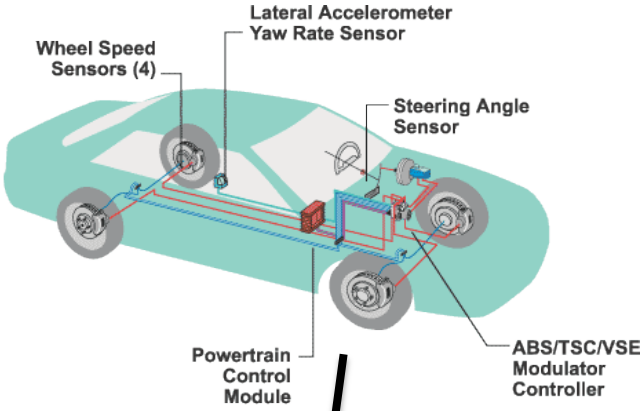
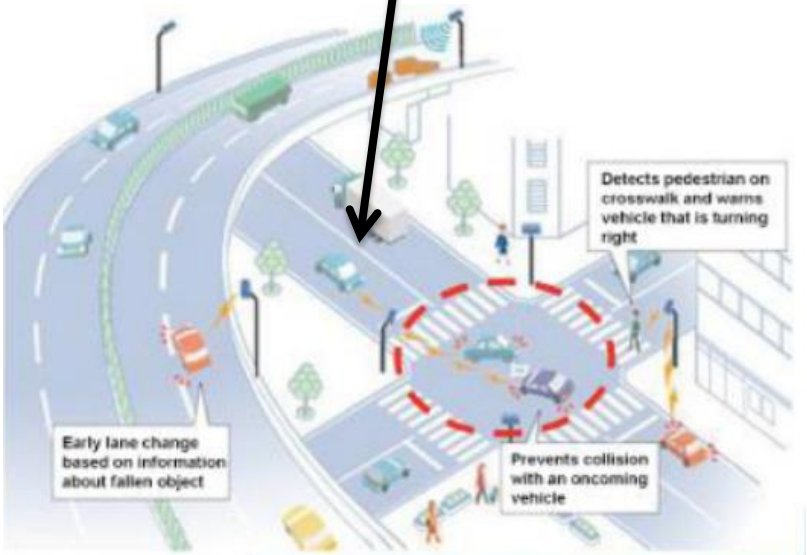
Towards Formal Specification Visualization for Testing and Monitoring of Cyber-Physical Systems

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



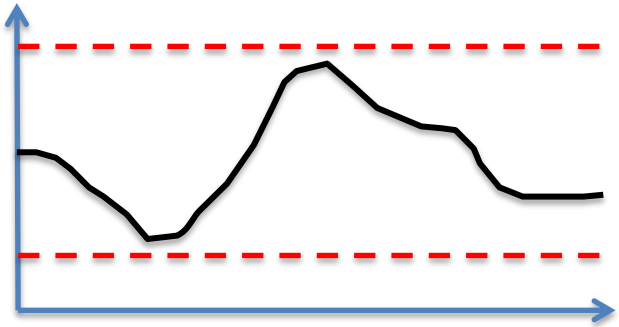


[Image by Ken Butts, Toyota]

wearable continuous glucose monitor

insulin infusion pump

Blood Glucose

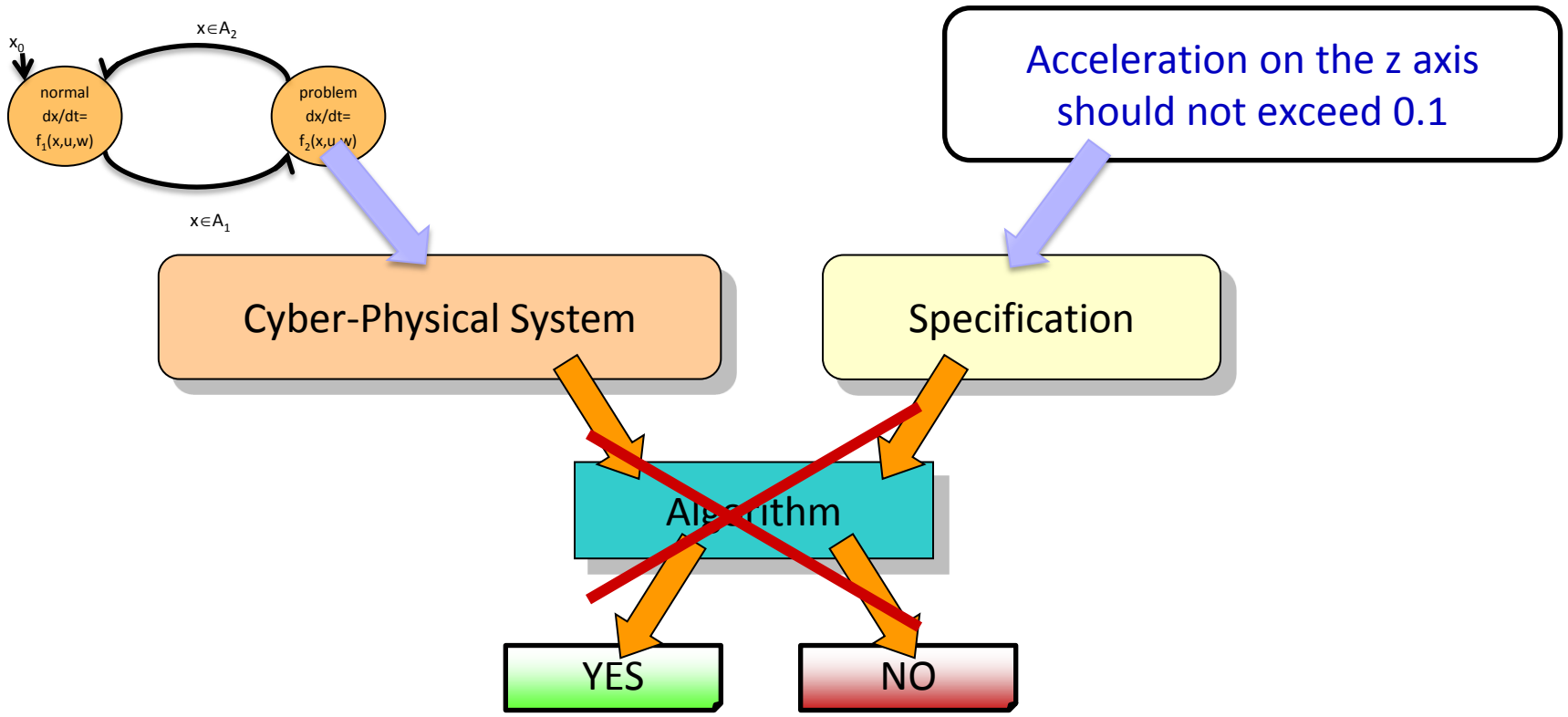


Time

Cyber-Physical Systems

Cyber-Physical Systems (CPS) refers to the class of systems that integrate (engineered) *physical systems* with *computational* and *communication systems*.

