Challenges and Opportunities in Design and Operation of Intelligent Cyber-Physical Systems

Invited Talk, 19th International Conference on Runtime Verification (RV)
Part of the 3rd World Congress on Formal Methods
Porto, Portugal. October 10, 2019

Akshay Rajhans, PhD
arajhans@mathworks.com
https://arajhans.github.io
Talk outline

- Cyber-physical systems: a feature classification
  - “Runtime” verification at design time: simulation as a proxy for run time
  - Runtime analysis at operation time: From CPS to IoT and Digital Twins
  - Challenges and future outlook
The term cyber-physical systems (CPS) refers to a new generation of systems with integrated computational and physical capabilities that can interact with humans through many new modalities.

The ability to interact with, and expand the capabilities of, the physical world through computation, communication, and control is a key enabler for future technology developments.”

- Helen Gill and Kisan Baheti, NSF
Control – closing the loop over the physical environment
Computation – fueled by Moore’s law

Moore’s Law – The number of transistors on integrated circuit chips (1971-2018)

Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years. This advancement is important as other aspects of technological progress – such as processing speed or the price of electronic products – are linked to Moore’s law.


Computation – fueled by Moore’s law

Cost of Printing > Annual Production

Source: Intel, IEEE
Communication – Nielsen’s law, Metcalfe’s law

Source: nngroup.com

CPS feature classification

- Distributed
- Connected
- Collaborative
- Coalition

Communication Axis

Distributed
- Sense
- Act
- Environment

Connected
- Sense
- Act
- Environment

Collaborative
- Sense
- Act
- Environment

Coalition
- Sense
- Act
- Environment

Automatic
- Sense
- Act
- Environment

Adaptive
- Sense
- Act
- Environment

Autonomous
- Sense
- Act
- Environment

Aware
- Sense
- Act
- Environment

Plan
- Sense
- Act
- Environment

Reflect
- Sense
- Act
- Environment

Computation Axis


Outline

- CPS feature classification
  - “Runtime” verification at design time: simulation as a proxy for run time
- From CPS to IoT and Digital Twins: runtime analysis
- Challenges and future outlook
Model-Based Design: Models as a proxy for the real system

Requirements

Architecture

Components

Specification and design

Integration and testing

Unit Testing

Integration Testing

Acceptance Testing

Implementation

Model-Based Design: Use of computational models throughout
Simulations for increasingly faithful proxies of runtime behavior

- Model-in-the-loop simulation: simulate / test the model
- Software-in-the-loop simulation: the generated code
- Processor-in-the-loop simulation: code on the processor
- Hardware-in-the-loop simulation: plant on real-time h/w
- Gaming-engine-in-the-loop: visualization, physics

Ride & handling  |  Chassis controls  |  ADAS / AD
Not today – how to address heterogeneity formally?


Models and Specifications

“overshoot is never more than 30% and settling time is less than $\tau$”

Specifications

Model $M$

$\text{Model } M$

A behavior $b$ that $M$ exhibits

Specification $S$

$\text{Specification } S$

A behavior $b$ that $S$ allows
Various verification problems

Abstraction

Reachability Analysis

Tight Overapproximation

Robust Testing

Robust Neighborhoods

Testing

Theorem Proving

Model checking

Specification Mining

Falsification
Robust testing a.k.a. simulation-based reachability analysis
Robust testing a.k.a. simulation-based reachability analysis


Related work by Fainekos, Pappas, Balkan, Tabuada, Zutshi, Sankaranarayanan, Kanade, Alur, …

Lyapunov analysis, contraction metrics, barrier certificates, concolic testing …
Toyota adoption a success story

Simulation-Based Approaches for Verification of Embedded Control Systems

JAMES KAPINSKI, JYOTIRMOY V. DESHMUKH, XIAOQING JIN, HISAIRO ITO, and KEN BUTTS
Formalizing specifications to enable falsification
Signal Temporal Logic as a success story

Barbaric reachability [KKMS03]
Quantitative Semantics for MTL [FP06]
STL Parameter Synthesis [ADMN11]
Efficient Monitoring of Quantitative STL [DFM13]
Learning STL specifications [BVPYB16]

STL introduced [MN04]
Falsification Testing [NFS+10]
Quantitative Semantics for STL [DM10]
Toyota promotes falsification testing and specification mining
2012 – [JDDS13][AH+14]
Trace diagnostics for STL [FMN15]
System diagnostics for STL [BFMN18]

Credit: Dejan Ničković (via Bruce Krogh), Oded Maler: A memory box full of diamonds, MT-CPS 2019.
An actual bug uncovered via falsification at Toyota

