Towards Formal Specification Visualization for Testing and Monitoring of Cyber-Physical Systems

Bardh Hoxha, Hoang Bach, Houssam Abbas, Adel Dokhanchi, Yoshihiro Kobayashi, and Georgios Fainekos

Email: {bhoxha,hbach,hyabbas,adokhanc,ykobaya,fainekos}@asu.edu
Cyber-Physical Systems

- Wheel Speed Sensors (4)
- Steering Angle Sensor
- Lateral Accelerometer Yaw Rate Sensor
- Powertrain Control Module
- ABS/TSC/VSE Modulator Controller

- Insulin infusion pump
- Wearable continuous glucose monitor

Blood Glucose vs. Time

[Image by Ken Butts, Toyota]
Cyber-Physical Systems (CPS) refers to the class of systems that integrate (engineered) physical systems with computational and communication systems.
In general, verifying a hybrid system is an **undecidable problem**!

What can be done?

• Previous approaches to the problem:
  • identifying decidable classes
  • barrier certificates / invariants
  • reachability algorithms
  • robust testing
  • systematic simulations / model based testing
  • statistical techniques

System complexity
Completeness of analysis
S-TaLiRo

S-TaLiRo is a testing and verification tool for Cyber-Physical Systems

Main functionalities:

• Falsification

• Parameter Estimation

• Expected Robustness for Stochastic Systems

• Runtime Monitoring

• Conformance Testing
Specification: Metric Temporal Logic (MTL)

Syntax:

$$\Phi ::= T | F | p | p | \Phi_1 \lor \Phi_2 | \Phi_1 \land \Phi_2 | \Phi_1 \Uparrow I \Phi_2 | \Phi_1 \mathcal{R} I \Phi_2$$

$I$ can be of any non-empty bounded or unbounded interval of $\mathbb{R}^+$

Derived operators:

Eventually (in the future)  $$F_I \Phi := T \mathcal{U} I \Phi$$

Always (globally)  $$G_I \Phi := F \mathcal{R} I \Phi$$

Koymans ’90, Specifying real-time properties with metric temporal logic
Robustness is a functional that associates a real extended number to each sample path $y$ of $Y$.

Positive robustness $\rightarrow$ signal satisfies the formula

Negative robustness $\rightarrow$ signal falsifies the formula
Robustness of MTL formulae

Robustness is a functional that associates a real extended number to each sample path \( y \) of \( Y \).

- **positive robustness** \( \rightarrow \) signal satisfies the formula
- **negative robustness** \( \rightarrow \) signal falsifies the formula

\[
\rho_\varphi : y(\cdot, \omega; \theta) \mapsto \rho_\varphi(y(\cdot, \omega; \theta)) \equiv \rho_\varphi(\omega, \theta) \in [-\infty, \infty]
\]

The average robustness \( U(\theta) \) captures the average behavior of the system for that \( \theta \). We minimize \( U(\theta) \) to find worst-case average behavior.

\[
U(\theta) = \mathbb{E}_P[\rho_\varphi(\omega, \theta)] = \int_{\Omega} \rho_\varphi(\omega, \theta) dP(\omega)
\]
The falsification method searches for counterexamples that prove that the system does not satisfy the specification.

Parameter Estimation

Given a parametric specification, system engineers would like to infer the ranges of parameters for which a property holds on the system.

Cyber-Physical System $\Sigma$

MTL Specification $\phi[\lambda]$

(monotonically non-decreasing or non-increasing)

Modify Cost Function

Stochastic Optimizer

Parameter range for which the systems does not satisfy the specification

Expected Robustness for Stochastic Systems

A verification method that enables system engineers that a stochastic system is robust and operates within set specifications.

Runtime Verification

In on-line monitoring, an independent monitor can observe the system execution/simulation without intruding on its functionality and it may report potential violations to a supervisor for further control actions.