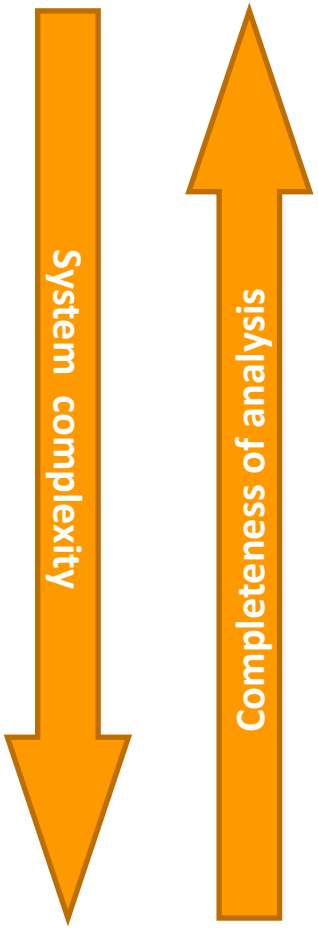
In general, verifying a hybrid system is an **undecidable problem**!



- R. Alur and C. Courcoubetis and N. Halbwachs and T. A. Henzinger and P.-H. Ho and X. Nicollin and A. Olivero and J. Sifakis and S. Yovine, The algorithmic analysis of hybrid systems, TCS
- Henzinger, Kopke, Puri, Varaiya, What's decidable about hybrid automata? Proceedings of the twenty-seventh annual ACM symposium on Theory of computing.

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



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)

until *release*

Syntax: $\Phi ::= \mathrm{T} \mid \mathrm{F} \mid p \mid \neg p \mid \Phi_1 \vee \Phi_2 \mid \Phi_1 \wedge \Phi_2 \mid \Phi_1 \mathcal{U}_I \Phi_2 \mid \Phi_1 \mathcal{R}_I \Phi_2$

I can be of any non-empty bounded or unbounded interval of R^+

Derived operators:

Eventually (in the future) $\mathrm{F}_I \Phi := \mathrm{T} \mathcal{U}_I \Phi$

Always (globally) $\mathrm{G}_I \Phi := \mathrm{F} \mathcal{R}_I \Phi$

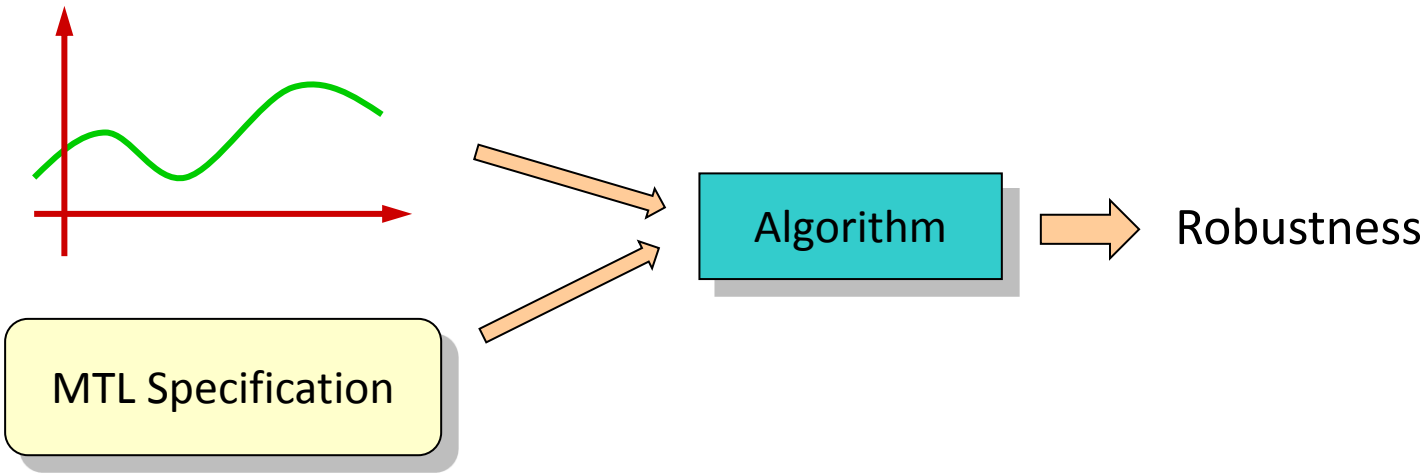
Koymans '90, *Specifying real-time properties with metric temporal logic*

Robustness of MTL formulae

Robustness is a functional that associates a real extended number to each sample path $\boldsymbol{y}$ of $\boldsymbol{Y}$.

positive robustness → signal satisfies the formula

negative robustness → signal falsifies the formula



Robustness of MTL formulae

Robustness is a functional that associates a real extended number to each sample path $\boldsymbol{y}$ of $\boldsymbol{Y}$.

positive robustness → signal satisfies the formula

negative robustness → signal falsifies the formula

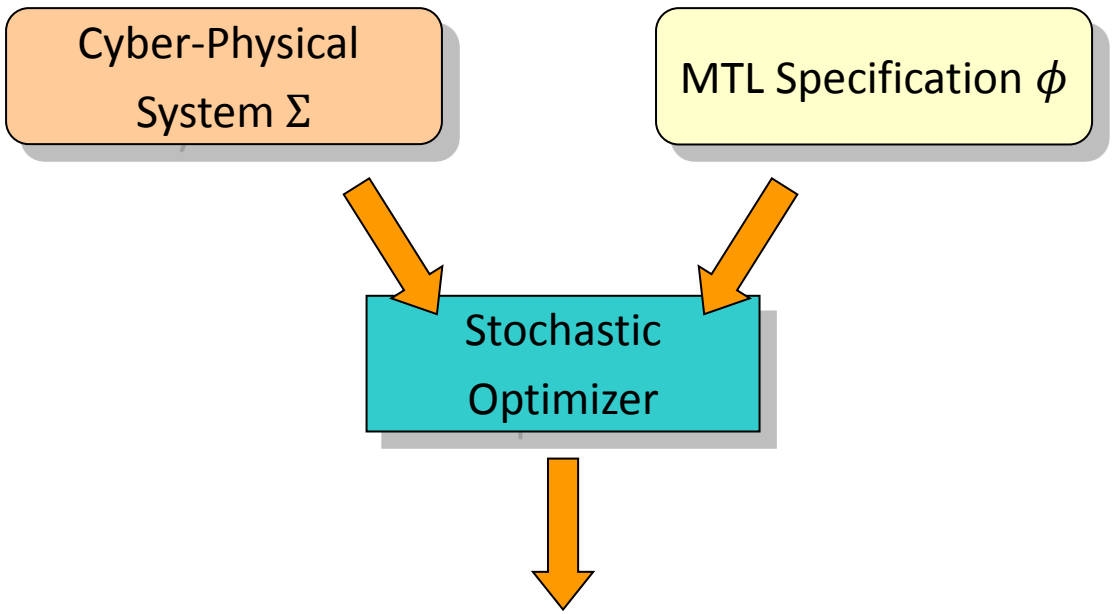
$$\rho_\varphi : y(\cdot, \omega; \theta) \mapsto \rho_\varphi(y(\cdot, \omega; \theta)) \equiv \rho_\varphi(\omega, \theta) \in [-\infty, \infty]$$

