Robustness Guided Testing and Verification for Cyber-Physical Systems

PI: Georgios Fainekos

School of Computing, Informatics and Decision System Engineering

Arizona State University

fainekos at asu dot edu

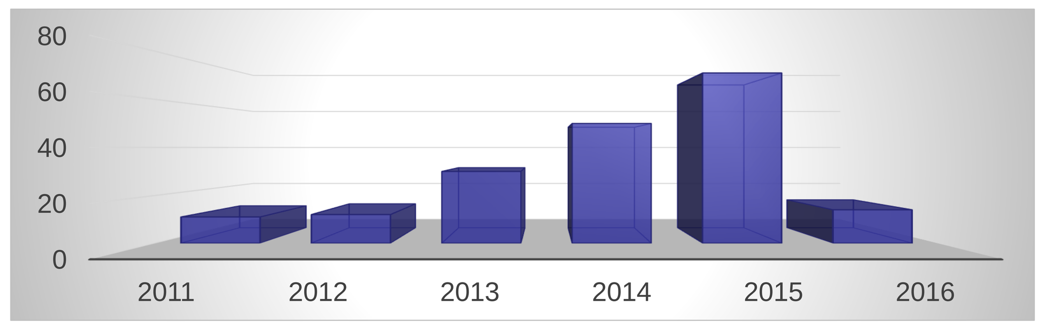
http://www.public.asu.edu/~gfaineko

NSF Award # 1350420

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Motivation

- The complexity of the software in Cyber-Physical Systems (CPS) is increasing almost exponentially with time.
- **Challenge:** The recent **multiple software related recalls** of automobiles and medical devices indicate that the current software development methods may be inadequate for safety critical software applications.



Automotive software recalls

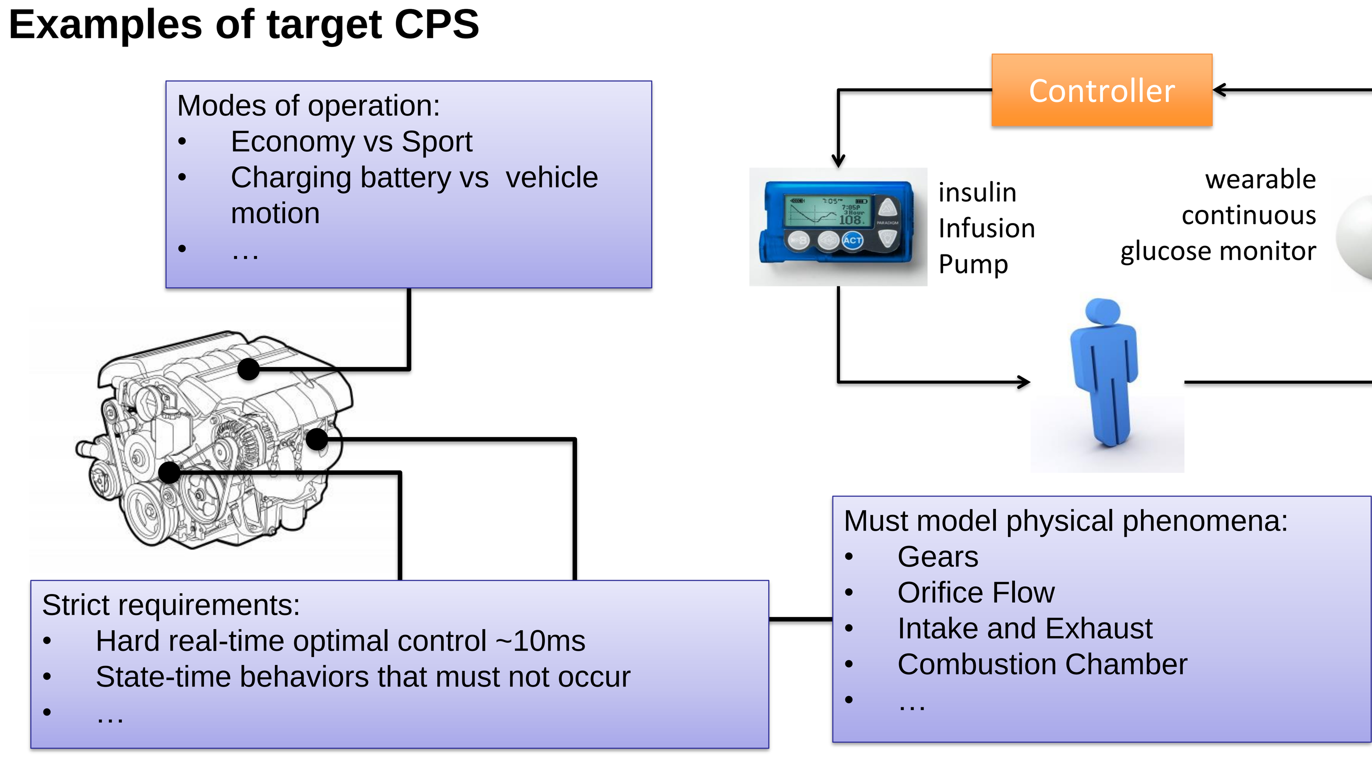
Source: J.D. Power with NHTSA data (2016)

