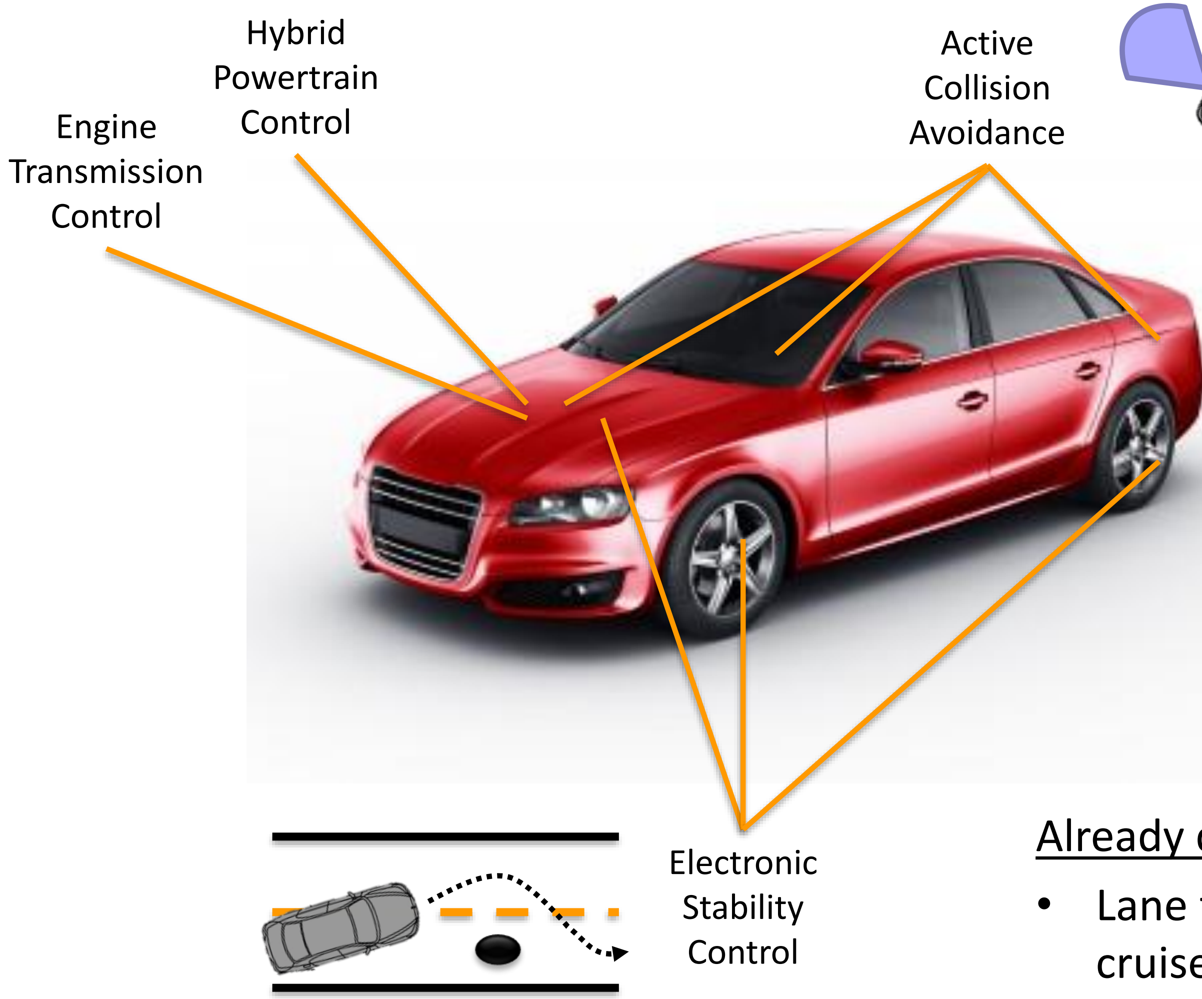
Specification guided testing and verification for Cyber-Physical Systems

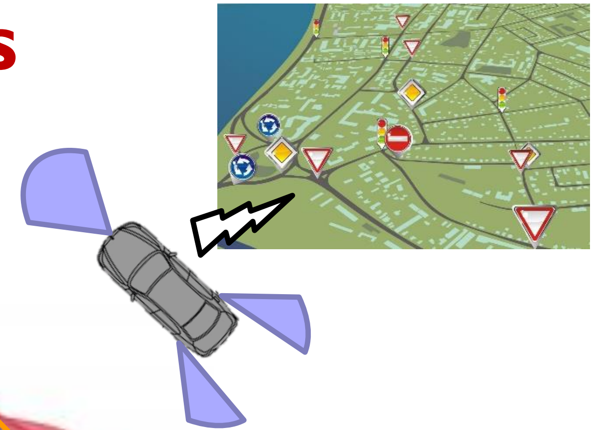
Georgios Fainekos

7th Halmstad Summer School on Testing, June 2017

fainekos at asu edu
http://www.public.asu.edu/~gfaineko
S-Taliro website: https://sites.google.com/a/asu.edu/s-taliro/

Modern Vehicles

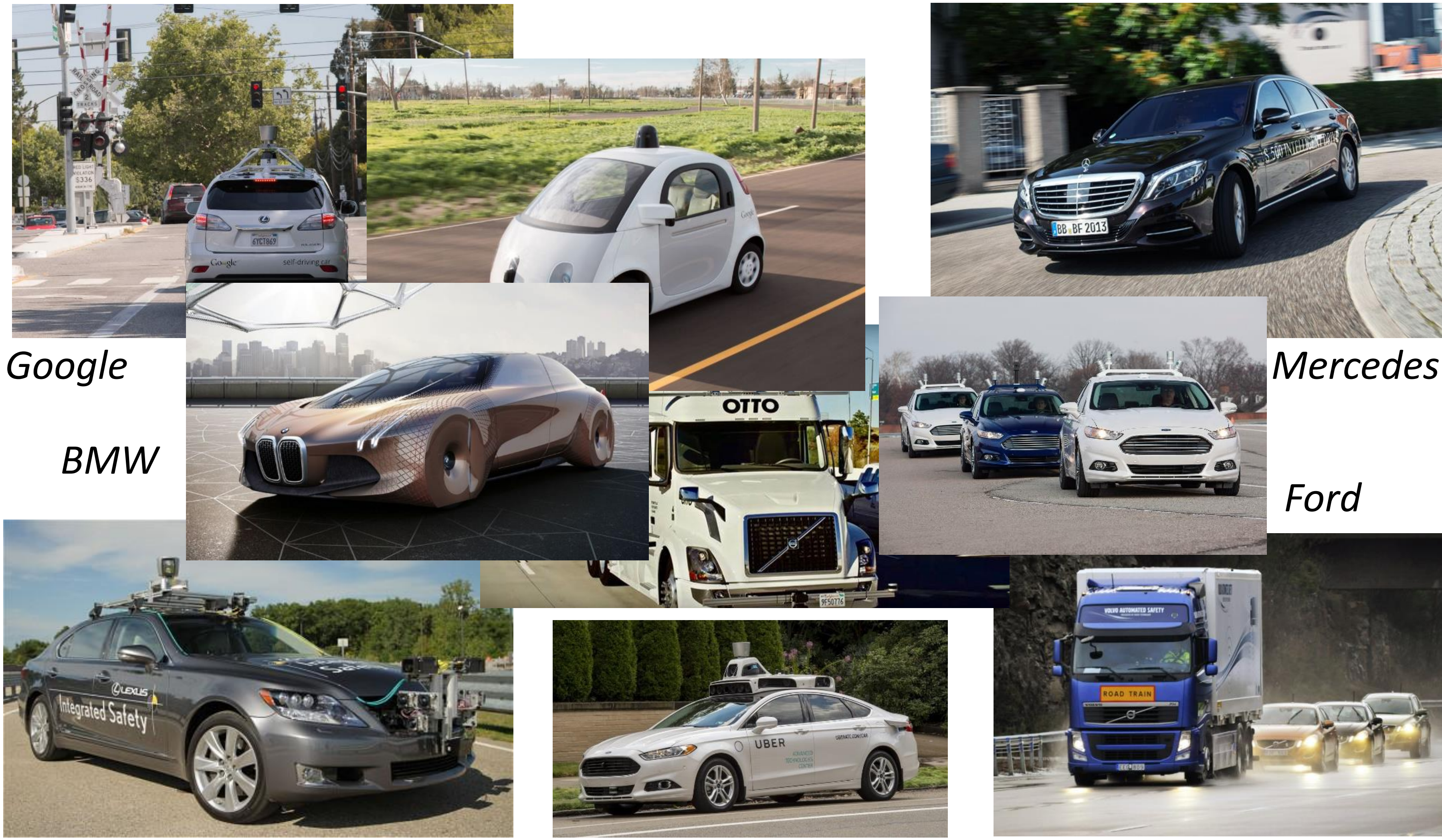




<u>Already demonstrated:</u>

- Lane following & Active cruise control
- Fully autonomous driving
- …

Autonomous cars are almost here!



Google

BMW

Mercedes

Ford

Toyota

Uber

Volvo

Trust? : Sampling of automotive recalls (~2011-12) due to software errors …

- "A software error may prevent the transmission from downshifting, such as shifting from 5th to 4t … ry of the problem. "This … ne stall, increasing the risk of a crash."

No downshifting from 5^{th} to 4^{th}

- … the software that "allows the ECU to establish a 'handshake' with the engine is in error. The EC … gine is found to be out of toler … pens, the ECU triggers a fault … tion outside its prescribed tolerances, a rough idle or stalling situation ensues."

Rough idling or stalling due to complicated adaptive ECU

- … to u … tain circums … tor to rotate in the direction opposite to that selected by the transmission.

Electric motor to rotate in the direction opposite to that selected by the transmission

- If the fault occ … e ignition while driving - which … so disables power steerin… Braking or pressing the cancel button will not work.

Cruise control does not disengage unless turning off the ignition

- …

Many more …

How serious this problem is?

The same holds for the medical device industry!

Is it always a software error?!?

https://www.youtube.com/watch?v=qQkx-4pFjus

A Tesla somewhere in Switzerland

Tesla cars: Clearly a marvel of modern engineering!

From the Tesla Model X Owner's manual (Not a bug!):

Warning: Traffic-Aware Cruise Control can not detect all objects and may not brake/decelerate for stationary vehicles, especially in situations when you are driving over 50 mph (80 km/h) and a vehicle you are following moves out of your driving path and a stationary vehicle or object, bicycle, or pedestrian is in front of you instead. Always pay attention to the road ahead and stay prepared to take immediate corrective action. Depending on Traffic-Aware Cruise Control to avoid a collision can result in serious injury or death. In addition, Traffic-Aware Cruise Control may react to vehicles or objects that either do not exist or are not in the lane of travel, causing Model S to slow down unnecessarily or inappropriately.

- Why the engineers cannot guarantee correct operation under all conditions?
- Can you prove / formally verify correctness?
- How do you even test such a system?

Are these just programming errors?!?

Could these be logical / design errors?!?

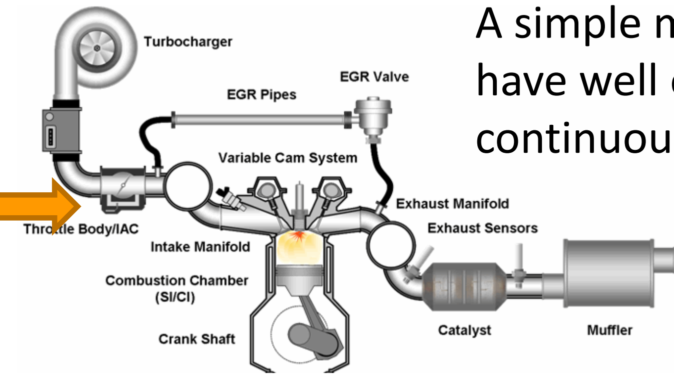
Can we even answer these questions efficiently and effectively?

WHY IS THE PROBLEM CHALLENGING?

Control design for powertrain

Vehicle dynamics & Environment

Engine dynamics



A simple model could have well over 60 continuous state variables.

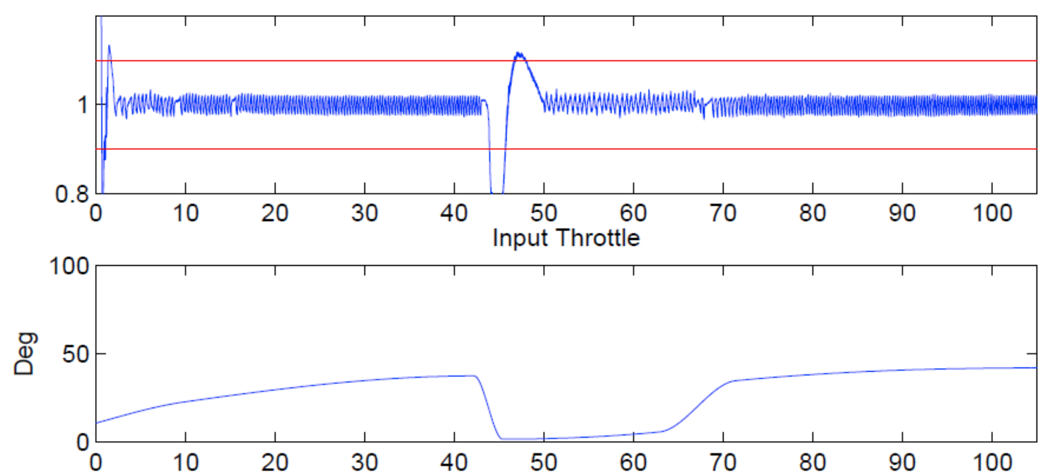
[Image: SimuQuest®]

<u>Requirement:</u> Whenever the normalized air-to-fuel ratio is outside [0.9,1.1], it will settle back inside the range within 1 sec, and stay there for at least 1 sec.