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

$$U(\theta) = \mathbb{E}_P[\rho_\varphi(\omega, \theta)] = \int_\Omega \rho_\varphi(\omega, \theta) dP(\omega)$$

Falsification

The falsification method searches for counterexamples that prove that the system does not satisfy the specification



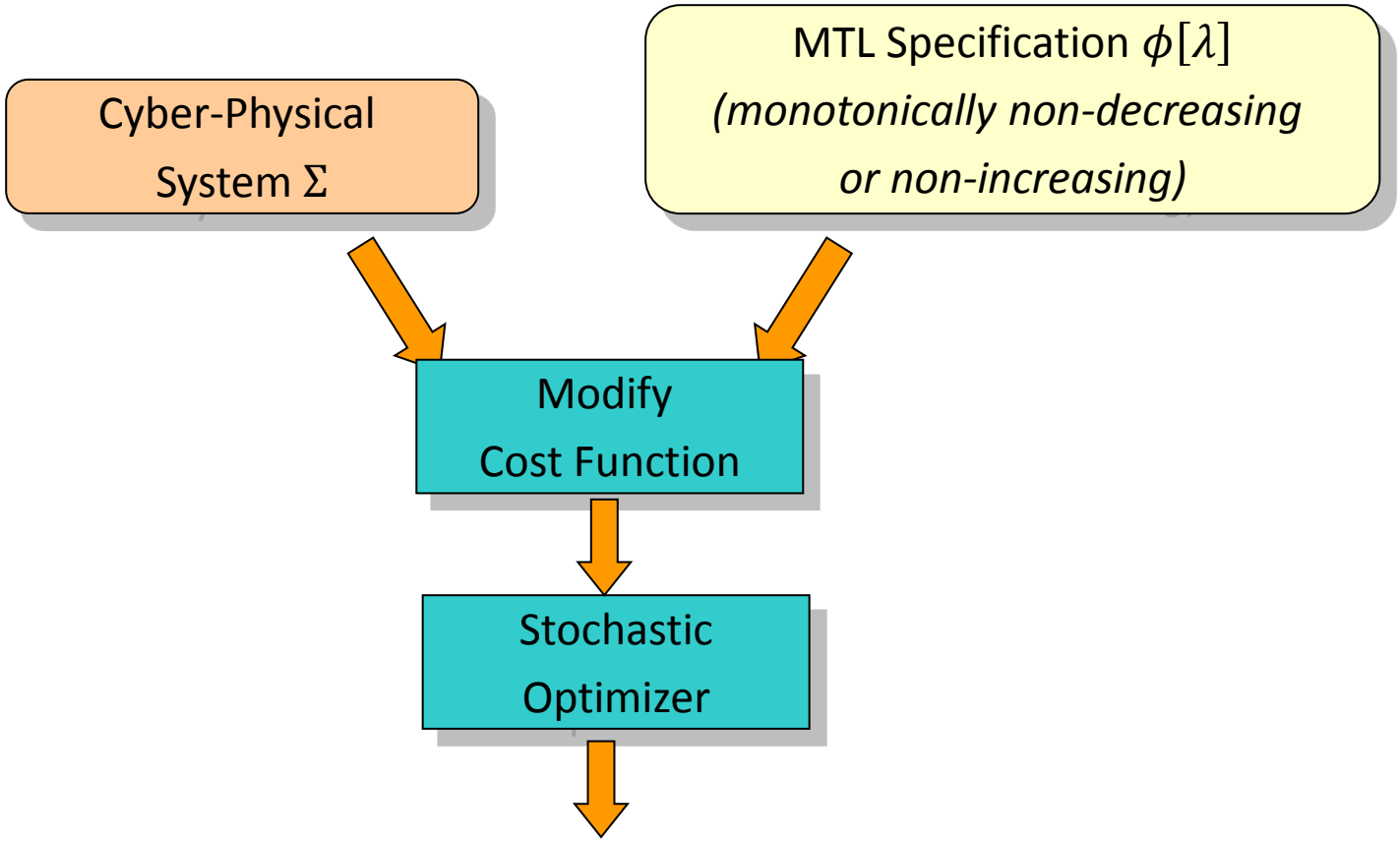
Minimum Robustness

with corresponding input signal and initial conditions

Abbas, et al, Probabilistic Temporal Logic Falsification of Cyber-Physical Systems, ACM TECS 2013