- **Model-Based Design (MBD)** has proven to be a viable approach to tame the complexity of developing software, especially, for CPS. However, testing for CPS still remains an ad-hoc process.
- **Formalizing requirements for known safety critical software recalls**

Requirement	Formalization
When in 5th gear and RPM drops below x, then the system should always switch from 5th to 4th gear.	$G((g=5 \wedge \omega<x) \rightarrow F_{[0,1]}\, g=4)$
The engine should never stall while idle.	$G(idle \rightarrow \omega>1100\ RPM)$
The electric motor should always rotate in the direction selected by the transmission.	$G((g\geq 1 \wedge \text{"other"} \rightarrow \omega_{em}>0)$
The cruise control should always disengage when the "turn off" button is pressed.	$G(turnoff \rightarrow F_{[0,1]}\, cc=off)$

G: Always (Globally) F: Eventually (Future)

- **Examples of target CPS**



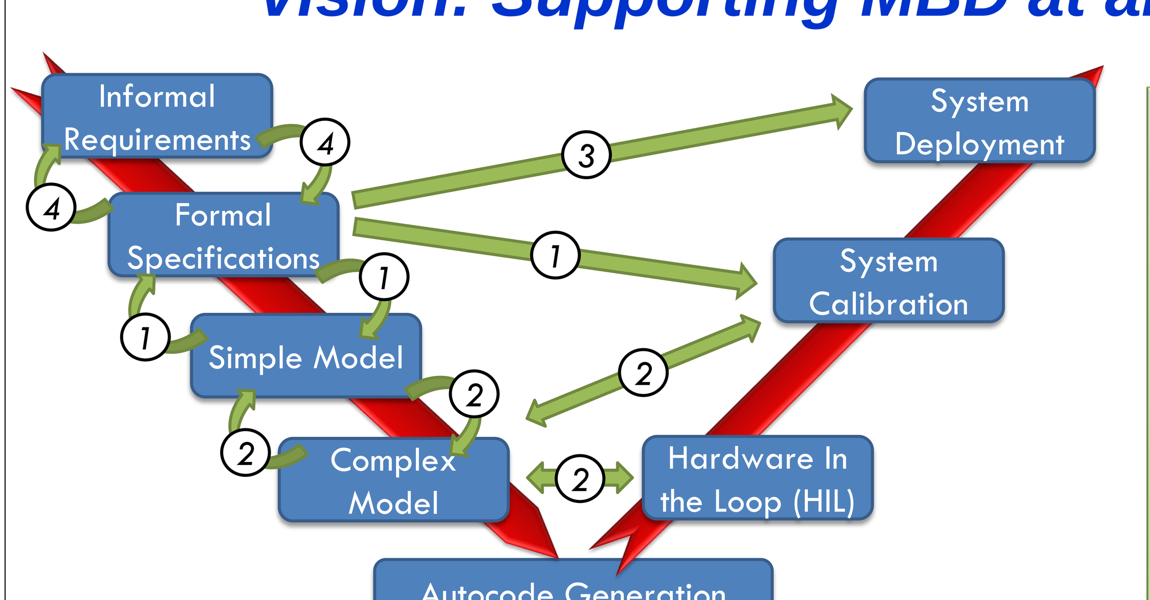
Project Summary

This project develops a theoretical framework as well as software tools to support testing and verification of CPS within a Model-Based Design (MBD) process.

The project's research comprises three components:

1. *development of conditions on the algorithms and on the structure of the CPS for inferring finite-time guarantees on the randomized testing process;*
2. *the study of testing methods that can support modular and compositional system design; and*
3. *investigation of appropriate notions of conformance between two system models and between a model and its implementation on a computational platform.*