Mining Requirements from Closed-Loop Control Models

Xiaoqing Jin
Univ. of California Riverside
jinx@cs.ucr.edu

Alexandre Donzé
Univ. of California Berkeley
donze@eecs.berkeley.edu

Jyotirmoy V. Deshmukh
Toyota Technical Center
jyotirmoy.deshmukh@tema.toyota.com

Sanjit A. Seshia
Univ. of California Berkeley
sseshia@eecs.berkeley.edu
Considerations for engineering adoption of temporal logics

- Engineers are not logicians – logic vocabulary could be a challenge
- Simple engineering concepts may require complex logical formulas
- Multiple modeling formalisms that interact
- Multiple combinations of time/signal domains, data types, solver settings
Grand Challenge I: Requirement Engineering

- Key challenge for Toyota, Bosch, and others
  - How do you present requirements to control designers?
  - How do they convey their intention without using formalisms?
  - Is Temporal Logic the right requirement language?
Logical and temporal assessments in Simulink Test

- Formalize and execute requirements directly as Test Assessments

Requirements

1. The difference between the room temperature and the set temperature should never exceed 6 degrees.

2. If the temperature difference exceeds 4 degrees for more than 2 seconds, then the pump shall activate for at least 2 seconds.

Formalize and execute
\[ \square_{[t_0,t_f]} \left( (x \bowtie a) \ominus (x \bowtie b) \right) \]
Authoring

>> sltestmgr

\[ [t_0, t_f] \phi \]
Authoring

>> sltestmgr

\[ \Box_{[t_0,t_f]} (\phi_1 \Rightarrow \Diamond_{[t_m,t_M]} \phi_2) \]
Authoring

$\texttt{sltestmgr}$

$\square_{[t_0, t_f]} (\phi_1 \Rightarrow \Diamond_{[t_m, t_M]} \phi_2)$
Symbol resolution and mapping
Assessment and explanation in case of failure

Visual Comparison: expected vs actual

Textual explanation

Assessment tree
Expression tree  \[ \square_{[t_0,t_f]} (\phi_1 \Rightarrow \lozenge_{[0,t_M]} \phi_2) \]

- Tested and failed
- Rising edge
- Did not get tested

\( (\phi_1 \Rightarrow \phi_2) \equiv \neg \phi_1 \lor \phi_2 \)?
Synchronization and interpolation

Research challenge: heterogeneity
- discrete and continuous time
- discrete and continuous value
  - STL $\triangleright$ LTL?
- Needing to up/down-sample may impact frequency domain characteristics
- Dataflow domain: cannot insert or remove data points

Currently not supported
References

Outline

- Cyber-physical systems: a feature classification
- “Runtime” verification at design time: simulation-based approaches
- Runtime analysis at operation time: From CPS to IoT and Digital Twins
- Challenges and future outlook
Models are useful in both design and operation

Internet of Things topology
Internet of Things topology

**mytoaster**
@mytoaster

Social Networking for your Toaster. The Internet of Things (IoT) powered by @ThingSpeak, created by @schuler

- Pittsburgh, PA
- nothans.com/social-network...
- Joined December 2008
- Photos and videos

**CheerLights**
@cheerlights

CheerLights is an InternetOfThings project by @schuler to synchronize lights to the same color at the same time all around the world. Not ThingSpeak

- Pittsburgh, PA
- cheerlights.com
- Joined November 2011
- 35 Photos and videos

Internet

Sense

Act

Environment
Industrial Internet of Things topology – Enterprise level operations

Smart assets – Edge systems – On-prem data center – Off-prem cloud

Local Communications – Long-Range Communications – Integration

Sense – Adapt – Plan – Reflect

Act – Environment – Act – Environment

Act – Environment – Act – Environment

Act – Environment – Act – Environment

Act – Environment – Act – Environment
A complex collection of tools, platforms, and protocols

- Chipmakers
- Transport protocols
- Operating systems running on edge or on-premises
- Cloud providers
- Streaming protocols for getting data in and out of the cloud platforms
- Services for managing data, timing, and other Industrial IoT requirements
- Dashboard tools for visualizing information in any area of the landscape
Applications at the Asset, the Edge, or Operational Technology Platform

- **Smart assets**
- **Edge systems**
- **On-prem data center**
- **Off prem cloud**

**Value of data to decision making**

- **Speed**
  - Milliseconds
  - Seconds
  - Minutes
  - Hours
  - Days
  - Months

- **Scope**
  - Hard real-time control
  - Real-time decisions
  - Time-sensitive decisions
  - Big Data processing on historical data

**Time-sensitive decisions**
Development for Fast and Highly-Deterministic Systems

- Compute limited but have a choice
- Data usually as a memory read
- Product design focus

Model-Based Design

- Multi-domain system modeling
- Parameter estimation
- Automatic code generation

Value of data to decision making

- Speed
  - Milliseconds
  - Seconds
  - Minutes
  - Hours

Smart assets

Edge systems

Local Communications
Development to OT/IT On-Prem and in Cloud

- Compute abundant but less control
- Data access as streaming messages
- Service focus