Parameter Estimation

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

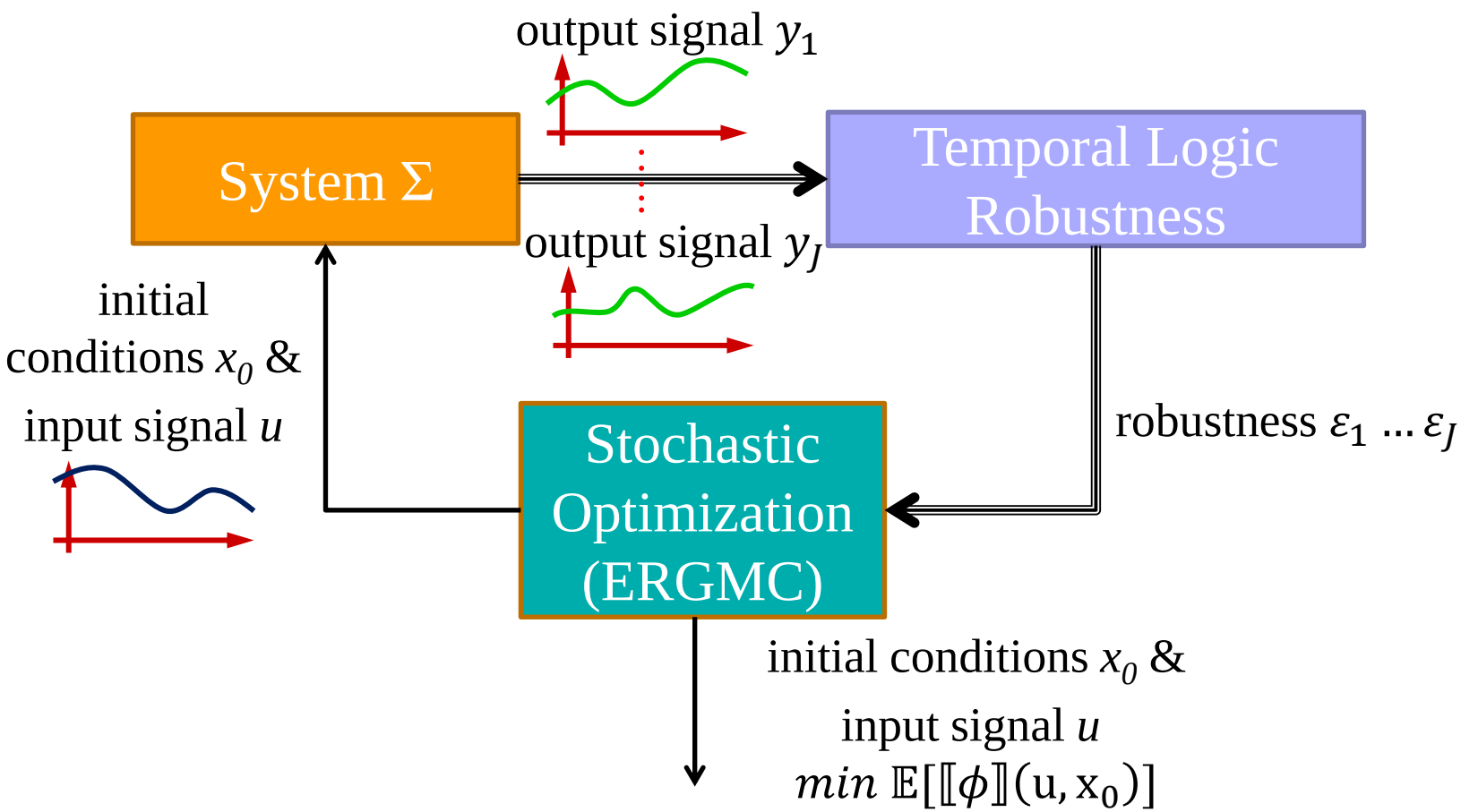


Parameter range for which the systems does not satisfy the specification

Yang, et al, Querying Parametric Temporal Logic Properties on Embedded Systems, ICTSS 2012

Expected Robustness for Stochastic Systems

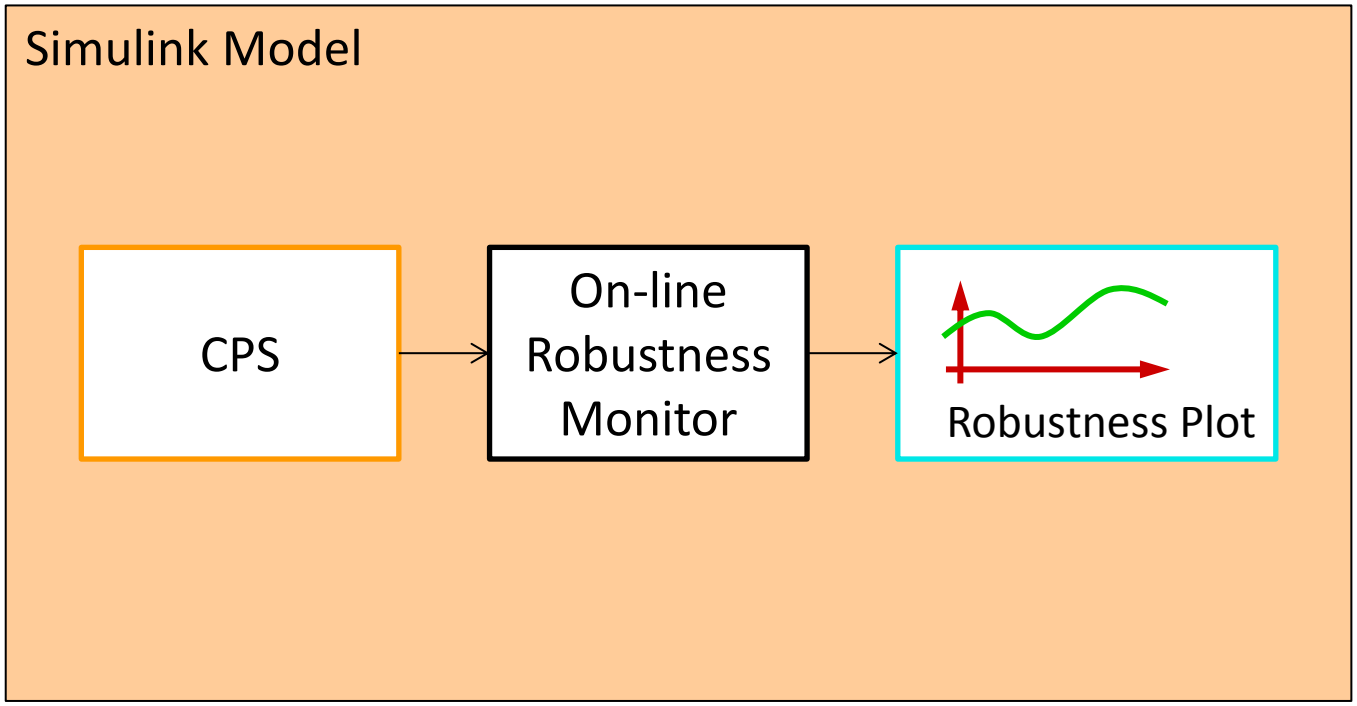
A verification method that enables system engineers that a stochastic system is robust and operates within set specifications.