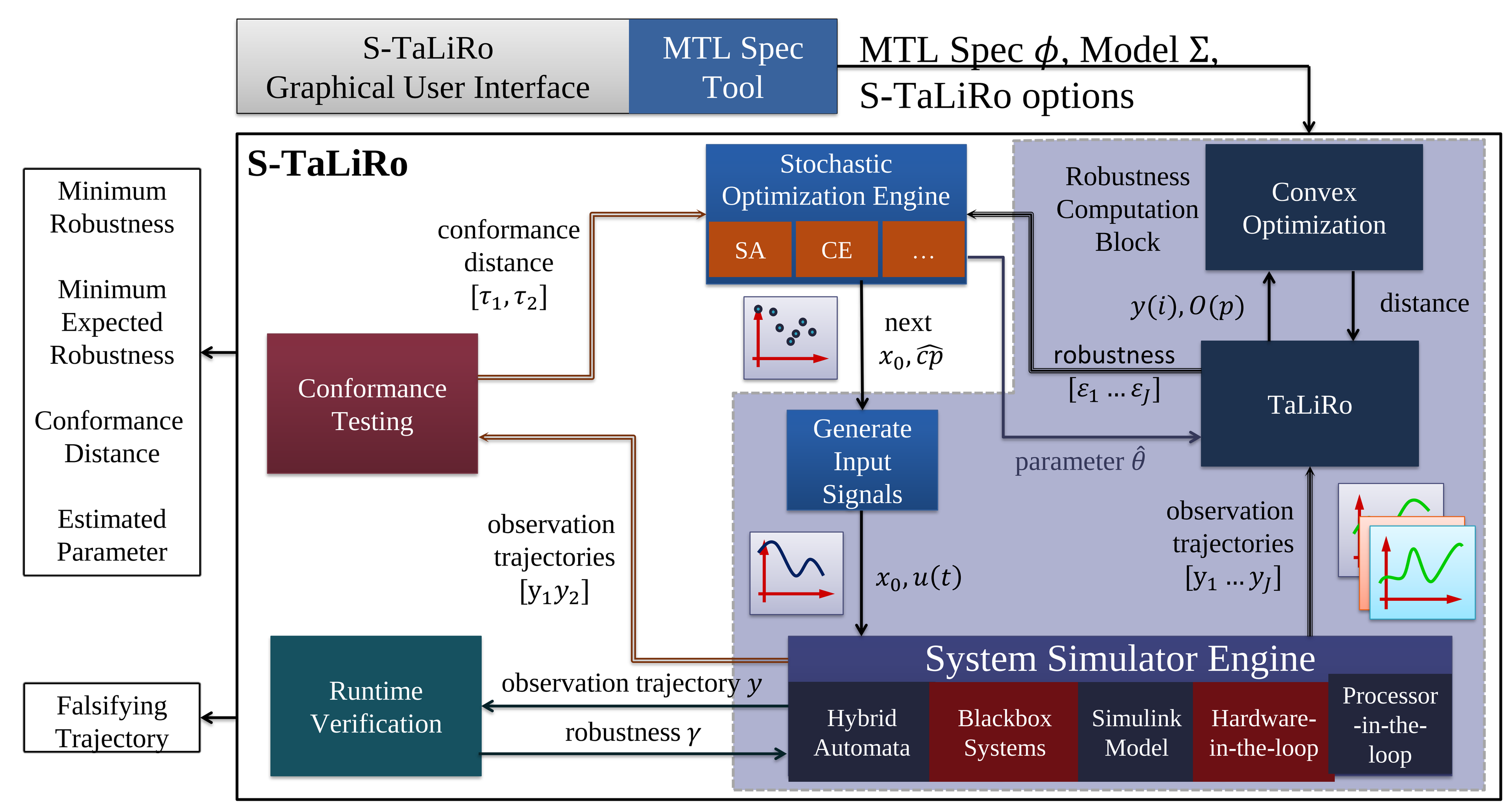
Vision: Supporting MBD at all stages



1. Testing formal specifications and specification mining
2. Conformance testing: models, HIL/PIL or tuned/calibrated model
3. Runtime monitoring of formal requirements
4. Specification analysis, visualization and debugging (inconsistencies, redundancy, etc.)

S-TaLiRo Tool Suite

All the results are / will be implemented in the S-TaLiRo Tool Suite