On-prem data center

Off prem cloud

Long-Range Communications

Integration

Time-sensitive decisions

Big Data processing on historical data

Stream Processing

Hadoop/Spark, and other enterprise IT integration

Scope

Time

Variety and Volumes of Data

Machine Learning and Deep Learning

Enterprise system integration, (on-prem/cloud)

Optimization
MathWorks Cloud

Public Clouds

Use MATLAB on virtual machines in public cloud environments like Amazon Web Services (AWS) and Microsoft Azure. These vendors provide access to on-demand computing resources. They also offer wide-ranging, prebuilt services for data storage, data streaming, elastic scaling, load balancing, security, and more.

If you are not a cloud expert, or if you want a head start, use a MathWorks published reference architecture. Templates in these reference architectures automatically create and configure the cloud infrastructure for running MATLAB. You can also adapt or extend the reference architectures to better meet your specific needs.

Learn more about running MATLAB and other products on:

[Links to AWS, Azure, and Other Clouds]
Create computational model of asset in operation
- Data-driven (MATLAB) or first-principles (Simulink) models
- Reuse models from development process (e.g. MBD)
- Kept up-to-date during asset operation (e.g. aging, wear, environment)

Use the computational model (digital twin) during operation
- Optimize fleet or system behavior
- Calculate control setpoints or parameters
- Predict future behavior or events
Reference example

1 Fault Classification Using MATLAB

- Machine Learning fault classifier model
- Visualization dashboard
- MATLAB
- Statistics & Machine Learning Toolbox
- MATLAB Production Server

2 “What-If” Analysis Using Simulink/Simscape Digital Twin

- Model tuned during operation
- Parallel sims to explore scenarios
- Simulink/Simscape
- Simulink Design Optimization
- MATLAB Parallel Server

Triplex Pump
Research Connections

- **Optimize**
- **Control**
- **Predict**
- **Diagnose**
- **Analyze**
- **Monitor**

---

**Coordinated Control of Wind Turbine Generator and Energy Storage System for Frequency Regulation under Temporal Logic Specifications**

Zhe Xu, Agung Julius and Joe H. Chow

2018 Annual American Control Conference (ACC), June 27-29, 2018, Wisconsin Center, Milwaukee, USA

---

**Advisory Temporal Logic Inference and Controller Design for Semi-autonomous Robots**

Zhe Xu*, Student Member, IEEE, Sayan Sah*, Student Member, IEEE, Botto Hu*, Student Member, IEEE, Sandipan Mishra, Member, IEEE, and A. Agung Julius*, Member, IEEE

2017 IEEE 56th Annual Conference on Decision and Control (CDC), December 12-15, 2017, Melbourne, Australia

---

**Provably Correct Design of Observations for Fault Detection with Privacy Preservation**

Zhe Xu, Sayan Saha and Agung Julius

---

**Verification of Hybrid Automata Diagnosability with Measurement Uncertainty**

Yi Deng, Alessandro D’Andrea, Maria D. Di Benedetto, Stefano Di Gennaro, and A. Agung Julius

2016 Annual American Control Conference (ACC), June 27-29, 2016, Wisconsin Center, Milwaukee, USA

---

**Robust Temporal Logic Inference for Hybrid System Observation: An Application on Occupancy Detection of Smart Buildings**

Zhe Xu, Yi Deng and Agung Julius

Abstract—In modern smart buildings modeled as hybrid systems, occupancy detection can be cast as observing the discrete state of a hybrid system using the available discrete and continuous system outputs. In this paper, we present a method to construct observers of the hybrid system to distinguish between different locations of the hybrid system by inferring temporal logic (TL) formulas from the simulated trajectories. We first approximate the system behavior by annotating likely many trajectories with finite robot take segments around them. These finite-robot take segments account for both spatial and temporal uncertainties. The inferred TL formulas classify different finite-robot take segments and then can be used for classifying the hybrid system behavior in a provably correct fashion. We implement our approach on a model of a smart building to distinguish two cases of room occupancy.

---

**MathWorks**

---
References


- https://www.mathworks.com/cloud.html
- https://github.com/mathworks-ref-arch
In summary

- Cyber-physical systems continue to gain intelligence and autonomy

- CPS are open, interconnected, and change after deployment

- Formal specification and simulation-based approaches fill an important scalability gap w.r.t formal verification

- Model-Based Design approaches are being supplanted by model-based operation

- Scalability to enterprise-level system will be the value driver
Thank you!

- Thank you to the RV organizers, Leo, and Bernd
- Dagstuhl on Specification and Verification of CPS
- Jean-Francois Kempf, Khoo Yit Phang, Isaac Ito, and team
- Terri Xiao, Dan Lluch, Jim Tung, Pieter Mosterman, and team