Abbas. et al, Robustness-Guided Temporal Logic Testing and Verification for Stochastic Cyber-Physical Systems, CYBER 2014

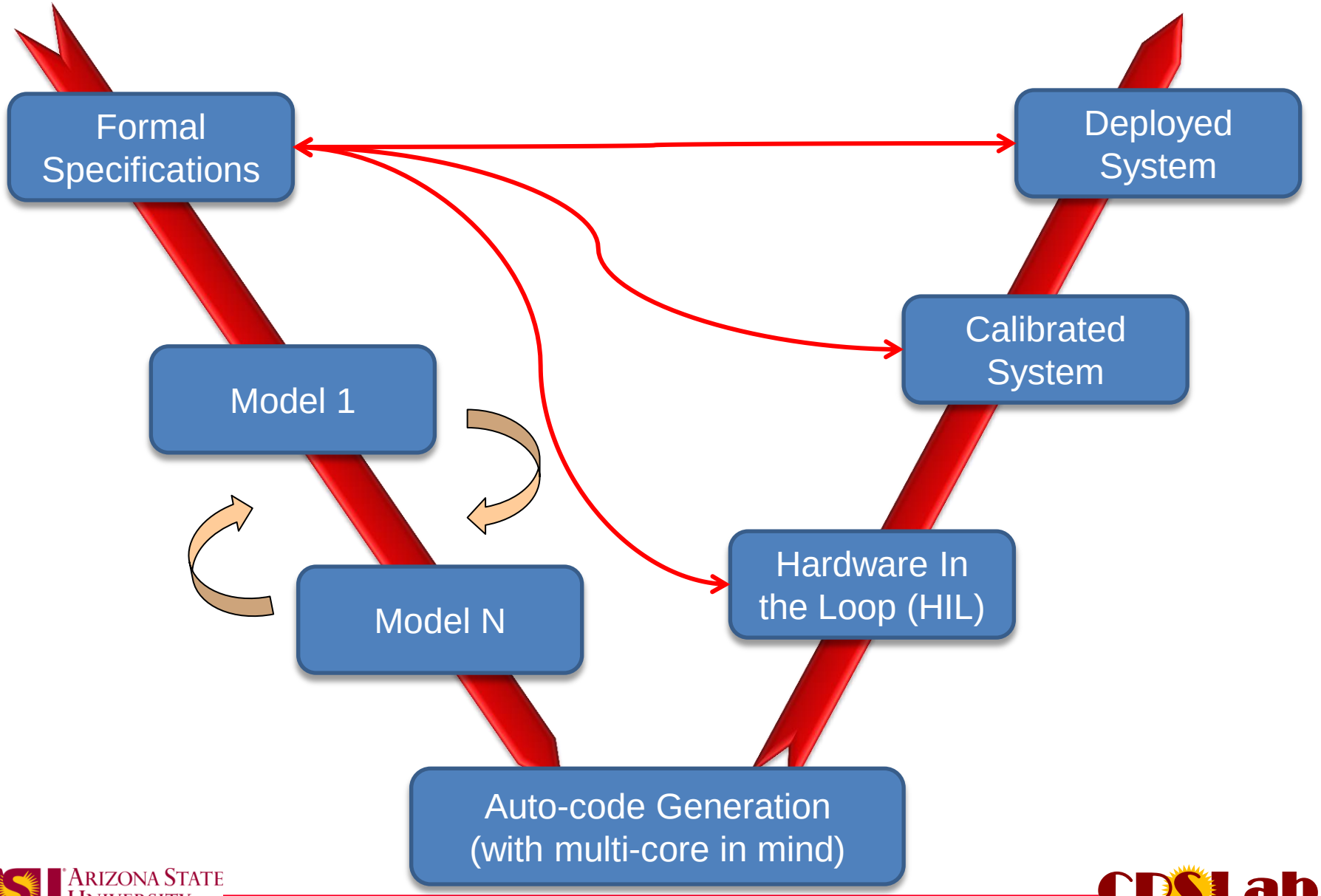
Runtime Verification

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.



Dokhanchi, et al, On-Line Monitoring for Temporal Logic Robustness, RV 2014

Conformance Testing



Problem statement

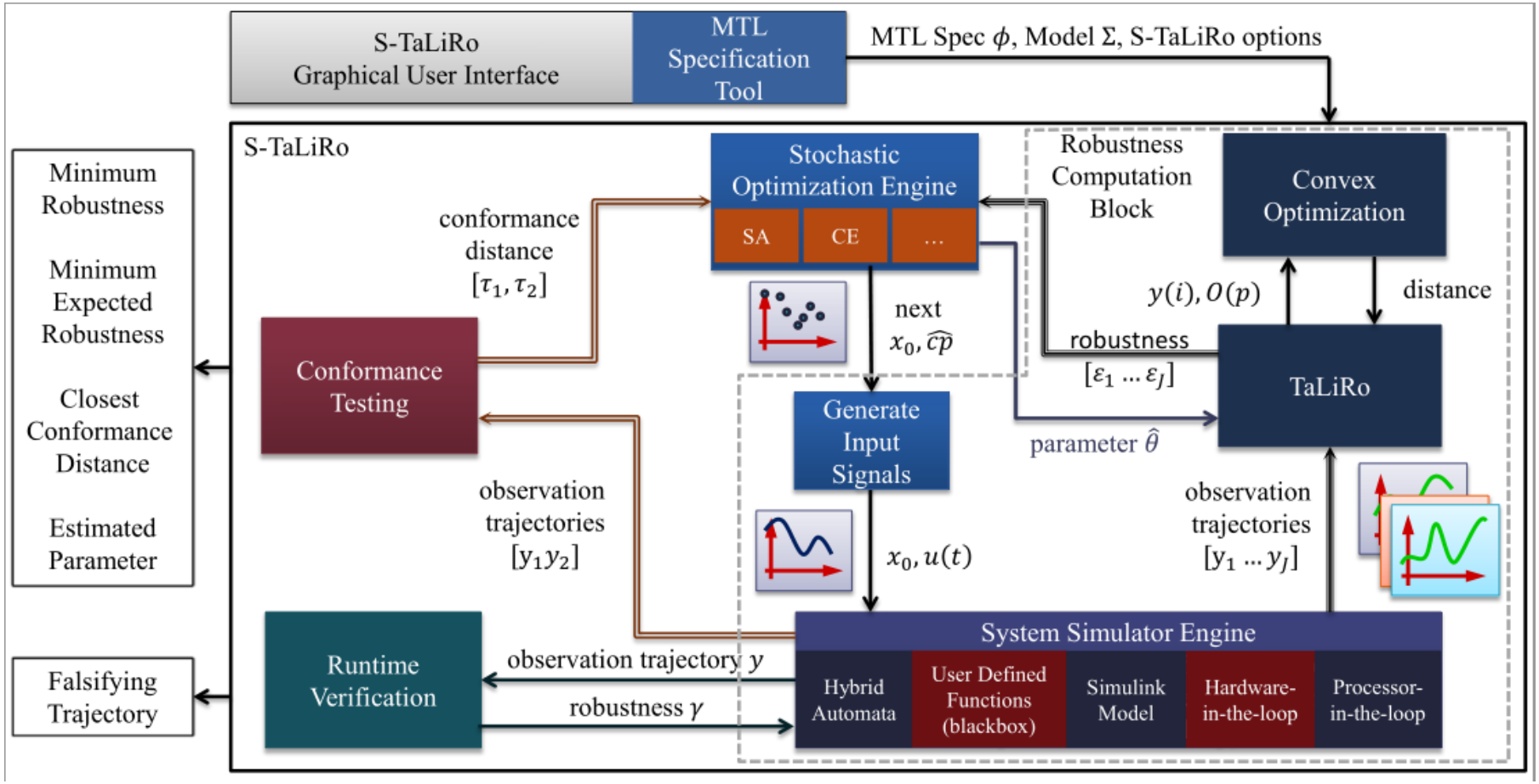
Given two (possibly black-box) systems H1 and H2:

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

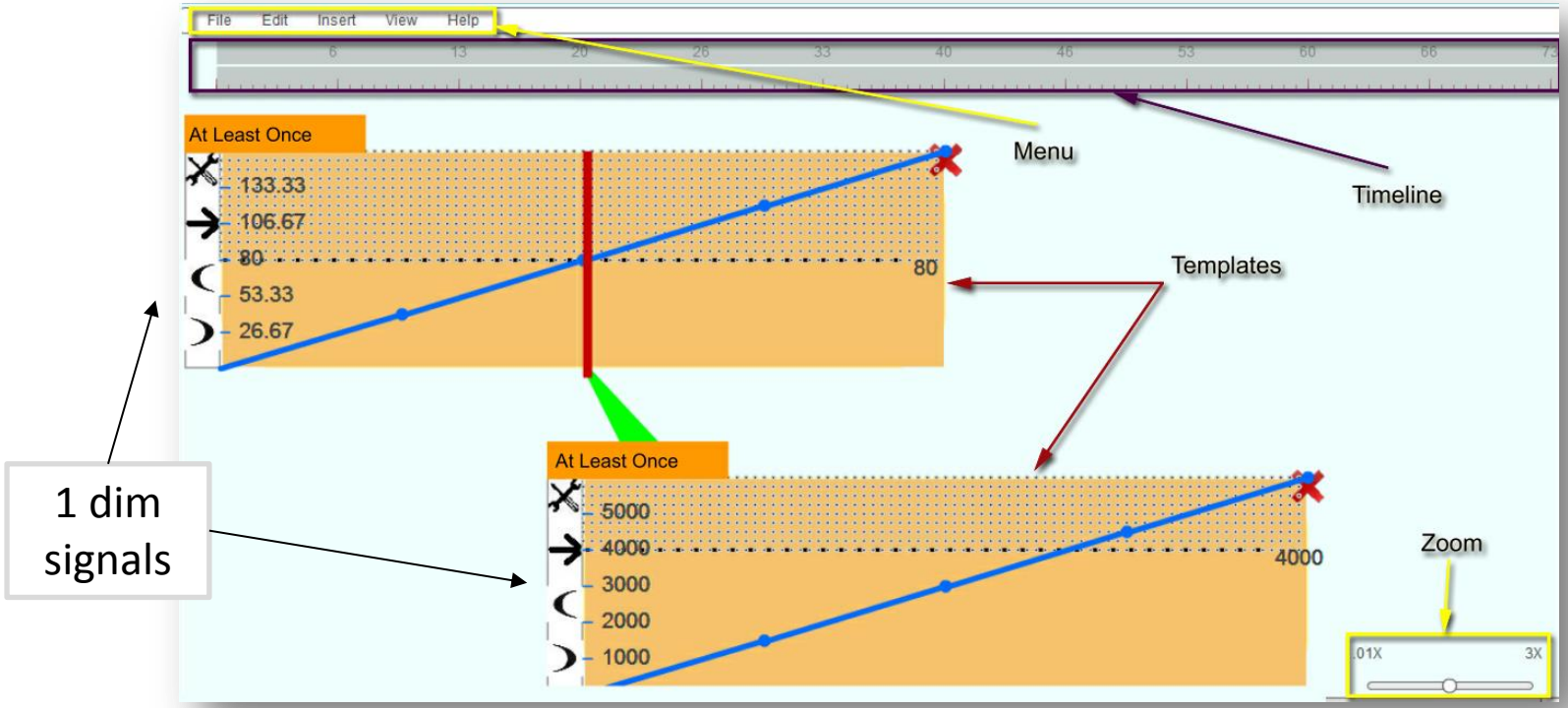


Visual Specification Tool

- 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

Visual Specification Tool



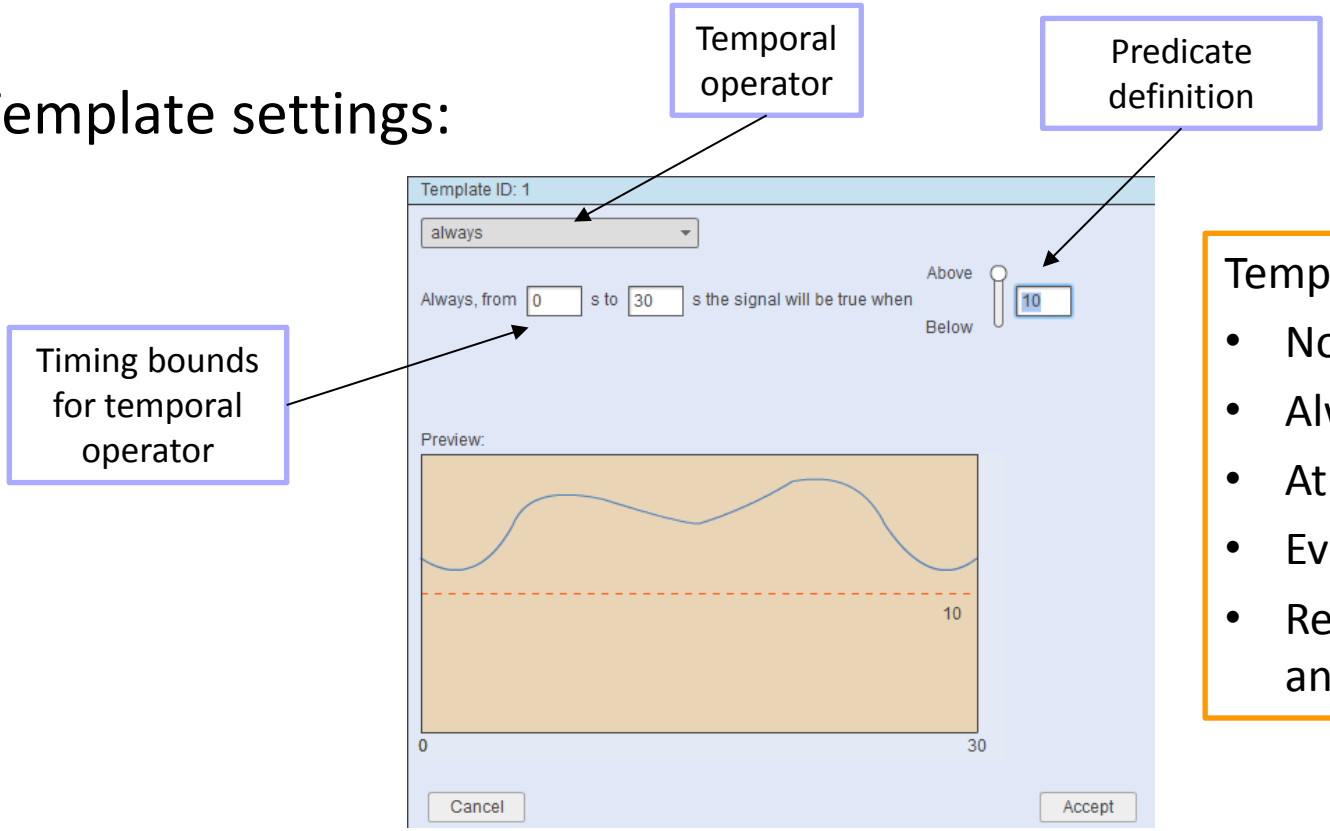
$$\phi = \Diamond_{[0,40]}((speed > 80) \rightarrow \Diamond_{[0,40]}(rpm > 4000))$$