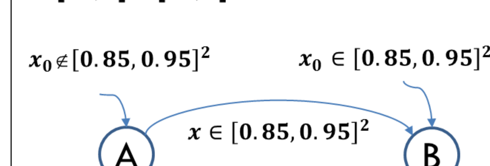


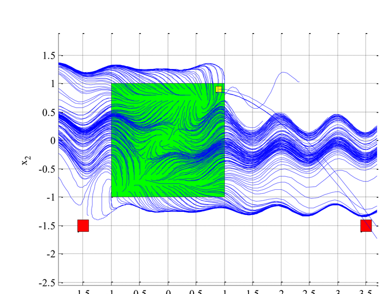
More information www.tinyurl.com/Staliro

Falsification with model & code coverage

System:
If $x_0 \in [-1,1]^2 \backslash [0.85,0.95]^2$ then
$\dot{x} = x - y + 0.1t$
$\dot{y} = y\cos(2\pi y) - x\sin(2\pi x) + 0.1t$
Else
$\dot{x} = x$
$\dot{y} = -x + y$

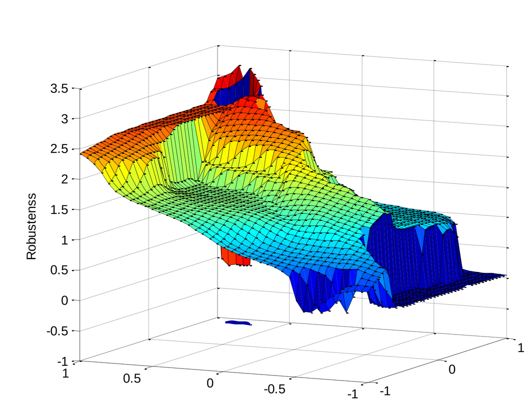
Initial conditions:
[-1,1] x [-1,1]

