts: the ground system and the onboard systems. The ground system periodically track the runtime states of each train. It sends critical train control parameters via its *Radio Block Centers* (RBC) to the trains. These parameters are received by each train's onboard system, which then assists the train driver to operate the train accordingly. A key parameter is the *Movement Authority* (MA), which specifies how far ahead the train is allowed to travel. The farthest end of an MA is called the *End-of-Authority* (EOA) [8]. Usually, if train B is following train A along a same railway track, B's EOA is never allowed to reach within *Rear Safe Distance* (RSD) from the rear of A. Meanwhile, each train maintains another online parameter called *Safe Braking Distance* (SBD) point. SBD point is a location along the railway track ahead of the train but closer than EOA. The distance between SBD point and EOA is the *theoretical* minimum distance needed for the train to completely stop. Therefore, one safety rule is that if a train reaches its SBD point, it should immediately brake, in attempt not to exceed EOA.

We studied a typical CBTC system, which is part of an Urban Railway System under construction in China. In the CBTC system, the RBCs try to update their respective trains' MAs/EOAs in every 500 milliseconds. On receiving an update, a train's onboard system derives a legal operation speed range by considering all factors including current states of the train, railway, wind speed etc..

The train is free to move within this speed range until reaching the current SBD point. But there are two safety rules:

Rule 1: Once the SBD point is reached, the train must brake in standard procedure in attempt to stop before EOA.

Rule 2: If the train has not received MA/EOA update from RBC for 5 seconds, the train will brake emergently.

To guarantee safety, we must model check that each train never exceeds MA nor collides into the train ahead.

2.2 Modeling

The CBTC system is a typical hybrid system involving continuous time physical kinetics and discrete time computer logics. Therefore, we attempt to use well-known hybrid automata[4] to model the system offline[9].

Our model consists of n trains running on the track, and m RBC centers that communicate with their respective trains. The hybrid automata for train and RBC center are shown in Fig.1.A and Fig.1.B respectively.

To be more focused, in the rest of the paper, we shall concentrate on one scenario that is of top concern for CBTC system designers (referred to as *top-concern scenario* in the following). The scenario is as follows. When a train $Train_i$ does not receive any signal for 5 seconds, it will brake emergently. People are concerned about whether it may still pass the EOA point or even collide into the train ahead. Mapping back to the hybrid automata, the above scenario corresponds to the case that train $Train_i$ move from location *compute* to *Ebraking* in Fig.1.A, and a verification target property that the physical position of $Train_i$ equals that of the train ahead (i.e., $Train_{i-1}$).

However, our offline model checking encounters great difficulties. First, due to the huge number of trains and RBCs and complex nonlinear control functions involved, the offline verification state space easily explodes. This makes offline verification impractical. Second, to calculate MA, SBD, and new speed range n and n', we need the exact numerical values of train states, including x (train location), $\dot{x}$ (train speed), $\ddot{x}$ (train acceleration), etc. (see function $k()$, $f()$, $g()$, and $g'()$ in Fig. 1). However, there is no good way to predict these numerical values *offline*. To give one example, the onboard system can only specify a range of speed that the train has to comply with. The exact train speed $\dot{x}$ at each time instance is

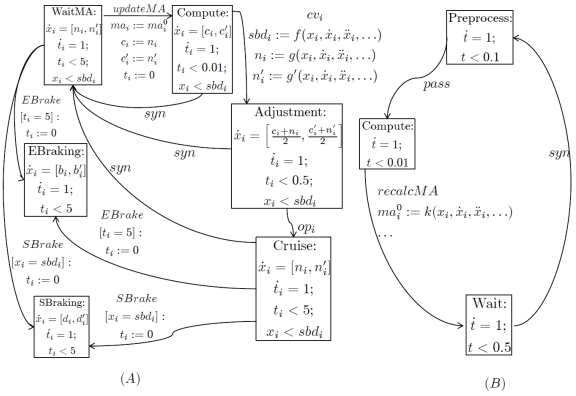

Figure 1: Hybrid Automata for CBTC: (A) for a train; (B) for an RBC Center. Note some details are omitted for simplicity.

affected by numerous factors, such as the human driver's maneuvering, train engine conditions, railway conditions, wind velocity etc.. The interactions of these factors are highly random and nonlinear. So are many other states of the train (such as x, $\ddot{x}$, ...).