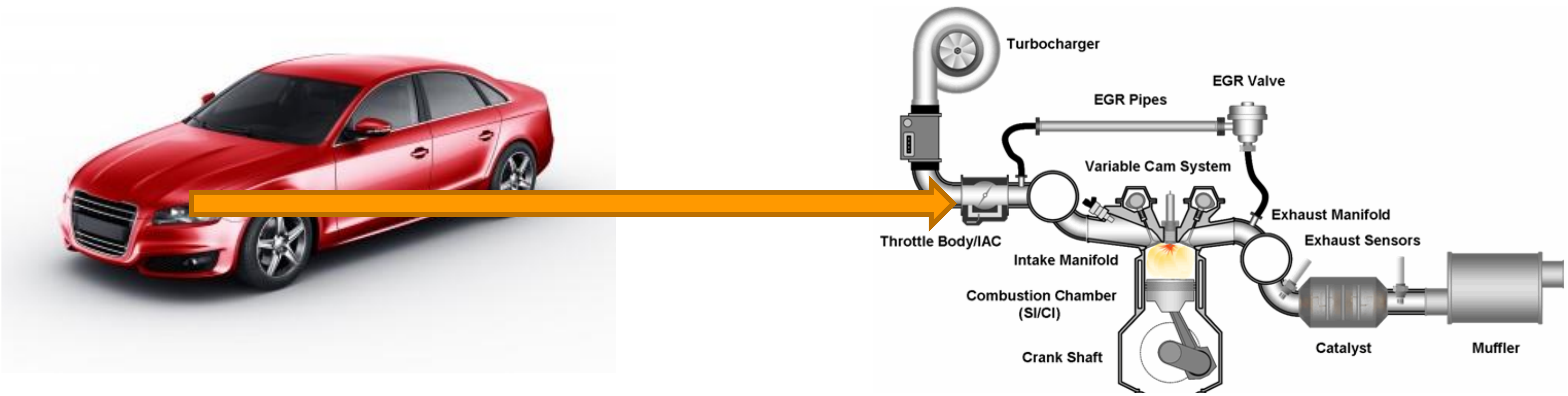
Controller design??

Challenges:

1. Noisy environment & high dim nonlinear dynamics
2. Hard real-time requirements <10ms



Engine models: Complex!



[Image: SimuQuest®]

Enginuity™ Modeling Approach

Orifice Flow

Isentropic Flow Model

$$\dot{m}_1 = A\frac{p}{\sqrt{RT}}\psi$$

$$\psi = \sqrt{\ldots[\max(\ldots) - \max(\ldots)]}$$

⋮

Intake and Exhaust Plenum

Mass Conservation

Energy Conservation

$$\dot{m}_2 = \begin{cases} > 0 & if\ p_1 > p_2 \\ = 0 & if\ p_1 = p_2 \\ < 0 & if\ p_1 < p_2 \end{cases}$$

Combustion Chamber

Energy Conservation

Heat Transfer

Heat Release

Ignition Delay

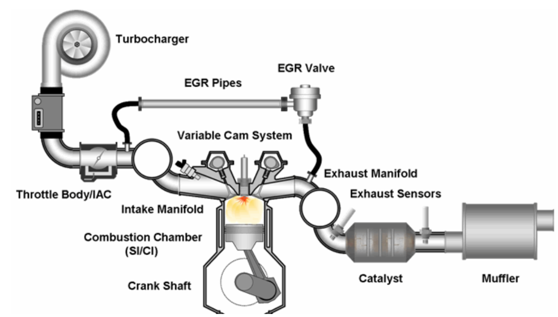
Fuel Injection Dynamics

⋮

…

Develop controllers and generate code

Engine dynamics



[Image: SimuQuest®]

Simplify model:

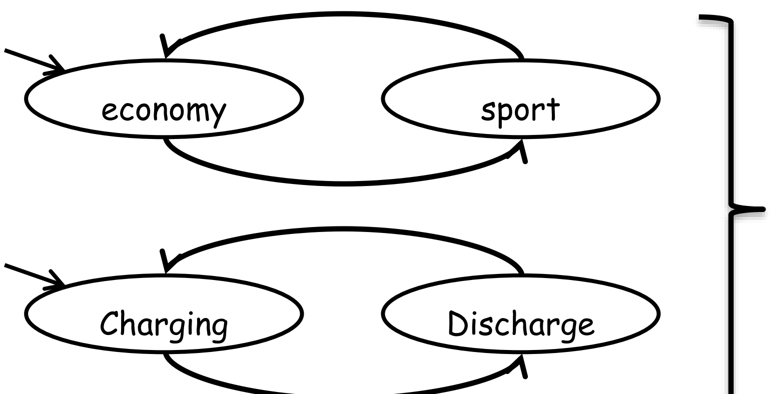
$$\dot{x} = Ax + Bu$$

or

$$\dot{x} = f(x,u), \#(x) \ll 60$$

Alternative path: PID tuning

Design control laws
e.g. idle speed control

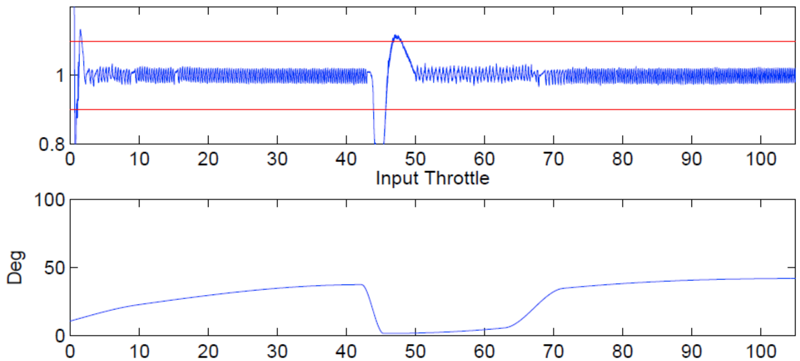



```
e Val_Lim(e1 : real; Min, Max : real)
           returns (s1 : real) ;
r xmin:real, xmax : real;
let
  (xmax , xmin) = if (Max >= Min)
                  then (Max , Min)
                  else (Min , Max) ;
   s1 = if (xmax <= e1)
           then xmax
           else (if (e1 > xmin)
                 then e1
                 else xmin) ;
te .
```

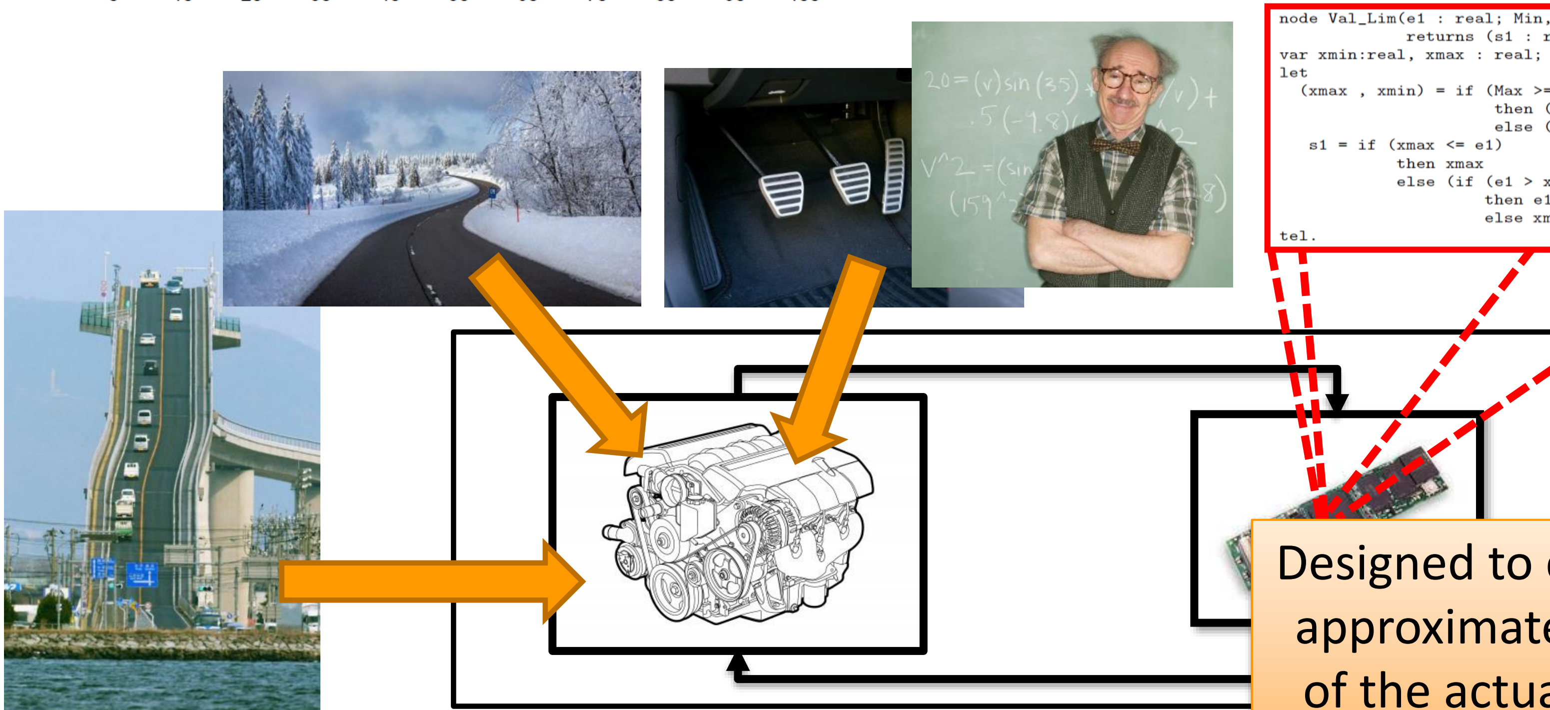
A mix of autocode and manual coding

Real-time execution guarantees

Control design for powertrain



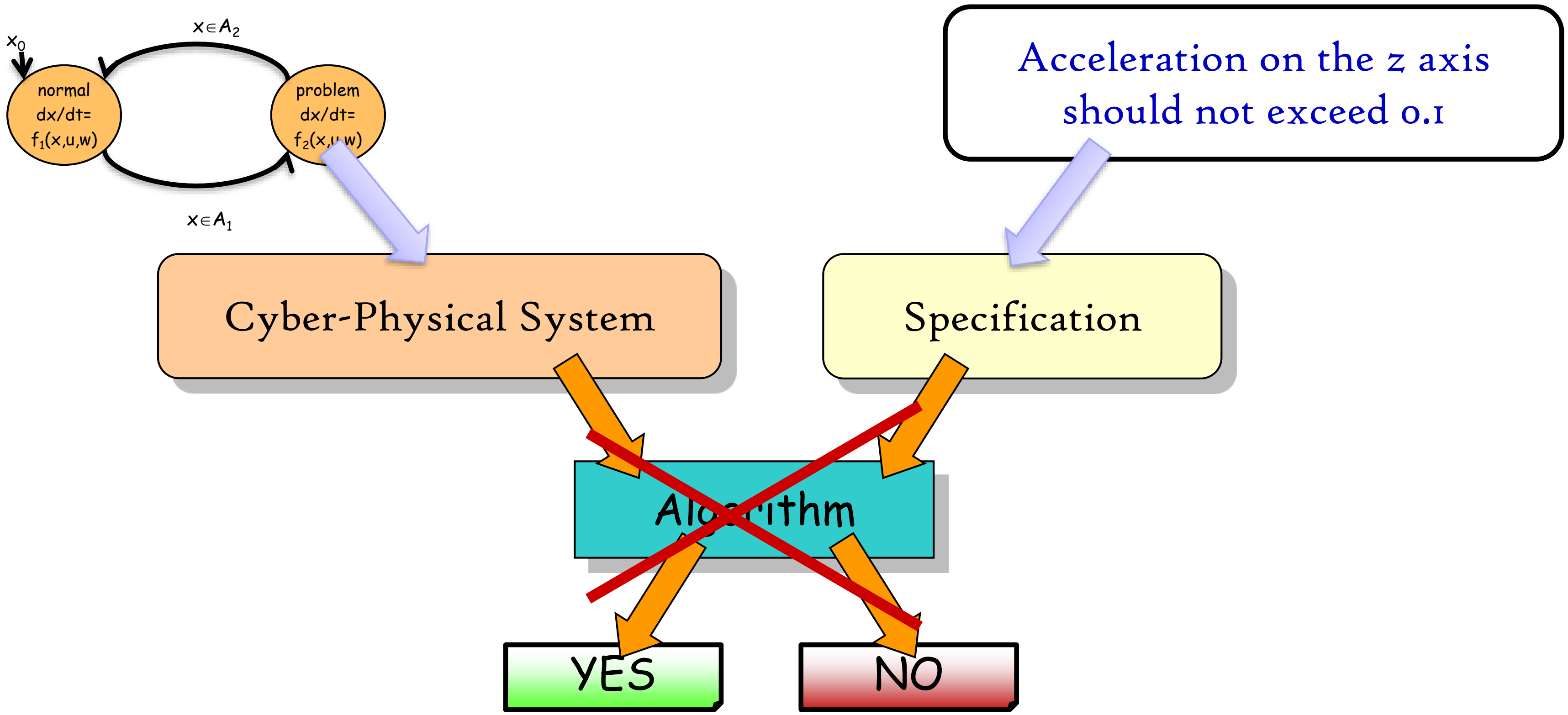
How can we guarantee that the embedded control system will satisfy the design requirements?



```
node Val_Lim(e1 : real; Min, Max : real)
          returns (s1 : real) ;
var xmin:real, xmax : real;
let
  (xmax , xmin) = if (Max >= Min)
                  then (Max , Min)
                  else (Min , Max) ;
   s1 = if (xmax <= e1)
        then xmax
        else (if (e1 > xmin)
              then e1
              else xmin) ;
tel.
```

Designed to control an approximated model of the actual system

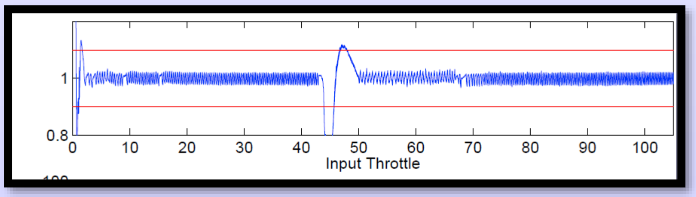
In general, verifying a hybrid system is an undecidable problem!