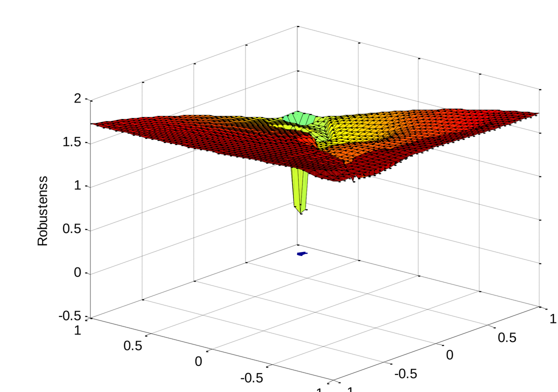


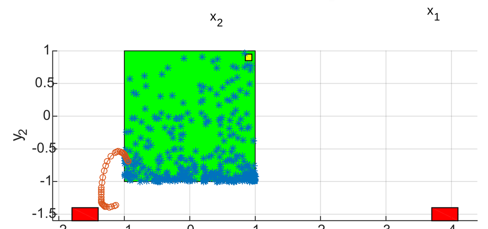


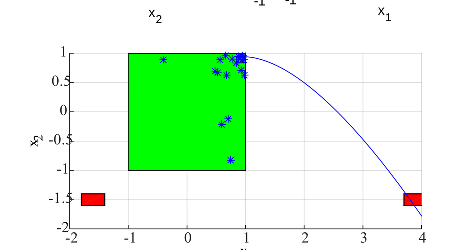
Specification:
$G_{[0,2]}\neg a \wedge G_{[0,2]}\neg b$
where
O(a) = [-1.6,-1.4] x [-1.6,-1.4]
O(b) = [3.4,3.6] x [-1.6,-1.4]

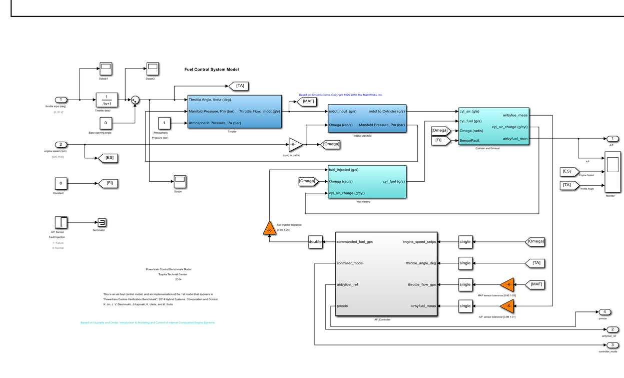
Specification:
$G_{[0,2]}\neg a \wedge G_{[0,2]}\neg b$
where
O(a) = [-1.6,-1.4] x [-1.6,-1.4] x {B}
O(b) = [3.4,3.6] x [-1.6,-1.4] x {B}





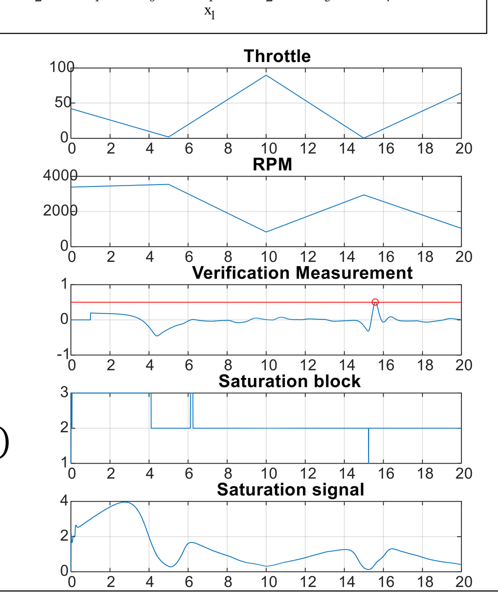






Air Fuel Control Model from X. Jin et al "Powertrain Control Verification Benchmark", HSCC 2014