To address the above difficulties, we instead try the online hybrid systems model checking of time-bounded short-run behavior. The problem immediately becomes simple.

Every time a train receives the update on MA/EOA, we can build a hybrid automaton for that train for the next 5 seconds (the maximum interval into the future that the train is allowed to receive no MA/EOA updates). Because we are building the automaton online, the exact numeric values of the train states are all available: just sample them when need them. Even better, because the modeling is conducted online on the train, the CBTC onboard system is already running, so we can directly use its SBD, n, and n' outputs.

This results in a much simplified train hybrid automaton, as shown in Fig. 2 (note many locations and transitions like Sbraking in Fig. 1 are omitted in this model, because we are now only focusing on the aforementioned top-concern scenario). The new MA numerical value ma_i^0 is got online from the RBC, the new SBD, n, and n' numerical values (sbd_i^0, n_i^0, and ${n'}_i^0$ in Fig. 2) are got online from the train's CBTC onboard system. For the train to verify its safety 5 seconds into the future, it also needs to build the hybrid automata for the trains right ahead and behind it. Fortunately, these three automata are the only automata a train needs to build every time it receives an update of MA/EOA. There is no need to build the automaton for RBC during online verification, as the train will have no interaction with the RBC till the rebuilding of the online model.

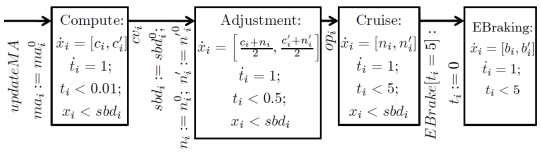

Figure 2: Scenario-Based Online Hybrid Automaton for $Train_i$.

2.3 Verification

To check the feasibility of our proposed online short-run model checking approach, we carry out proof-of-concept experiments. The hybrid systems model checker is BACH[10, 11], a toolset for building *Linear Hybrid Automata* (LHA) models and verifying bounded reachability properties. The computation platform is a DELL workstation (Intel Core2 Quad CPU 2.4 GHz, 4 GB RAM).

From the experiments, we find that our online modeling and verification can be finished in only 58 milliseconds, which is far less than the RBC updating period of 500 milliseconds. Meanwhile, the runtime memory overhead of the computation is also very small[1].

3. MDPNP IN LASER TRACHEOTOMY

To show broad applicability of our proposed approach, we also study whether this approach can be used in *Medical Device Plug-and-Play* (MDPnP, `http://www.mdpnp.org`), another representative application area of CPS.

The objective of MDPnP is to let disparate embedded medical devices collaborate in complex medical scenarios, to guarantee safety and enable more advanced medical treatments. A typical MDPnP application involves various embedded medical devices of drastically different discrete program logics; it also involves the patient-in-the-loop, which can be regarded as a continuous system. Meanwhile, as medical applications are often life critical, the safety of interactions between medical devices, patient, and medical personnel must be verifiable. Hybrid systems verification thus becomes an indispensable procedure.

In MDPnP system verification, medical device programs can be modeled with classic tools, such as automata; the patient modeling, however, is often a challenge. Although there are cases where patient can be modeled as classic continuous systems (such as linear feedback control systems [12, 13]), human body is often too complex a continuous system to fit in any mature modeling methods. This problem is particularly eminent for offline verification, as it needs a model that predicts the patient's long term reactions to treatments. For example, if 1ml of Morphine is injected, how will the patient's SpO_2 curve react in the next 10 minutes? The real-world answers can vary greatly depending on the patient's age, weight, other medicines taken, and many other known or unknown factors. Even worse, these factors are often affecting each other in an unpredictable way.