- 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.

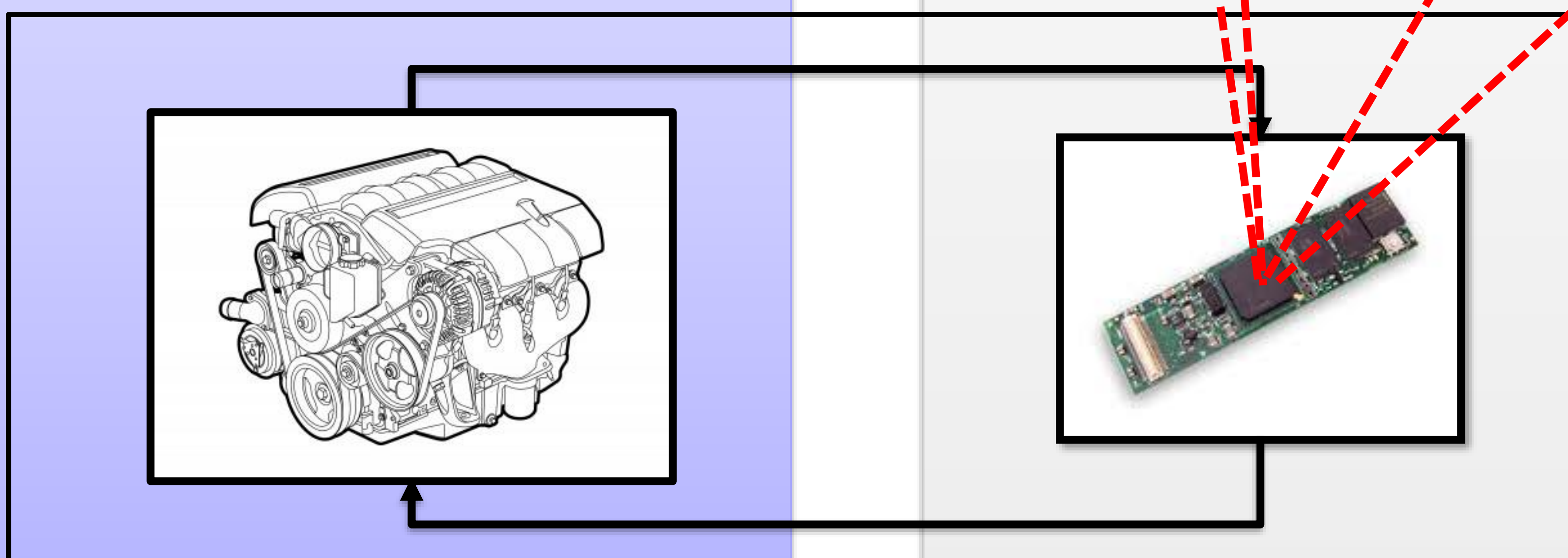
Control design for powertrain

Properties to check are typically on the physical side!



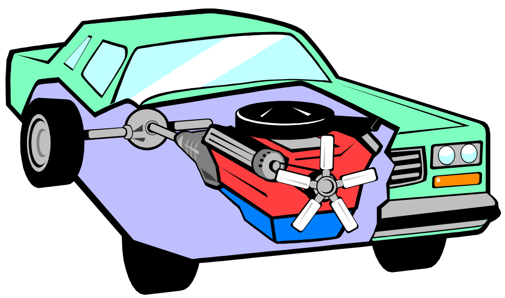
Classical software testing methods apply here!
Still valuable, but …

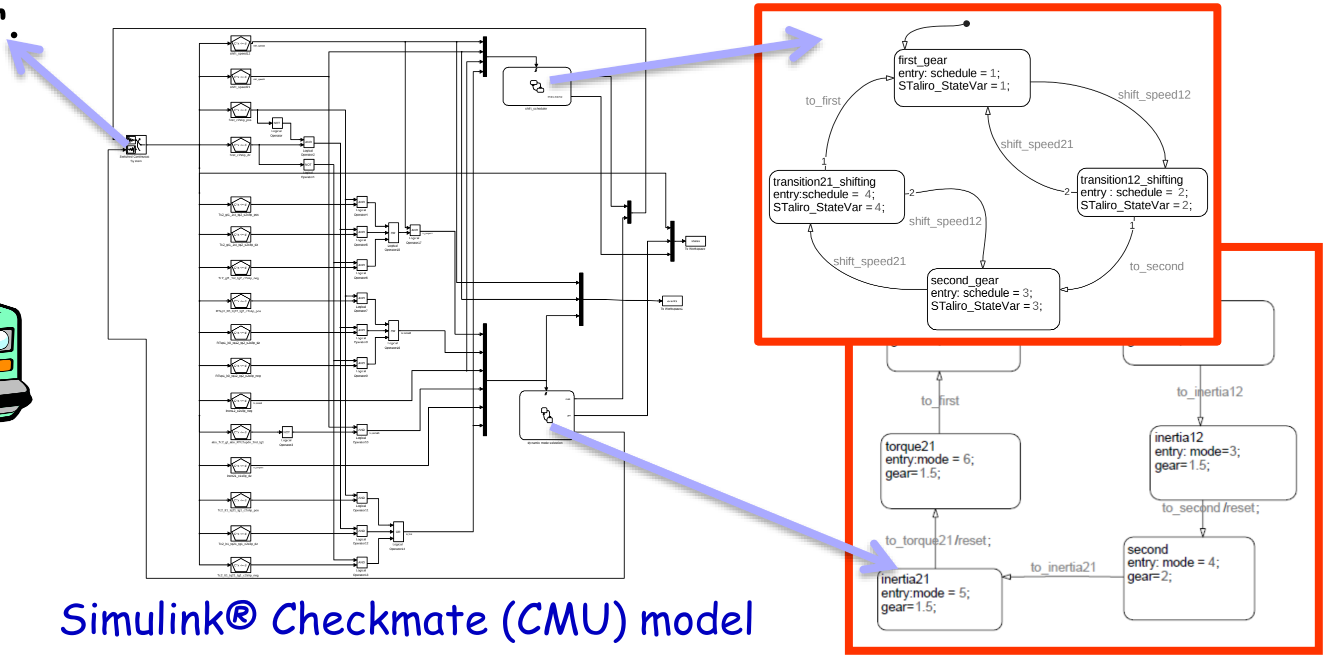
```
node Val_Lim(e1 : real; Min, Max : real)
          returns (s1 : real) ;
var xmin:real, xmax : real;
let
  (xmax , xmin) = if (Max >= Min)
                  then (Max , Min)
                  else (Min , Max) ;
   s1 = if (xmax <= e1)
          then xmax
          else (if (e1 > xmin)
                then e1
                else xmin) ;
tel.
```



Powertrain Challenge Problem*

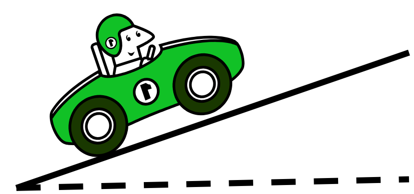
6 state var.





Simulink® Checkmate (CMU) model

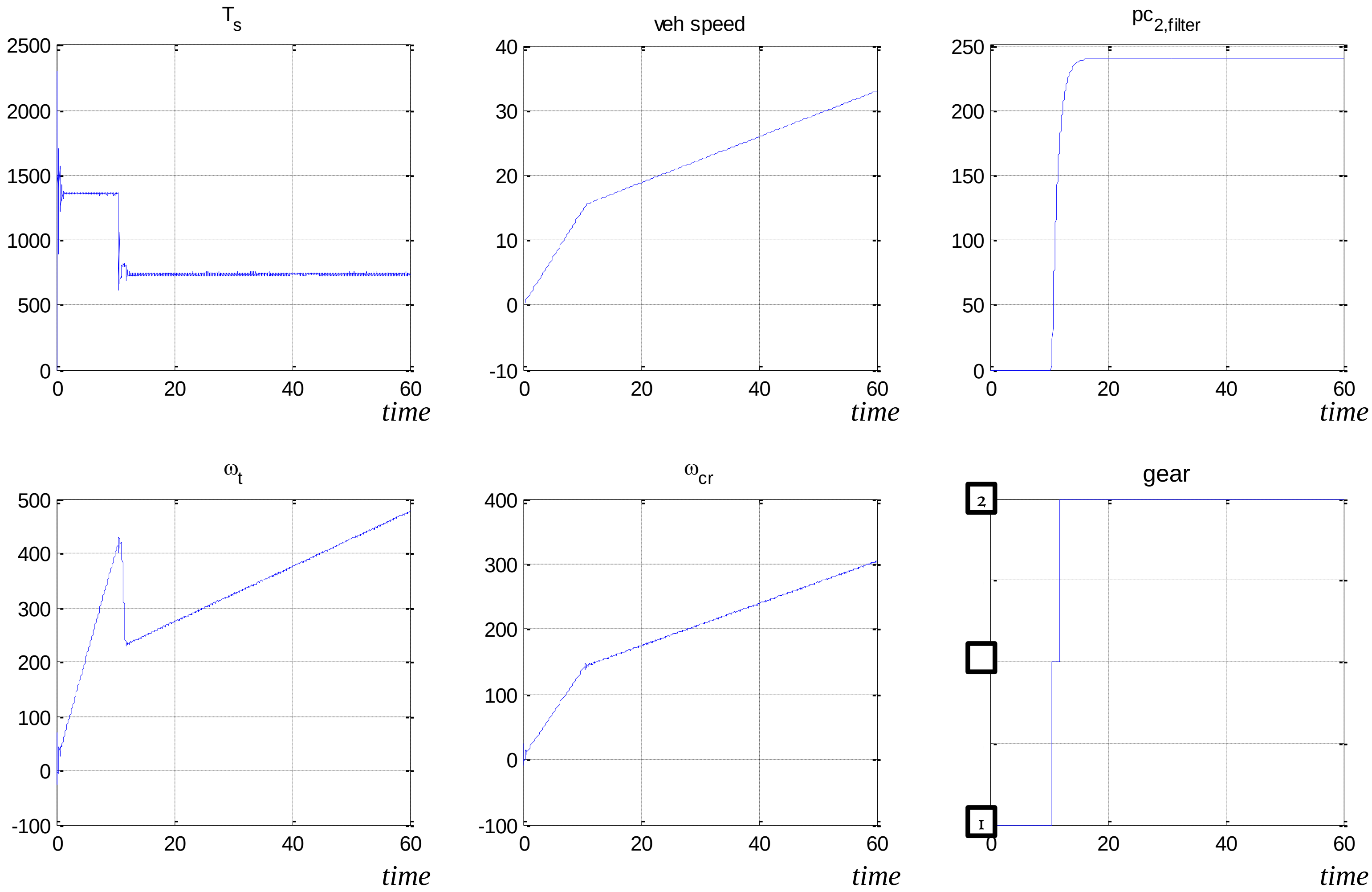
Specification: For constant throttle and road grade the vehicle should not switch from gear 2 to gear 1 to gear 2



* A. Chutinan and K. R. Butts, "Dynamic analysis of hybrid system models for design validation," Ford Motor Company, Tech. Rep., 2002