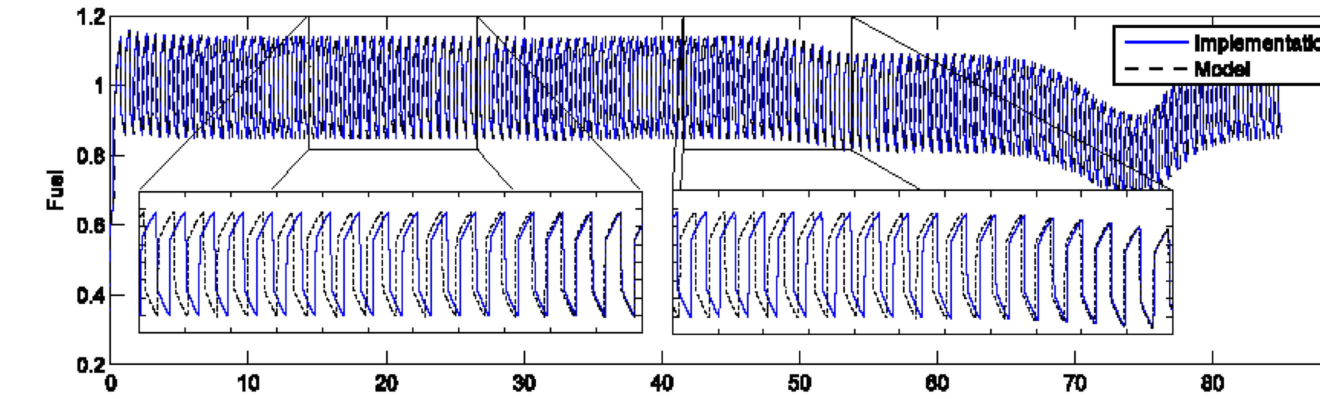
- Uniform random sampling: No falsification after 500,000 tests
- S-Taliro with simulated annealing sampling I falsification after 4,982 tests
- Spec: $G_{[0,20]}(VM \leq 0.5) \vee \neg(F_{[0.1,\infty)} LowerSaturation)$
- Robustness Value : 0.5000226



References For a list of relevant background references and technical notes, please contact the PI.

A generic conformance notion

In general, determining that the outputs of the Model and the Implementation are "close enough", i.e., conformant, is application-dependent and relies on expertise.



We propose (T,J,τ,ε)-closeness as a generic conformance notion. This notion is appropriate for continuous-time, discrete-time, and hybrid-time systems.

(T,J,τ,ε)-closeness: Consider two trajectories y, and y' of Σ and Σ', respectively. Given $T>0$, $J>0$, $\tau>0$, and $\varepsilon>0$, we say y and y' are (T,J,τ,ε)–close if:

For all (t,j) in the support of y s.t. $t\leq T$ and $j\leq J$, there exists (s,j) in the support of y', such that $|t-s|<\tau$ and $\|y(t,j)-y'(s,j)\|<\varepsilon$ (and the symmetric notion)

The largest (τ,ε) such that all trajectories of Σ and Σ' are (T,J,τ,ε)–close is the conformance degree between Σ and Σ'.

Benefits of (T,J,τ,ε)-closeness as a generic notion of conformance:

- Only requires the ability to simulate the system – black boxes O.K.
- Can be tested early in the design cycle before all the instrumentation is in place for more targeted testing.
- Captures differences in timing characteristics as well as state values
- Real-valued: can speak of a conformance degree and rank Implementations based on how well they conform to the Model.

Property transfer between (τ,ε)-close systems

Model 1: $y_1 = H_1(x_1,u)$ $\preccurlyeq_{\tau,\varepsilon}$ Model 2: $y_2 = H_2(x_2,u)$

$H_1 \models \varphi'$? $\quad H_2 \models \varphi$

Theorem: If $H_1 \preccurlyeq_{\tau,\varepsilon} H_2$ and $H_2 \models_O \varphi$, then $H_1^\tau \models_{O_\varepsilon} [\varphi]_\tau$