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.

Visual Specification Tool

Specifications defined by templates and relationships between them

Template settings:

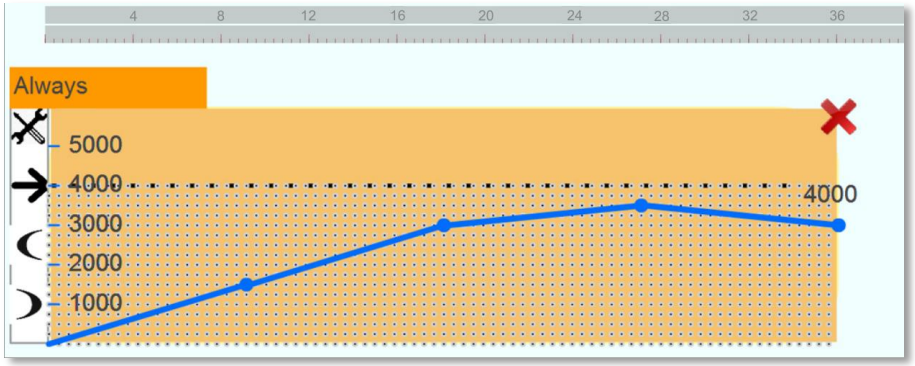


Temporal operators:

- Now
- Always
- At least once
- Eventually Always
- Repeatedly often and finally

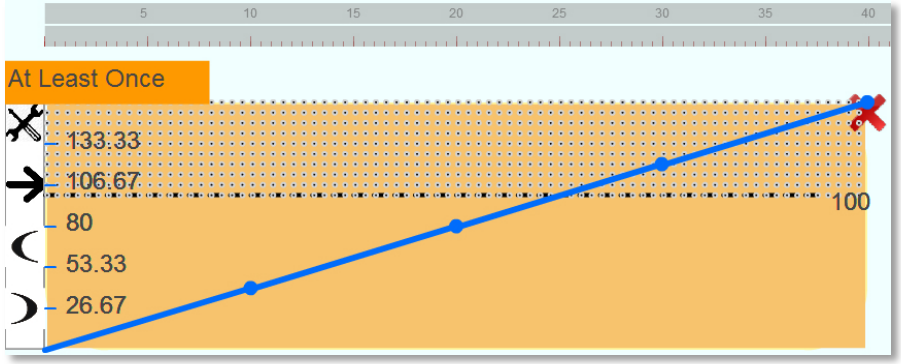
Visual Specification Tool

Example 1:



$$\Box_{[0,36]}(rpm < 4000)$$

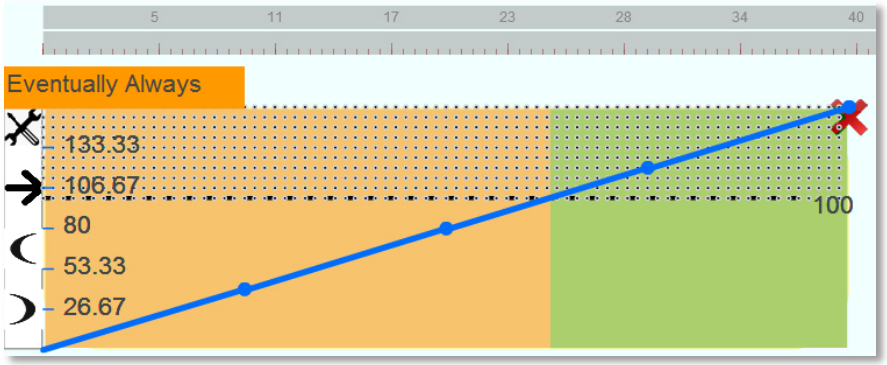
Example 2:



$$\Diamond_{[0,40]}(speed > 100)$$

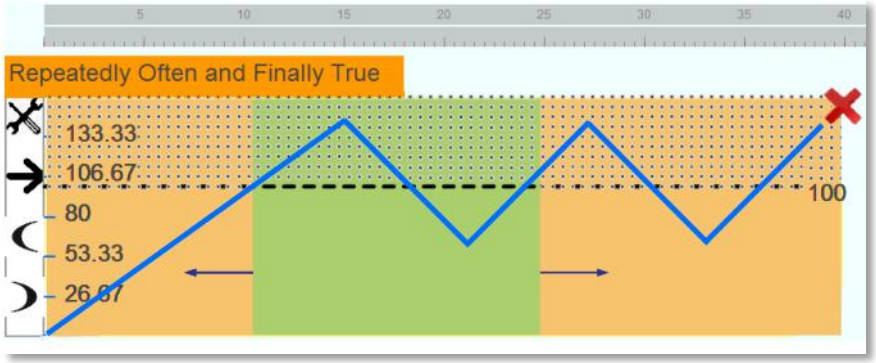
Visual Specification Tool

Example 3:



$$\Diamond_{[0,30]}\Box_{[0,10]}(speed > 100)$$

Example 4:

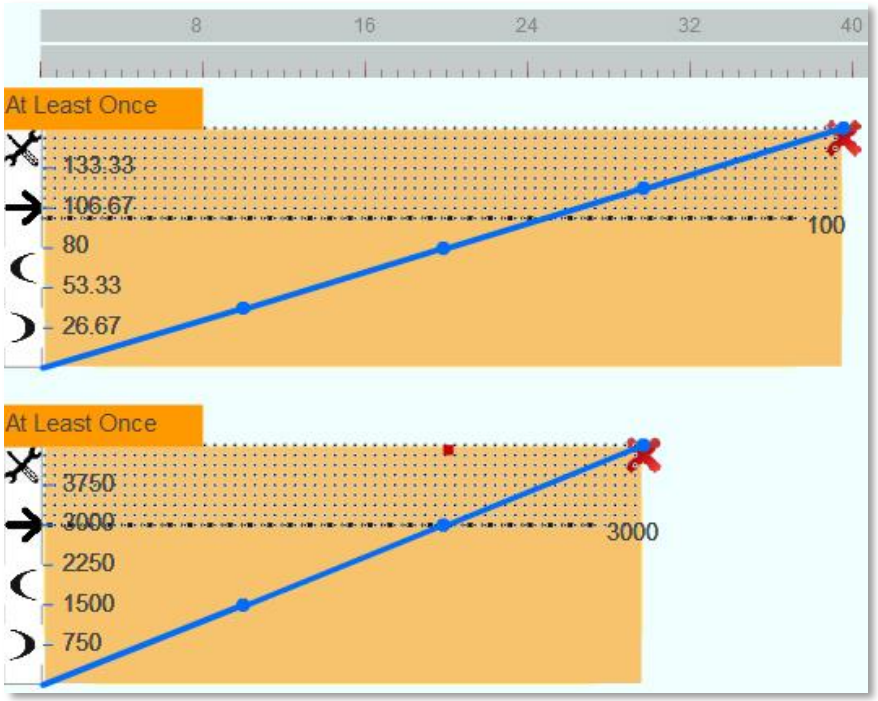


$$\Box_{[0,40]}\Diamond_{[0,13]}(speed > 100)$$

Visual Specification Tool

Example 5:

Implication

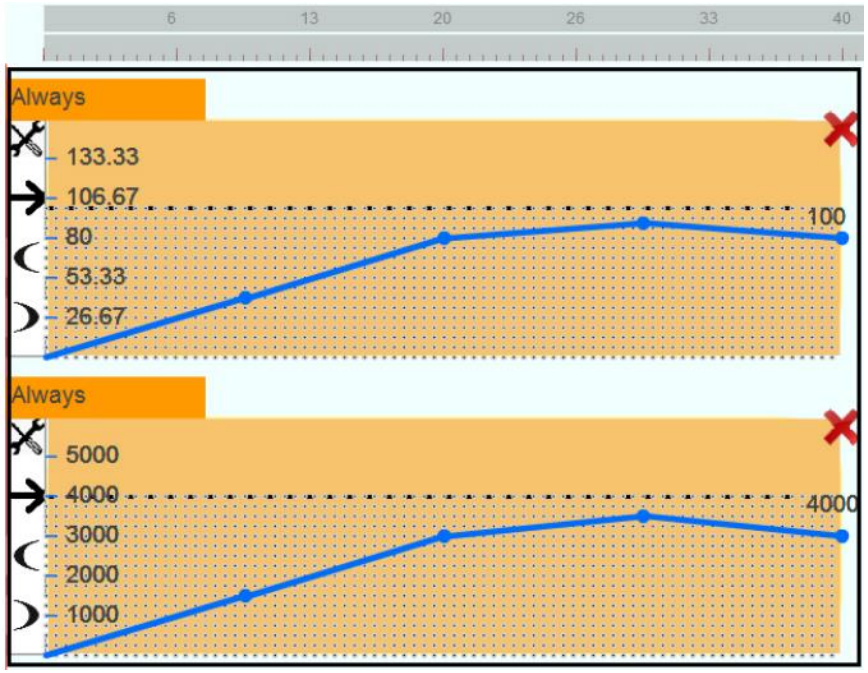


$$(\Diamond_{[0,40]}(speed > 100)) \rightarrow (\Diamond_{[0,30]}(rpm > 3000))$$

Visual Specification Tool

Example 6:

Conjunction



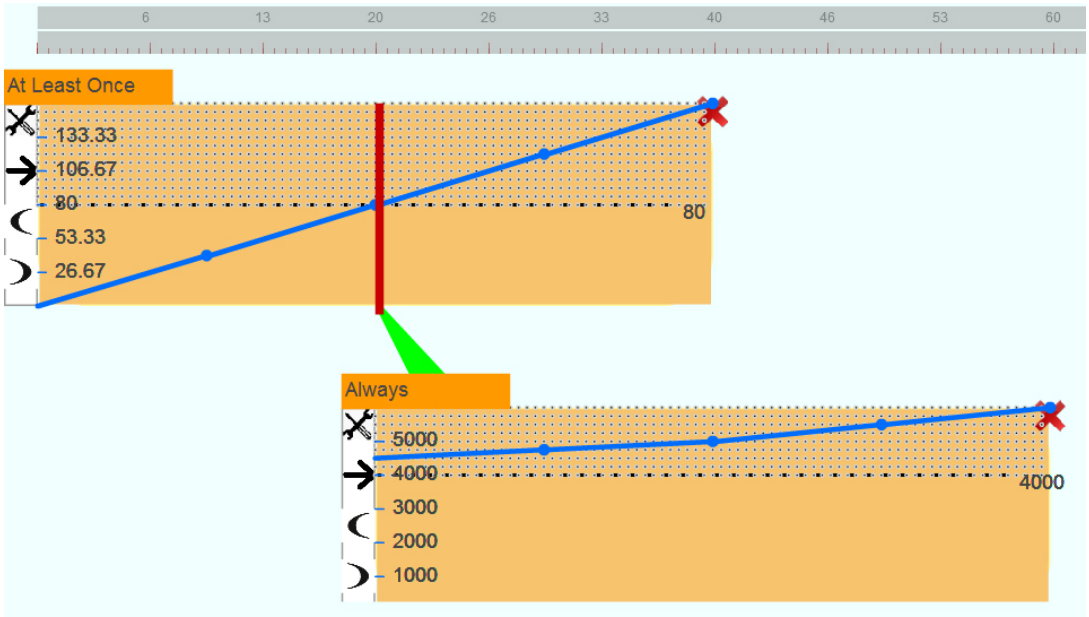
$$(\Box_{[0,40]}(speed < 100)) \wedge (\Box_{[0,40]}(rpm < 4000))$$

Visual Specification Tool

Example 7:

Nested

sequence



$$\Diamond_{[0,40]}((speed > 80) \rightarrow \Box_{[0,40]}(rpm > 4000))$$

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 asses 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

NSF

TOYOTA

CENTER FOR EMBEDDED SYSTEMS
an NSF Industry/University Cooperative Research Center

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