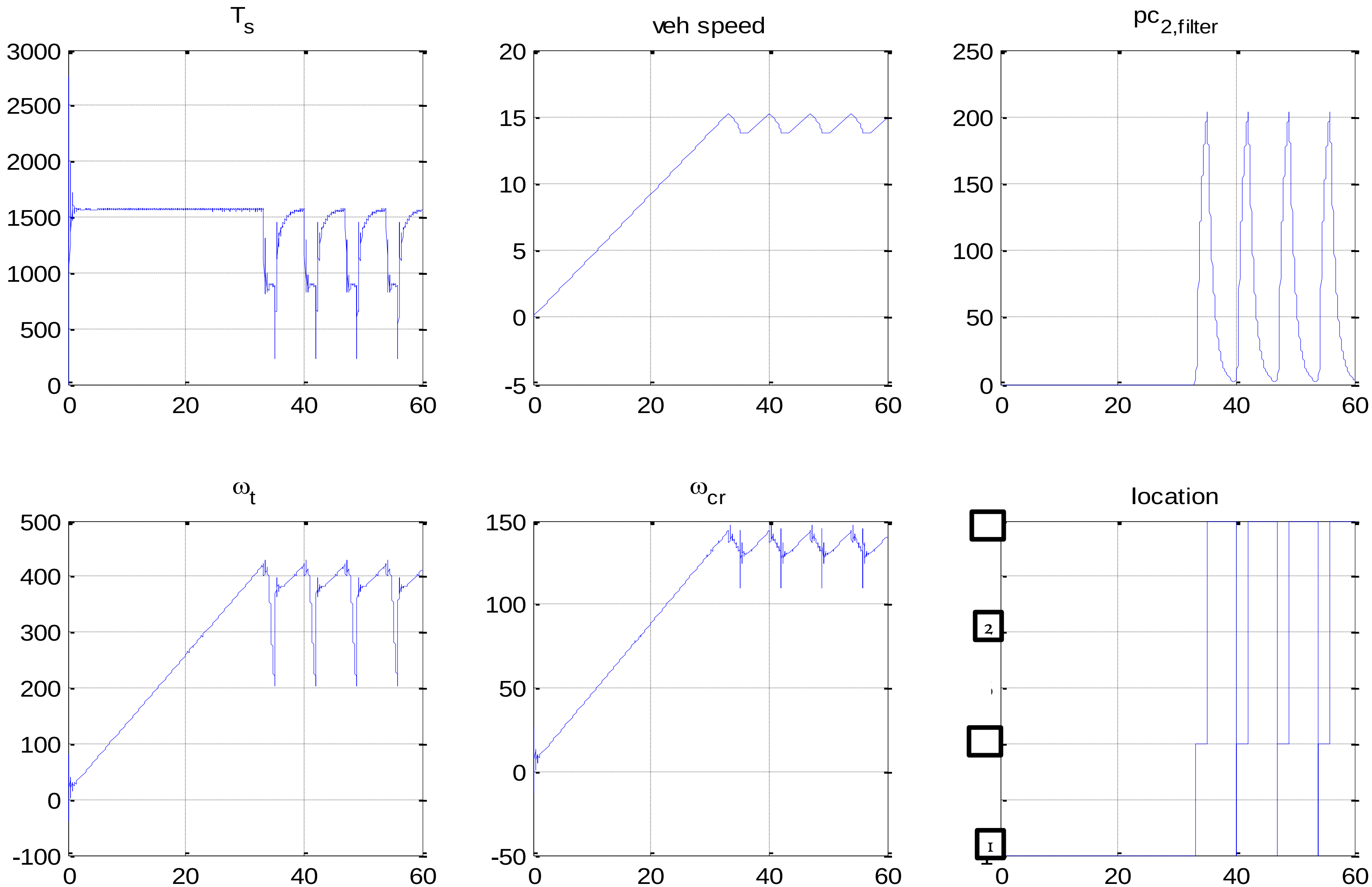
Correct behavior

Throttle = 80, Grade = 0.1



Bad behavior

Throttle ≅ 93.9, Grade ≅ 0.2453



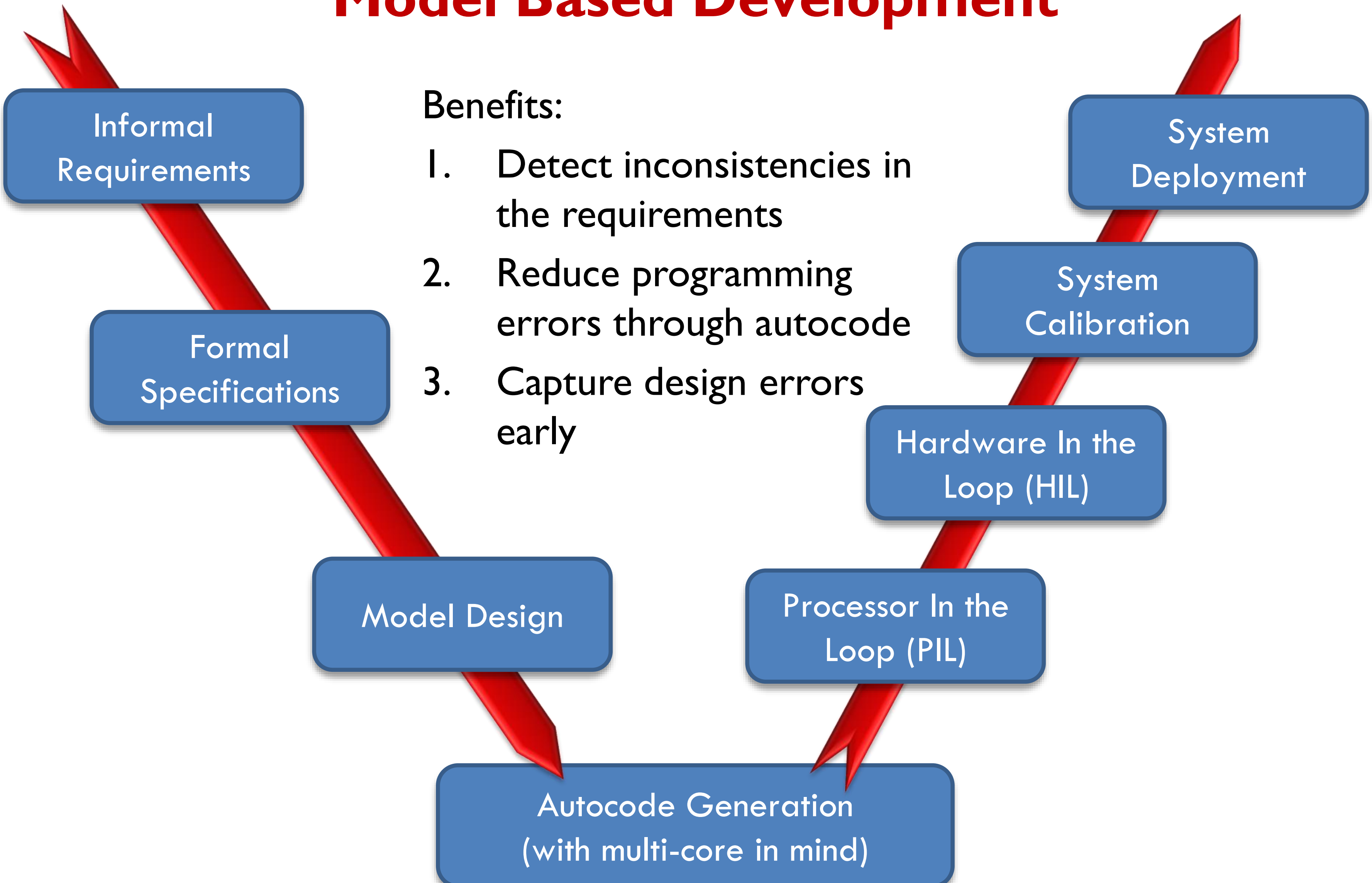
Overview

- Motivation
    - Quick intro to control synthesis challenges
    - Model Based Development
- Formal requirements for CPS
- Requirements driven falsification
- Autonomous vehicle testing
- Parameter mining in requirements
- Conformance testing
- Testing based verification
- Vision, Other topics & Future work

Promising approach to tame complexity: Model Based Development

Benefits:

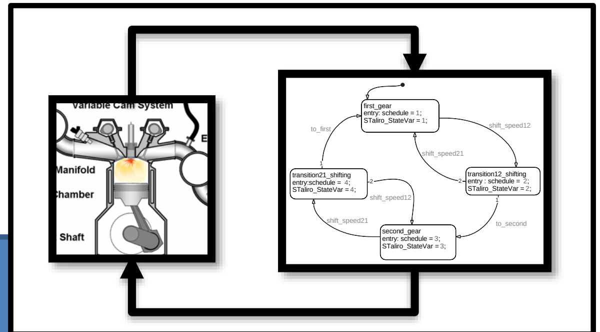
1. Detect inconsistencies in the requirements
2. Reduce programming errors through autocode
3. Capture design errors early



V-Process in Model Based Development

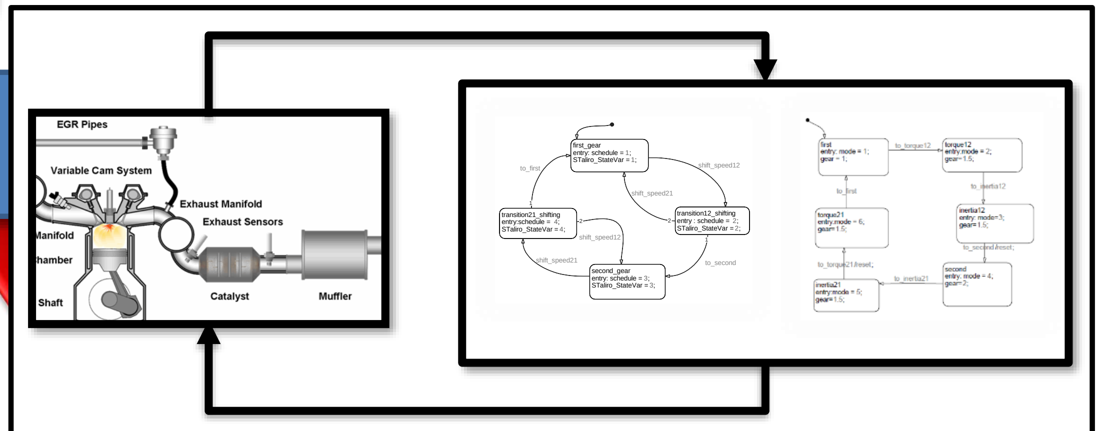
(Formal) Specifications

Model 1



Model updates / modifications until desired level of fidelity is achieved for the targeted physical system.

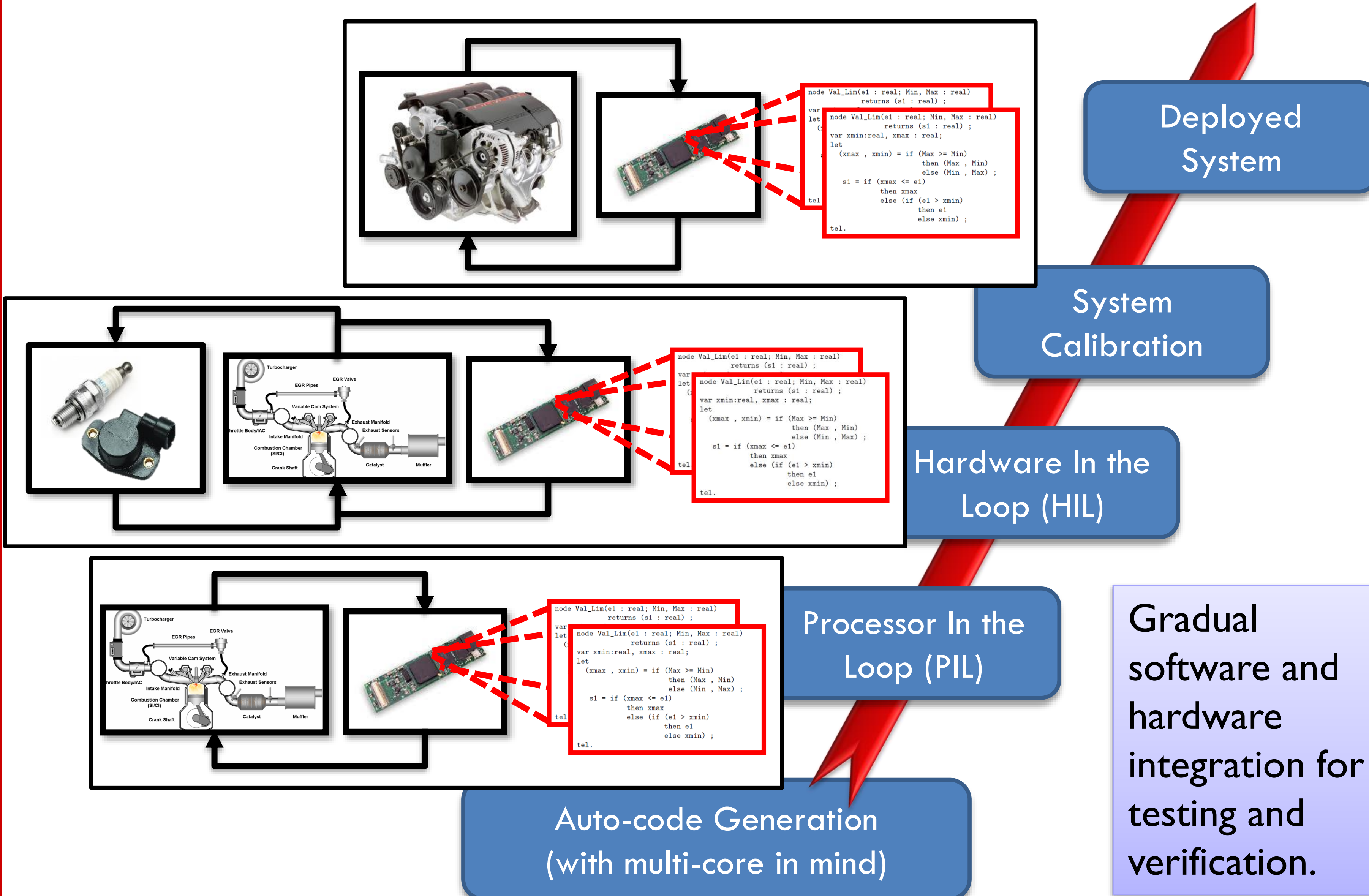
Model 2



...