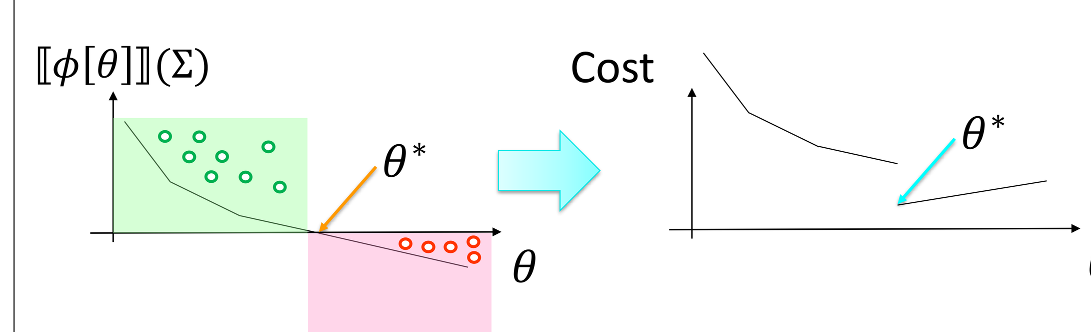
Compositionality

Σ_1, Σ_2 $\approx_{(\tau,\varepsilon)}$ Σ'_1, Σ'_2

$\varphi \dashv \quad \vDash \varphi'$

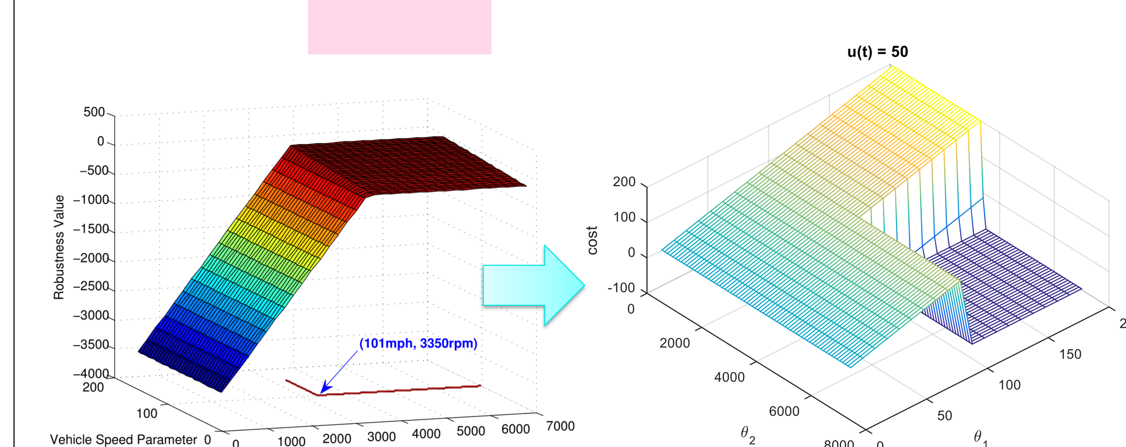
Specification Mining

Given a parametric MTL formula $\phi[\vec{\theta}]$ with a vector of m unknown parameters and a system Σ, find the set $\Psi = \{\theta^* \in \Theta \mid \Sigma \nvDash \phi[\theta^*]\}$



Non-Increasing robustness with respect to θ

$$\min_{\theta\in\Theta}\min_{\mu\in\mathcal{L}_\tau(\Sigma)} \left(f(\theta) + \begin{cases} \gamma + [\![\phi[\theta]]\!](\mu) & \text{if } [\![\phi[\theta]]\!](\mu)\geq 0 \\ 0 & \text{otherwise}\end{cases}\right)$$



Non-Decreasing robustness with respect to $f(\vec{\theta})$

$$\max_{\theta\in\Theta}\max_{\mu\in\mathcal{L}_\tau(\Sigma)} \left(f(\theta) + \begin{cases} \gamma - [\![\phi[\theta]]\!](\mu) & \text{if } [\![\phi[\theta]]\!](\mu)\geq 0 \\ 0 & \text{otherwise}\end{cases}\right)$$

In case of multiple parameters to be mined, we have "one-sided" Pareto front.

Specification Debugging

Questions:

- Are we checking the correct requirements?
- Are the auto-generated test cases meaningful?