This forces us to think of online verification instead. Because for short run, many patient parameters are quite predictable. For example, a patient's SpO_2 level cannot jump from 99% to 59% in just one second; it has to follow a smooth curve. We can therefore use mature prediction tools, such as linear regression, to make accurate short-run predictions (in fact this is what the doctors do: take anesthesia for example, an anesthetist has to keep watching the patient's most current vital signs to predict and adjust online). Our preliminary evaluations on laser tracheotomy show that online hybrid system verification is a promising way to go.

The laser tracheotomy MDPnP scenario is as follows. Due to general anesthesia, the patient is paralyzed, hence has to depend on the ventilator to breath. The ventilator has three states: pump out (patient inhale oxygen), pump in (patient exhale), and halt (patient exhale naturally due to chest weight). The demand is that when the laser scalpel is to cut windpipe, the windpipe oxygen level must be lower than a threshold to prevent fire[14]. Therefore, the ventilator

[1]Due to space limit, the detailed information of the case studies of the CBTC project and the related works are referred to the extended technical report version [9] of this paper.

must stop pumping out (i.e. stay in either pump in or halt state) for enough long time before the laser scalpel can turn on. However, the ventilator can neither stop pumping out for too long, or the patient may suffocate due to too low blood oxygen level. Therefore, the whole scenario implies a complex hybrid system involving windpipe oxygen sensor, blood oxygen sensor, ventilator, laser scalpel, supervisor (the central computer for decision making), the patient, and the surgeon.

In practice, the windpipe oxygen level can be effectively modeled with first order differential equations [14, 12]. However, due to complex biochemistry, the offline modeling of blood oxygen level is very difficult. Therefore, we choose to use online verification.

Every second, the windpipe oxygen sensor and the blood oxygen sensor update their readings: we denote windpipe oxygen level and blood oxygen level at time t as $O_2(t)$ and $SpO_2(t)$ respectively. Without loss of generality, suppose at time instance t_0, sensor readings are updated. Then we build online a hybrid system to describe the medical scenario for interval $(t_0, t_0 + 1]$ (second), and verify the safety of the system for this interval. This hybrid system involves five hybrid automata, for ventilator, laser scalpel, supervisor, patient, and surgeon respectively. Due to space limit, we use the most representative example of the patient hybrid automaton to illustrate our online modeling (see Fig.3). In Fig.3, we use existing classic differential equations[14, 12] to model $O_2(t)$. We know the blood oxygen level changes slow enough at the time granularity of seconds. Therefore, we can safely use the $SpO_2(t)$ readings in the past 10 seconds to estimate (e.g. via linear regression) $S\dot{p}O_2(t_0)$. We denote the estimation as $Estimate(S\dot{p}O_2(t_0))$. Again, since blood oxygen level changes slow enough at the time granularity of seconds, we can safely assume $S\dot{p}O_2(t) \equiv Estimate(S\dot{p}O_2(t_0))$ $(\forall t \in (t_0, t_0 + 1])$. Thus we have a practical online model of the patient for interval $(t_0, t_0 + 1]$ (second) as shown in Fig.3.

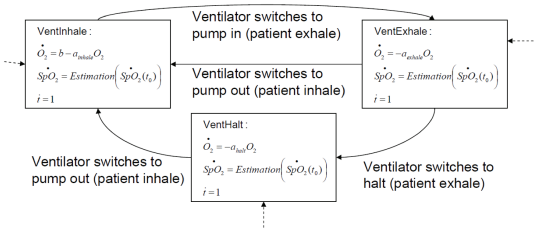

Figure 3: Online Hybrid Automaton For Patient

We carry out a preliminary experiment using PHAVer[15], another state-of-the-art linear hybrid automata model checker, to verify the system. The computing platform is a Lenovo Thinkpad X201 with Intel Core i5 and 2.9 G memory, running 32-bit Ubuntu 10.10. Our result shows that (with certain simplification) within 0.27 second, we can verify this online composition hybrid system, which is also less than the sampling period: 1 second.