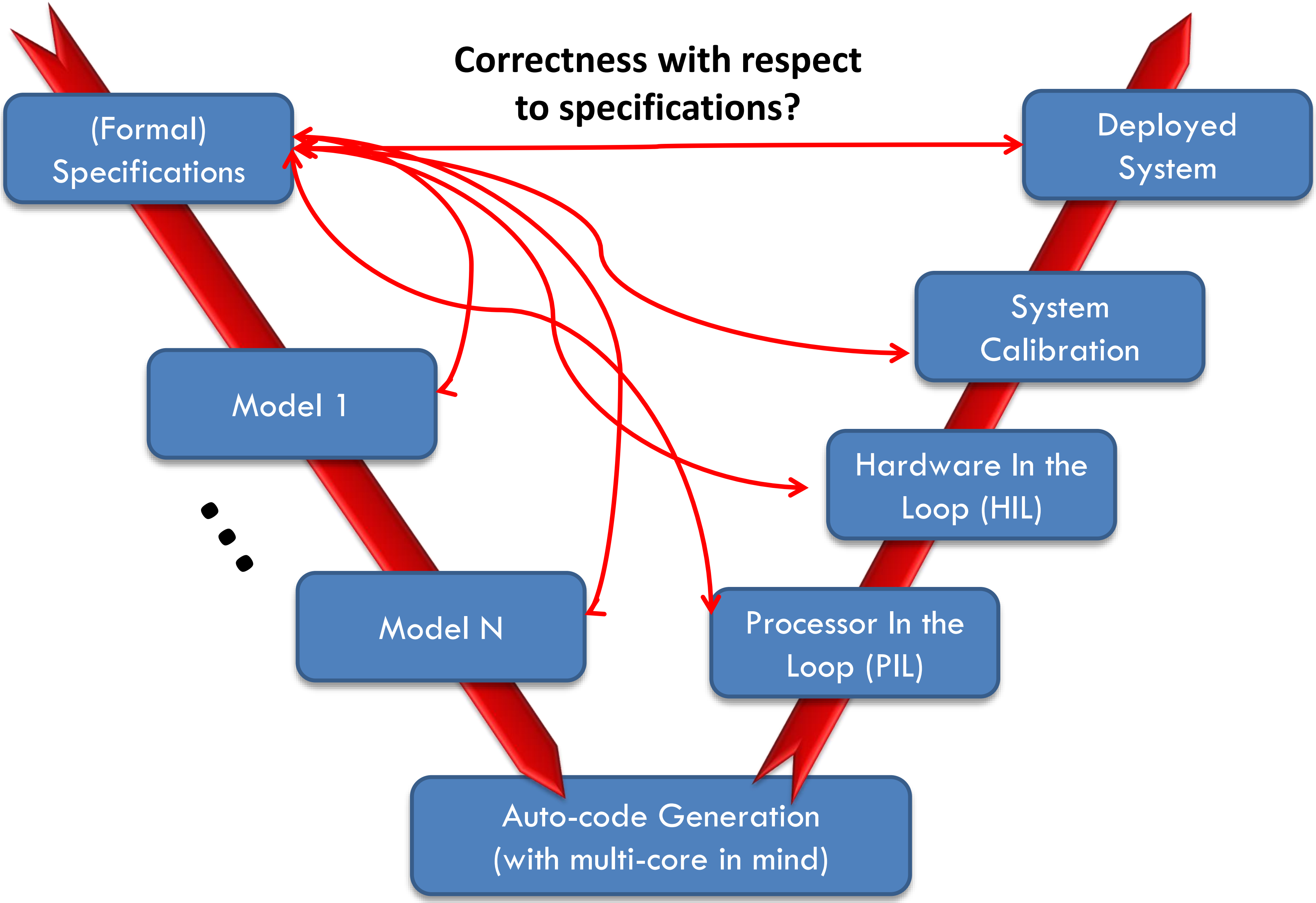
Model N

V-Process in Model Based Development



Gradual software and hardware integration for testing and verification.

V-Process in Model Based Development



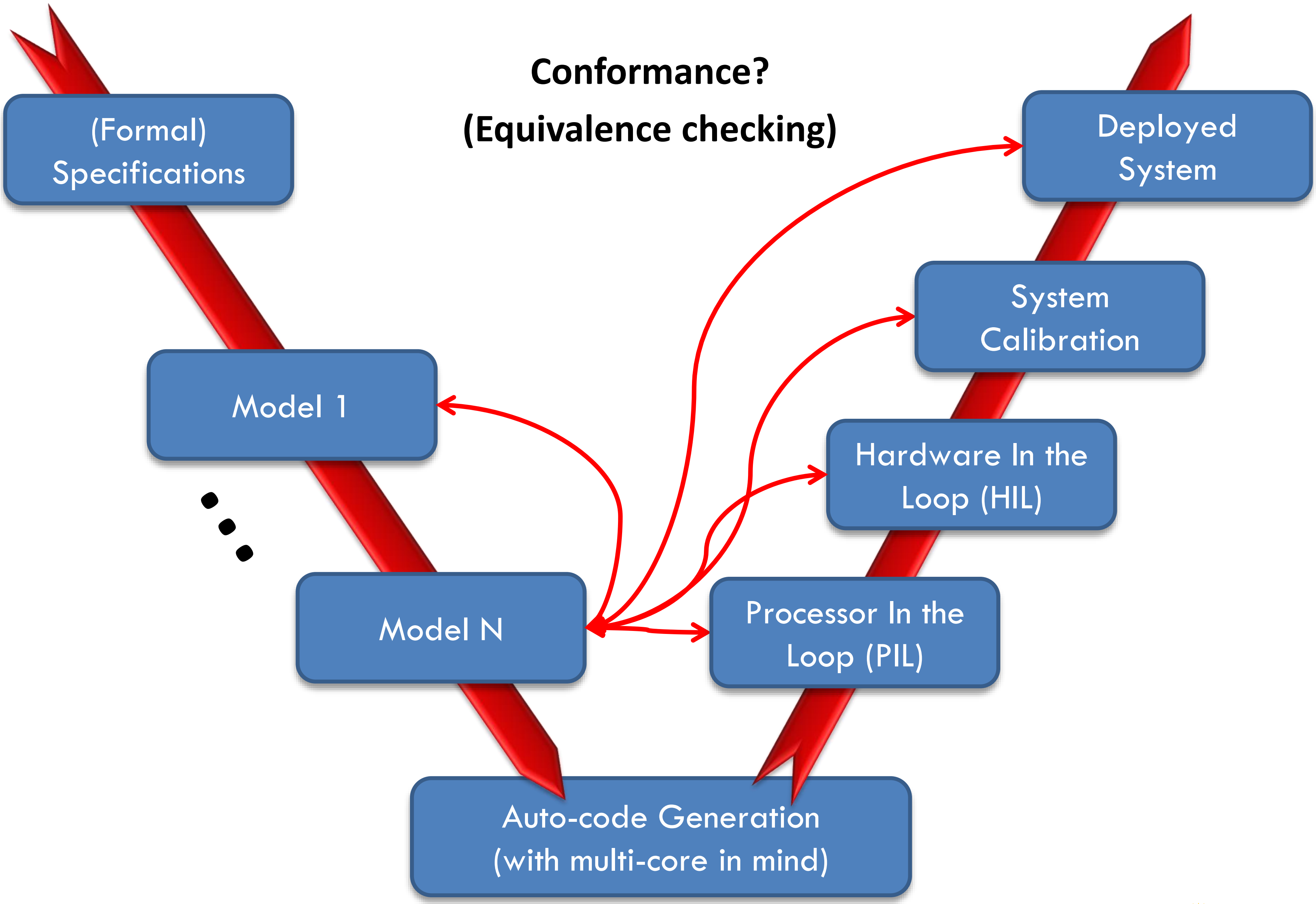
V-Process in Model Based Development

Correctness with respect to specifications?

Challenges in verifying specifications:

- Undecidable problem
  - [as opposed to checking digital circuits]
- Scalability
  - [hundreds of real-valued state variables]
  - [nonlinear dynamics]
  - [physical phenomena not modeled through ODEs, PDEs etc]
  - [time consuming simulations]
- Blackbox components in the model
  - [which may be statefull]
- Hardware in the loop
  - [reproducibility, record & playback, etc]

V-Process in Model Based Development



V-Process in Model Based Development

Conformance?

Challenges in verifying conformance:

- Undecidable problem
  - [as opposed to checking digital circuits – finite state machines]
- Each model version is deterministic (or at most a stochastic) model
  - [Behavior inclusion between models cannot be checked]
- Thus, we need to talk about "distance" between the system behaviors.
  - [What is an appropriate notion of distance?]
- Blackbox components in the model
  - [which may have memory – history matters]
- Hardware in the loop
  - [reproducibility, record & playback, etc]

Overview

Joint work with
George Pappas
University of Pennsylvania

- Motivation
  - Quick intro to control synthesis challenges
  - Model Based Development
- Formal requirements for CPS
- Requirements driven falsification
- Autonomous vehicle testing
- Parameter mining in requirements
- Conformance testing
- Testing based verification
- Vision, Other topics & Future work

Trust? : Sampling of automotive recalls (~2011-12) due to software errors …

- "A software error may prevent the transmission from downshifting, such as shifting from 5th … y of the problem. … e stall, increasing
- … the software that "allows the ECU to establish a 'handshake' with the engine is in error. The E… ngine is found to be out of tol… ppens, the ECU triggers a fau… lition outside its prescribed tolerances, a rough idle or stalling situation ensues."
- … to u… tain circums… otor to rotate …
- If the fault … gnition while driving - w… disables power ste…
- …

When in 5th gear and RPM drops below x, then the system should always switch from 5th to 4th gear.

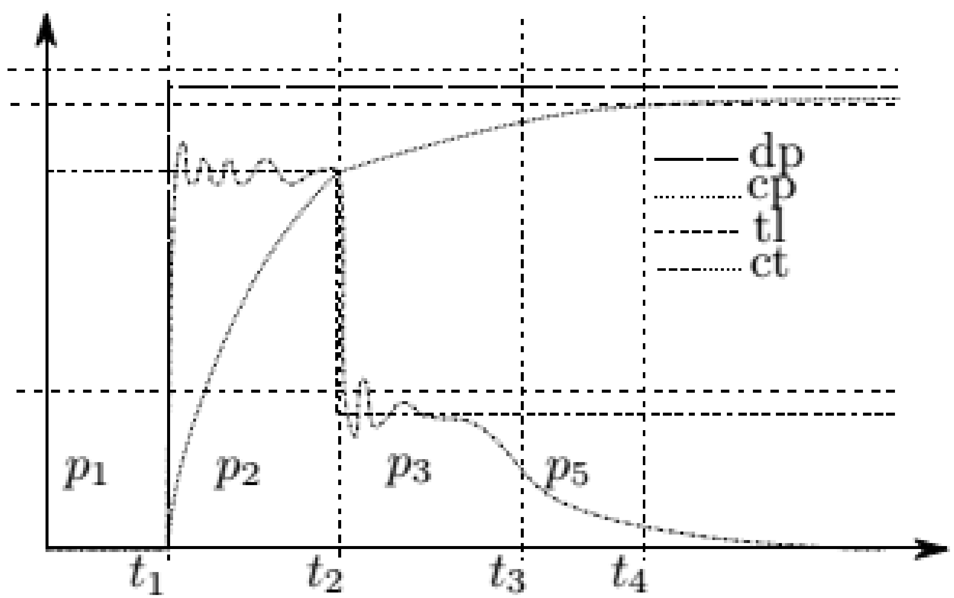
The engine should never stall while idle.

The electric motor should always rotate in the direction selected by the transmission.

The cruise control should always disengage when the "turn off" button is pressed.

How complex can specifications be*?