Problem statement

Given two (possibly black-box) systems H1 and H2:

Calculate the **closeness** between them

**Verify** that H2 satisfies some specification

**Conclude automatically** (without checking) what related specifications are satisfied by H1
S-TaLiRo modular architecture

S-TaLiRo Graphical User Interface

MTL Specification Tool

MTL Spec $\phi$, Model $\Sigma$, S-TaLiRo options

S-TaLiRo

Conformance Testing

conformance distance $[\tau_1, \tau_2]$

observation trajectories $[y_1 y_2]$

Runtime Verification

observation trajectory $y$

robustness $\gamma$

Stochastic Optimization Engine

Generate Input Signals

next $x_0, \hat{\varphi}$

$y(i), O(p)$

robustness $[\varepsilon_1 ... \varepsilon_f]$

parameter $\hat{\theta}$

Robustness Computation Block

Convex Optimization

distance

TaLiRo

observation trajectories $[y_1 ... y_f]$

System Simulator Engine

Hybrid Automata

User Defined Functions (blackbox)

Simulink Model

Hardware-in-the-loop

Processor-in-the-loop

Minimum Robustness

Minimum Expected Robustness

Closest Conformance Distance

Estimated Parameter

Falsifying Trajectory
Visual Specification Tool

• Developing MTL specifications requires a level of mathematical training that many users may not have.

• The training required takes a certain amount of time and effort

• Writing formal specifications is an error prone task

These are some of the main reasons there is a reduced willingness of the industry to adopt formal specifications
If within the first 5 seconds, vehicle speed goes over 100, then from that moment on, the engine speed (rpm), for the next 5 seconds, should always be over 4000.

\[ \phi = \Box_{[0,40]}((speed > 80) \rightarrow \Box_{[0,40]}(rpm > 4000)) \]
Visual Specification Tool

Specifications defined by templates and relationships between them

Template settings:

Temporal operators:
- Now
- Always
- At least once
- Eventually Always
- Repeatedly often and finally
Visual Specification Tool

Example 1:

Example 2:

\[ [0,36] (rpm < 4000) \]

\[ [0,40] (speed > 100) \]
Visual Specification Tool

Example 3:

\[ [0,30] \diamond [0,10] (speed > 100) \]

Example 4:

\[ [0,40] \diamond [0,13] (speed > 100) \]
Visual Specification Tool

Example 5: Implication

\((\Diamond_{[0,40]}(speed > 100)) \rightarrow (\Diamond_{[0,30]}(rpm > 3000))\)
Example 6:
Conjunction

\[
(\Box_{[0,40]}(\text{speed} < 100)) \land (\Box_{[0,40]}(\text{rpm} < 4000))
\]
Example 7: Nested sequence

\[ \diamond_{[0,40]} (\text{speed} > 80) \rightarrow \square_{[0,40]} (rpm > 4000) \]
Modularity and Development Challenges

Benefits of modularity in S-TaLiRo:
• Interchangeable optimization functions
• Independent uses of the robustness computation functions independently
• Independent uses of the simulation functions
• Improved maintainability and scalability

Challenges:
• Academic environment – students developing code and improving their coding skills
• Maintaining high quality code
• Maintaining backward compatibility with previous versions of Matlab
• Support for individual toolboxes, managing licenses etc.
Conclusions

• Up-to-date overview and functionalities of S-TaLiRo
• Discussion on the challenges faced in using MTL specifications in the industry
• Proposed a graphical formalism to facilitate the use of formal specifications in the industry
• Discussion on the modularity of the tool and the challenges faced in development

Future work:
• Authors will finalize development of the visual specification tool
• Conduct user studies to assess the effects of the proposed approach
• Consider the inclusion of the Until operator in the graphical formalism
Acknowledgements

This work was partially supported under NSF awards CNS 1116136, IIP-0856090 and the NSF I/UCRC Center for Embedded Systems.

Sponsors

Tools at: https://sites.google.com/a/asu.edu/s-taliro/