4. CONCLUSION

Through our case studies in CBTC systems and MDPnP, we show that online hybrid systems model checking for CPS systems time-bounded short-run behavior is a promising approach for the verification of CPS dependability and safety.

Our case studies show this approach is effective in addressing the two major challenges facing CPS hybrid systems model checking. First, by just looking into short-run future, and by fixing multiple parameters with online numerical values, the verification state space is greatly reduced. Second, as many physical-world systems are quite predictable/describable for short-run future, and as many parameters are fixed due to online modeling, the previously hard-to-model-offline CPS systems can be easily modeled online.

Our experiment results show that the online modeling and verification can be finished quickly. This further corroborates the feasibility of our proposed approach.

5. REFERENCES

[1] E. Lee, "Cyber- physical systems- are computing foundations adequate?" *Position paper for NSF workshop on Cyber-Physical Systems: Research Motivation, Techniques and Roadmap*, 2006.

[2] E. Clarke, O. Grumburg, and D. Peled, *Model Checking*. MIT Press, 1999.

[3] P. Tabuada, *Verification and Control of Hybrid Systems: A Symbolic Approach*. Springer, 2009.

[4] T. Henzinger, "The theory of hybrid automata," *Proc. of LICS'96*, pp. 278–292, 1996.

[5] E. Clarke, B. Krogh, A. Platzer, and R. Rajkumar, "Analysis and verification challenges for cyber-physical transportation systems," *National Workshop for Research on High-Confidence Transportation Cyber-Physical Systems: Automotive, Aviation and Rail*, 2008.

[6] B. I. Silva, O. Stursberg, B. H. Krogh, and S. Engell, "An assessment of the current status of algorithmic approaches to the verification of hybrid systems," *Proc. of CDC'01*, vol. 3, pp. 2867–2874, Dec. 2001.

[7] B. Finkbeiner, S. Sankaranarayanan, and H. Sipma, "Collecting statistics over runtime executions," *ENTCS*, vol. 70:4, 2002.

[8] A. Platzer and J. Quesel, "European train control system: A case study in formal verification," *Proc. of ICFEM'09*, pp. 246–265, 2009.

[9] L. Bu, X. Chen, L. Wang, and X. Li, *Online Verification of Control Parameter Calculations in Communication Based Train Control System*. [Online]. Available: {http://arxiv.org/abs/1101.4271}

[10] L. Bu, Y. Li, L. Wang, X. Chen, and X. Li, " BACH 2: Bounded ReachAbility CHecker for Compositional Linear Hybrid Systems," *Proc. of DATE'10*, pp. 1512–1517, 2010.

[11] L. Bu and X. Li, "Path-oriented bounded reachability analysis of composed linear hybrid systems," *International Journal on Software Tools for Technology Transfer*, DOI:10.1007/s10009-010-0163-9.

[12] D. Arney, M. Pajic, J. Goldman, I. Lee, R. Mangharam, and O. Sokolsky, "Toward patient safety in closed-loop medical device systems," *Proc. of ICCPS'10*, Apr. 2010.

[13] J. X. Mazoit, K. Butscher, and K. Samii, "Morphine in postoperative patients: Pharmacokinetics and pharmacodynamics of metabolites," *Anesthesia and Analgesia*, vol. 105, no. 1, pp. 70–78, 2007.

[14] C. Kim, M. Sun, S. Mohan, H. Yun, L. Sha, and T. F. Abdelzaher, "A framework for the safe interoperability of medical devices in the presence of network failures," *Proc. of ICCPS'10*, Apr. 2010.

[15] G. Frehse, "PHAVer: Algorithmic Verification of Hybrid Systems past HyTech," *Proc. of HSCC'05*, pp. 258–273, 2005.