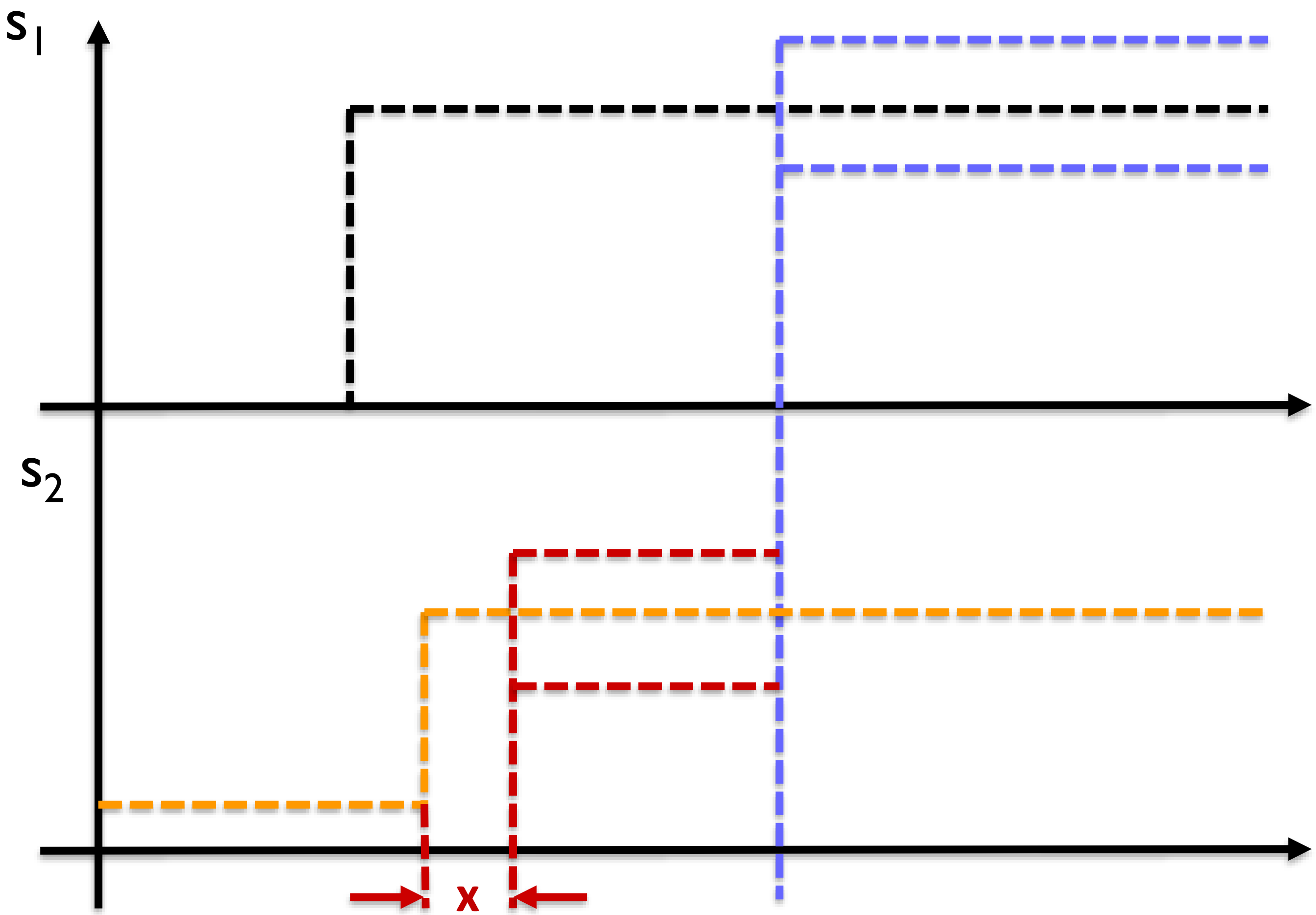
NL: During the position (cp) regulation after a step input on demand (dp), when the absolute value of the maximum torque limit (tl) decreases with a step (precondition), the absolute value of the actuator response in torques (ct) must be less than the torque limit plus 10% in less than 10 ms (postcondition)



* *H. Roehm, R. Gmehlich, T. Heinz, J. Oehlerking and M. Woehrle: Industrial Examples of Formal Specifications for Test Case Generation, ARCH 2015*

Specification: When ORANGE event happens after the BLACK EVENT, signal s_2 should stabilize in the RED region within x time units. Signal s_2 should only stay in the RED region only until signal s1 has stabilized in the BLUE region.

How do we mathematically capture such requirements so that we can automatically verify/test a system?



Example adapted from Bosch requirements

Metric Interval Temporal Logic: Semantic Intuition

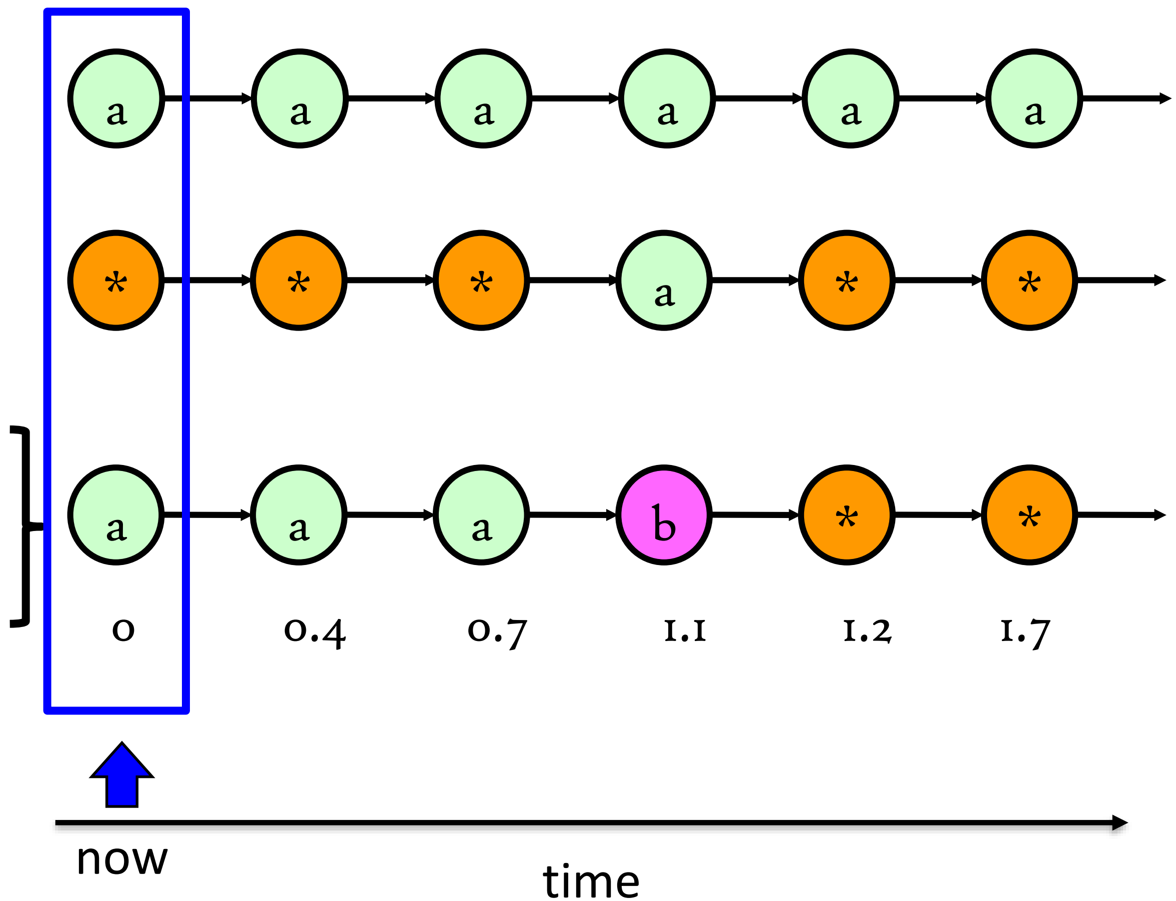
$$\phi \ ::= \top \mid p \mid \neg\phi \mid \phi_1 \vee \phi_2 \mid G_I\phi \mid F_I\phi \mid \phi_1 U_I \phi_2$$

Ga - always a

$F_{[1,3]}a$ - eventually a

$a\ U\ b$ - a until b

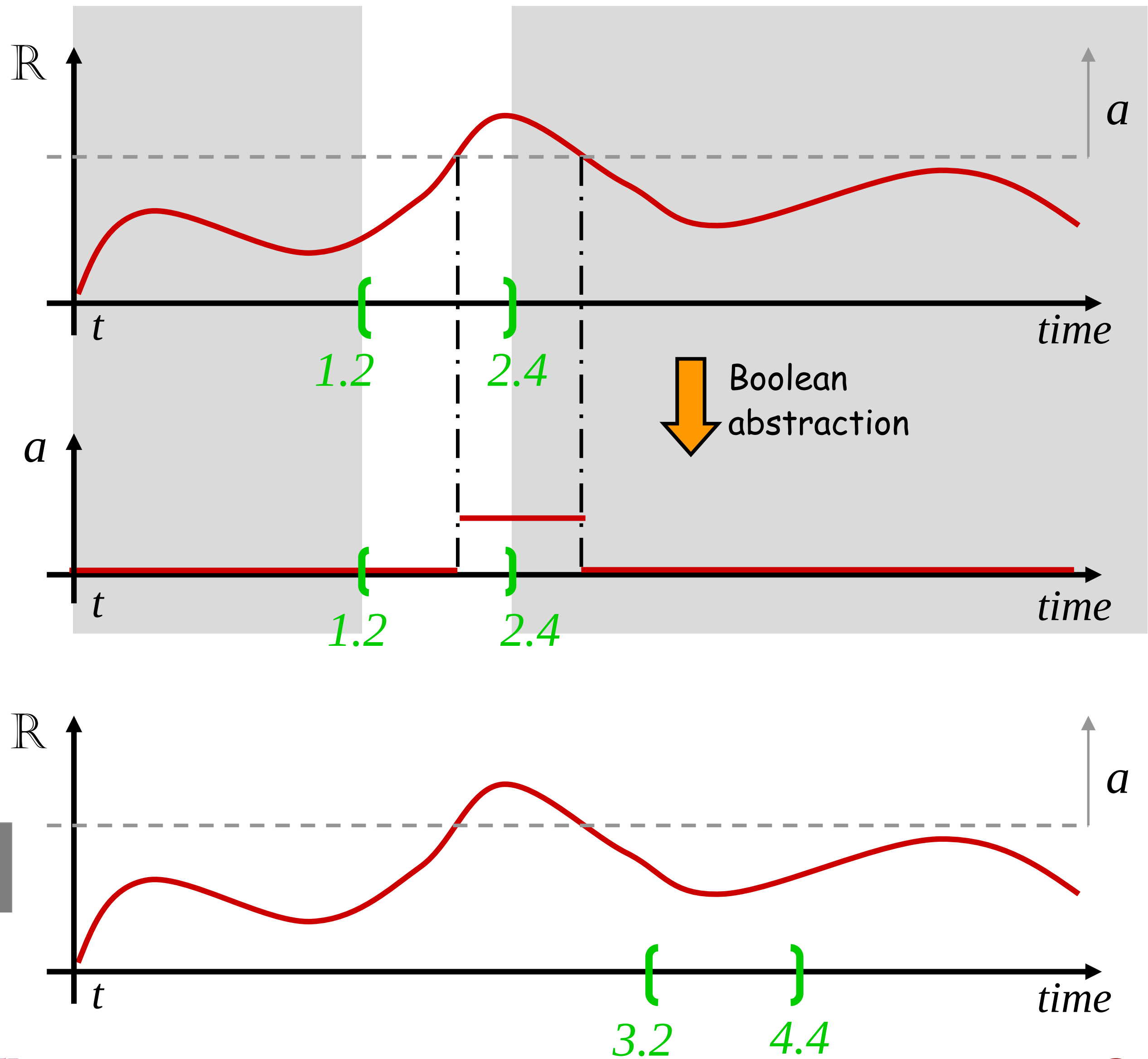
$a\ U_{[1,1.5]}\ b$ - a until b



MTL : An example for signals

$F_{[1.2,\ 2.4]}\ a$

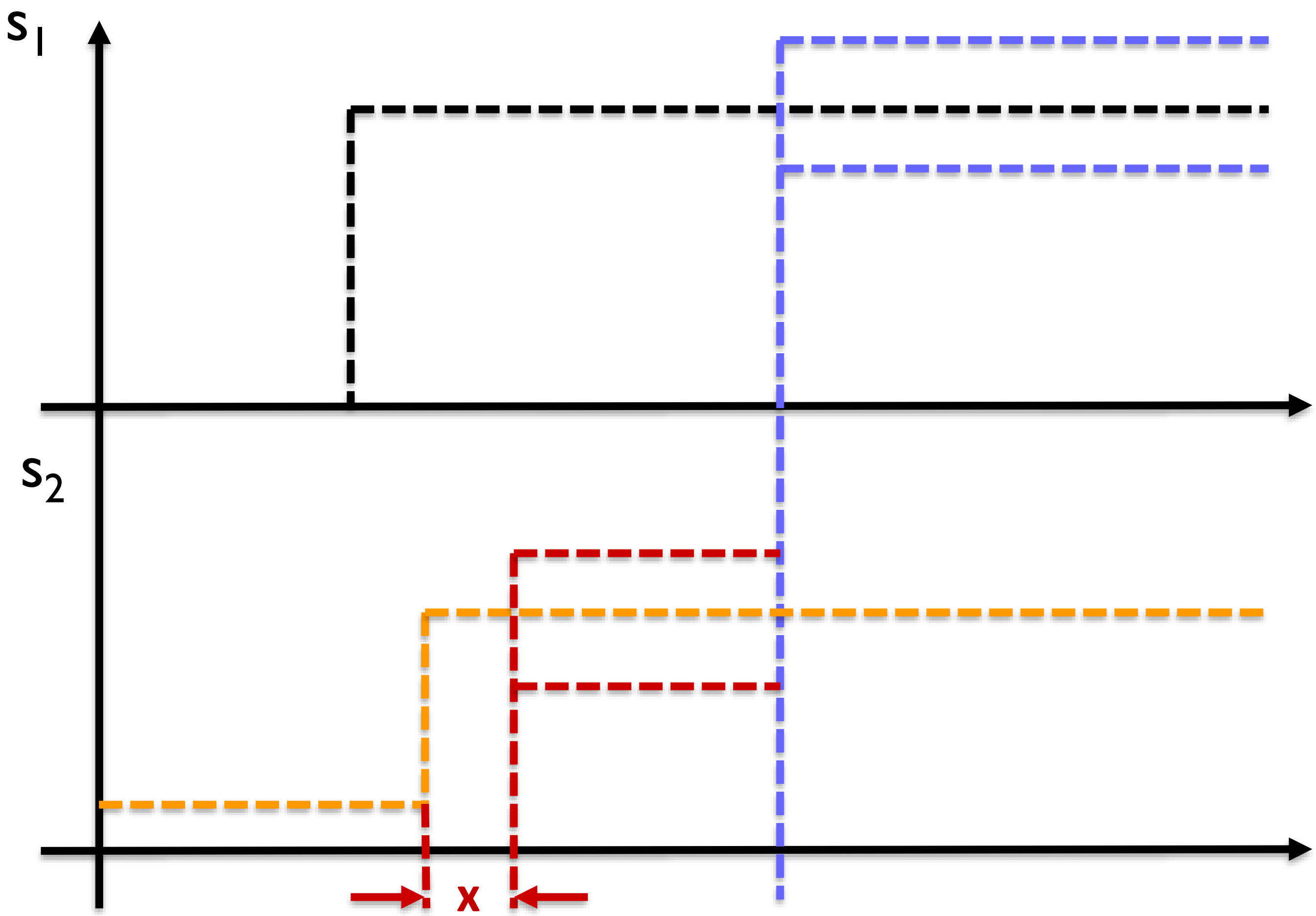
$G_{[3.2,\ 4.4]}\ \neg a$



Possible formalizations?

$$G(\ (\text{Orange} \wedge P_{[0,y]}\ \text{Black}) \rightarrow F_{[0,x]}(\ (\text{s2 in red})\ U\ G\ (\text{s1 in blue})\ ))$$

$$G(\ (\text{Orange} \wedge P_{[0,y]}\ \text{Black}) \rightarrow G_{[x,\infty)}(\ (\text{s2 in red}) \vee G\ (\text{s1 in blue})\ ))$$



Example adapted from Bosch requirements

Formalizing Complex Specifications

1. Find values for the initial parameters such that starting from 0 speed, the gear transitions from second to first to second.

$$\varphi_1 = \neg F(gear_2 \wedge F(gear_1 \wedge F gear_2))$$

2. A more "useful" property is to find constrain the gear change from second to first to second not happen within 2.5 sec.

$$\varphi_2 = G((\neg gear_1 \wedge X\ gear_1) \rightarrow G_{[0,2.5]} \neg gear_2)$$

3. Verify that the jitter is within acceptable limits

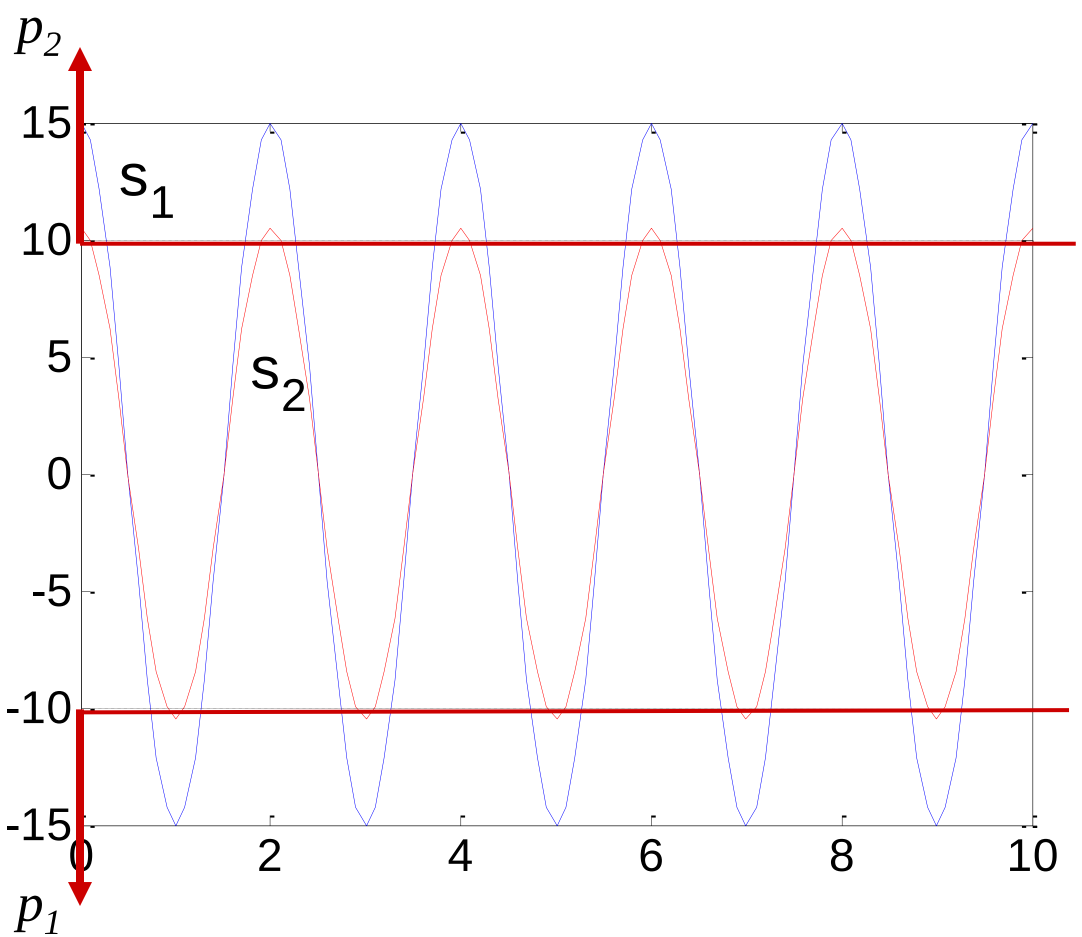
$$\varphi_3 = G(gear_{21} \rightarrow |dTs/dt| < 450)$$

Boolean semantics are problematic for CPS:

Two different signals can satisfy the same spec, but …

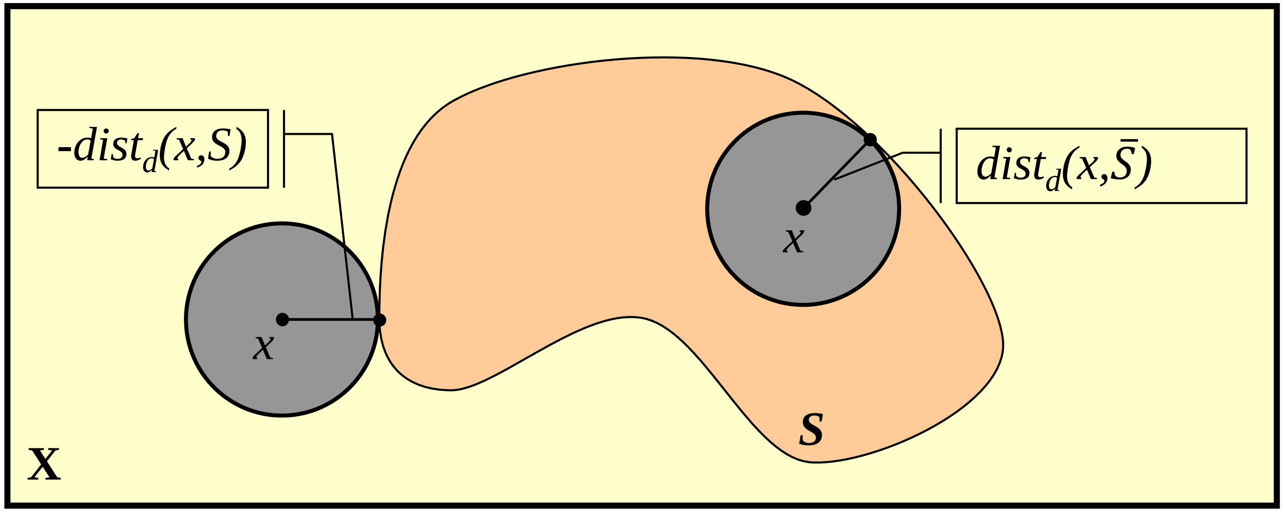
MTL Spec:

$G(p_1 \rightarrow F_{\leq 2}\, p_2)$



Robust Semantics for MTL

$$[\![\underbrace{x \in S}_{p}]\!](x,t) = Dist(x(t), S)$$



$$[\![\neg\varphi_1]\!](x,t) = \sim[\![\varphi_1]\!](x,t)$$

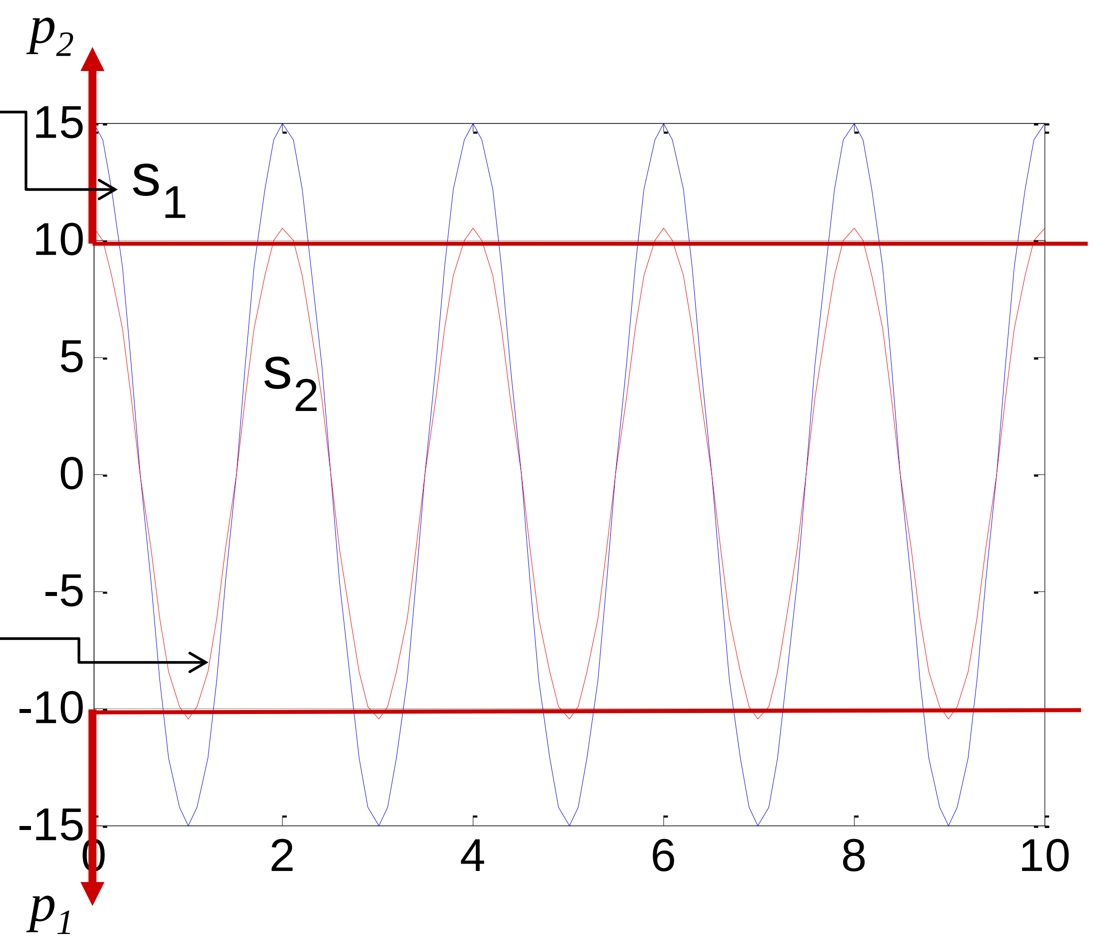
$$[\![\varphi_1 \vee \varphi_2]\!](x,t) = \max([\![\varphi_1]\!](x,t), [\![\varphi_2]\!](x,t))$$

$$[\![\varphi_1 U_I \varphi_2]\!](x,t) = \sup_{t' \in t \oplus I} \max([\![\varphi_2]\!](x,t'), \inf_{t'' \in [t,t')} [\![\varphi_2]\!](x,t''))$$

Now satisfaction can be quantified …

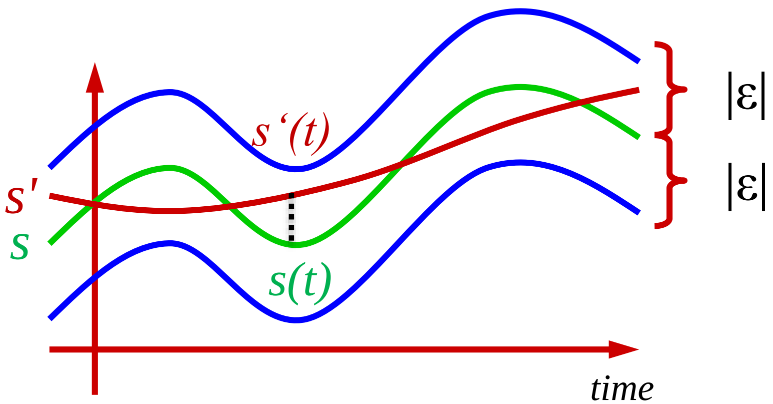
$[\![G(p_1 \rightarrow F_{\leq 2}\, p_2)]\!](s_1) = 5$

$[\![G(p_1 \rightarrow F_{\leq 2}\, p_2)]\!](s_2) = 0.5$

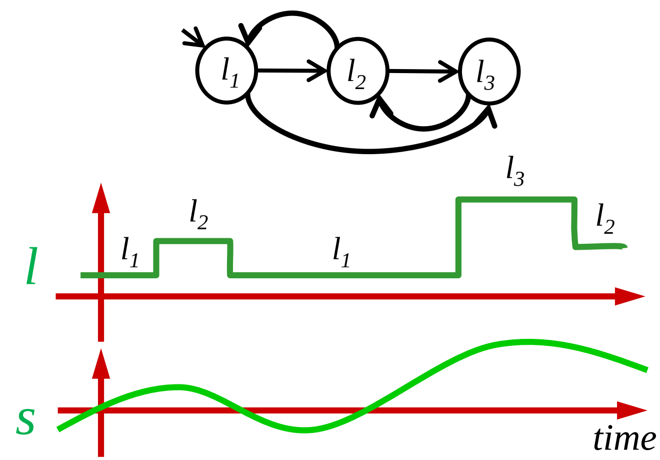


Theoretical Guarantees

Theorem: Let ϕ be an MTL formula, s be a (continuous or discrete time) signal and $|\varepsilon|>0$ be the *robustness parameter* of ϕ with respect to s, then for all s' in $B_\rho(s,\varepsilon)$ we have that $s \models \phi$ *iff* $s' \models \phi$



$\rho(s,s') = \sup_t d(s(t),s'(t))$
where d is a metric



$\rho(s,s') = \sup_t \mathbf{d}((s,l)(t),(s',l')(t))$
where $\mathbf{d}$ is a generalized quasi metric

Abbas et al, *Probabilistic Temporal Logic Falsification of Cyber-Physical Systems*, ACM TECS 2013
Fainekos and Pappas, *Robustness of temporal logic specifications for continuous-time signals*, TCS 2009

Robust Semantics for MTL

$$[\![x \in S]\!](x,t) = Dist(x(t), S)$$
$$[\![\neg\varphi_1]\!](x,t) = \sim[\![\varphi_1]\!](x,t)$$
$$[\![\varphi_1 \vee \varphi_2]\!](x,t) = \max([\![\varphi_1]\!](x,t), [\![\varphi_2]\!](x,t))$$
$$[\![\varphi_1 U_I \varphi_2]\!](x,t) = \sup_{t' \in t \oplus I} \max([\![\varphi_2]\!](x,t'), \inf_{t'' \in [t,t')} [\![\varphi_2]\!](x,t''))$$

Algorithm I

- Based on formula re-writing
- Suitable for runtime monitoring algorithms
- *Details Fainekos & Pappas, RV 2006*

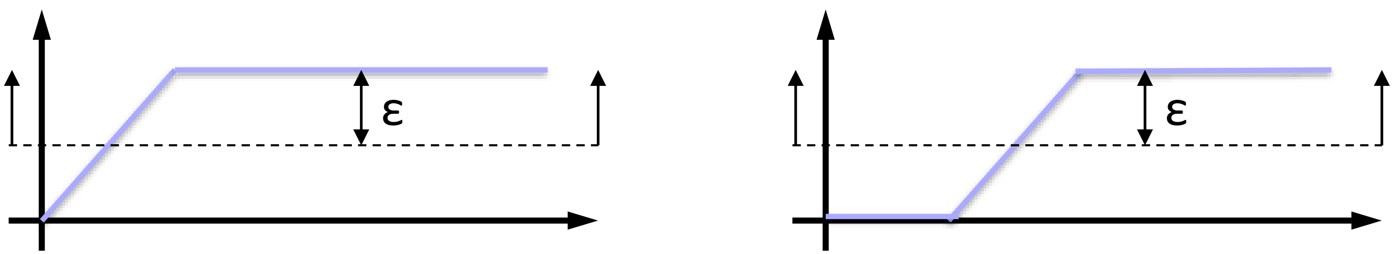
Algorithm II

- Based on dynamic programming
- Suitable for offline testing
- **MTL formulas: O(|ϕ| |τ| c),** where c = max $_{0\leq j\leq|\tau|,\ I\in T(\phi)}$ |[j, max J(j, I)]|
- *Details Fainekos et al ACC 2012*

Algorithms I & II adapted from prior results on Boolean semantics by Thati, Rosu and Havelund

Keep in mind …

- State robustness does not capture robustness with respect to time:
  - These signals have the same robustness value with respect to the specification "eventually go above the threshold"



  - For such cases time robustness or integration of state robustness must be utilized.

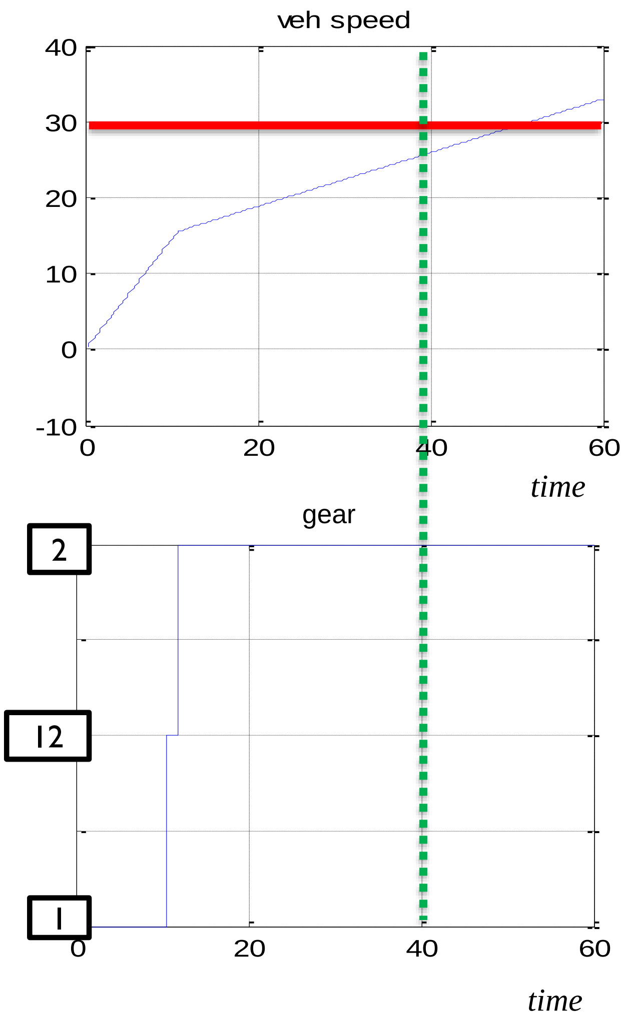
See discussion and extensions in :
- Donze & Maler, Robust satisfaction of Temporal Logic over Real-valued signals, FORMATS, 2010
- Akazaki & Hasuo, Time Robustness in MTL and Expressivity in Hybrid System Falsification, CAV, 2015
- Many other follow up papers …

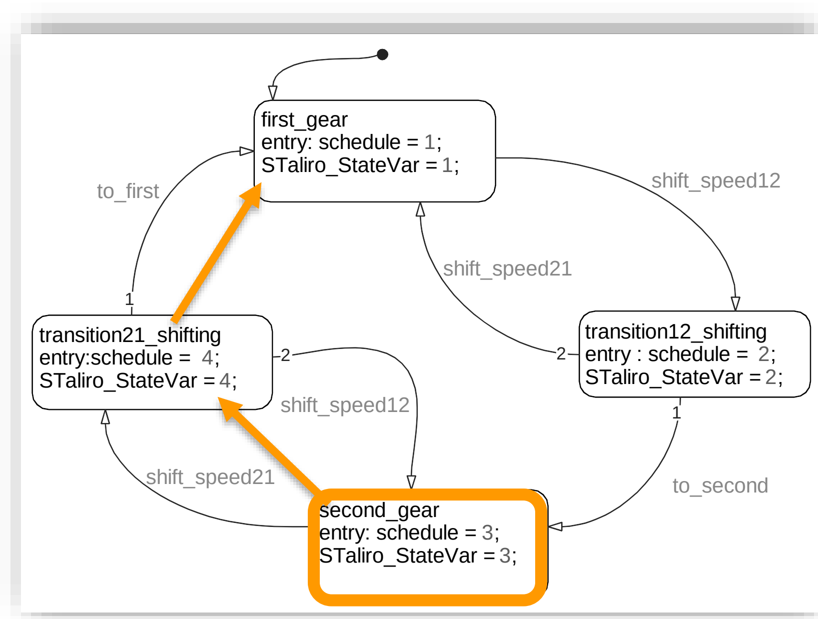
Example 1: Hybrid trajectory robustness

Specification: "*Within the time interval [40,60] do not get into gear 1 with speed greater than 30*"

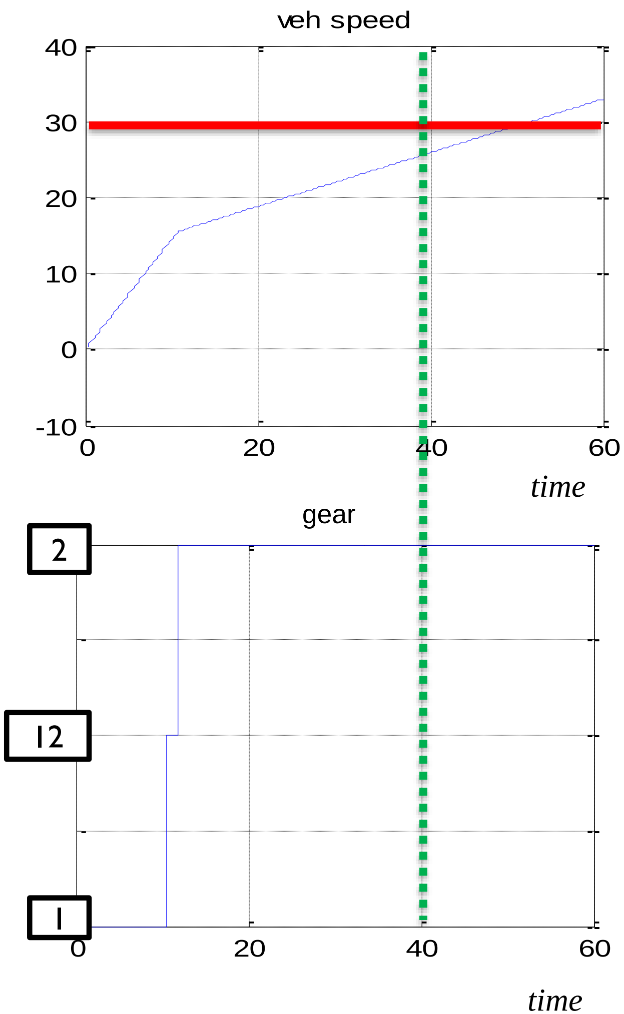
$$\psi_1 = G_{[40,60]} \neg(gear=1 \wedge v \geq 30)$$
$$= G_{[40,60]} (gear \neq 1 \vee v < 30)$$

Robustness: ε = <2, 21.9736>





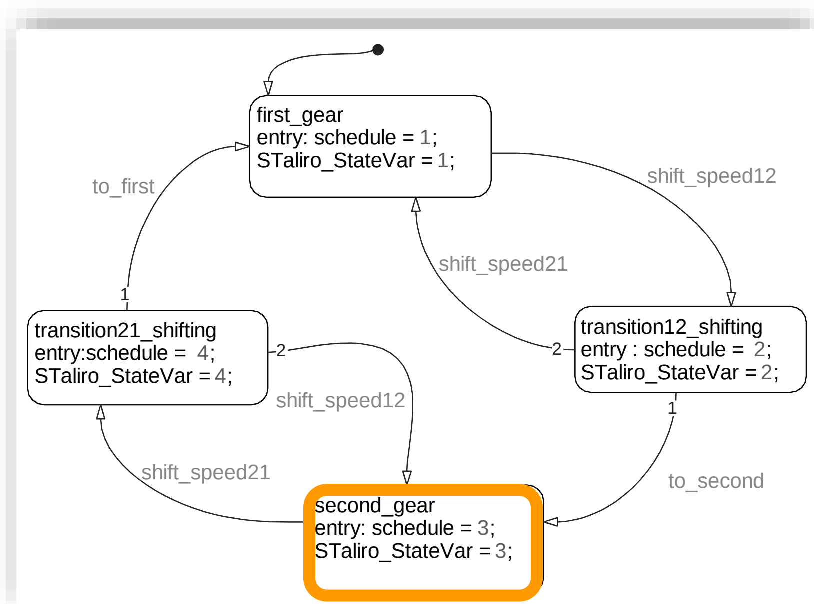
Example 2: Hybrid trajectory robustness



Specification: "*Within the time interval [40,60] do not get into gear 2 with speed greater than 30*"

$$\psi_1 = G_{[40,60]} \neg(\text{gear}=2 \wedge v \geq 30)$$
$$= G_{[40,60]} (\text{gear} \neq 2 \vee v < 30)$$

Robustness: ε = <0, -2.9334>

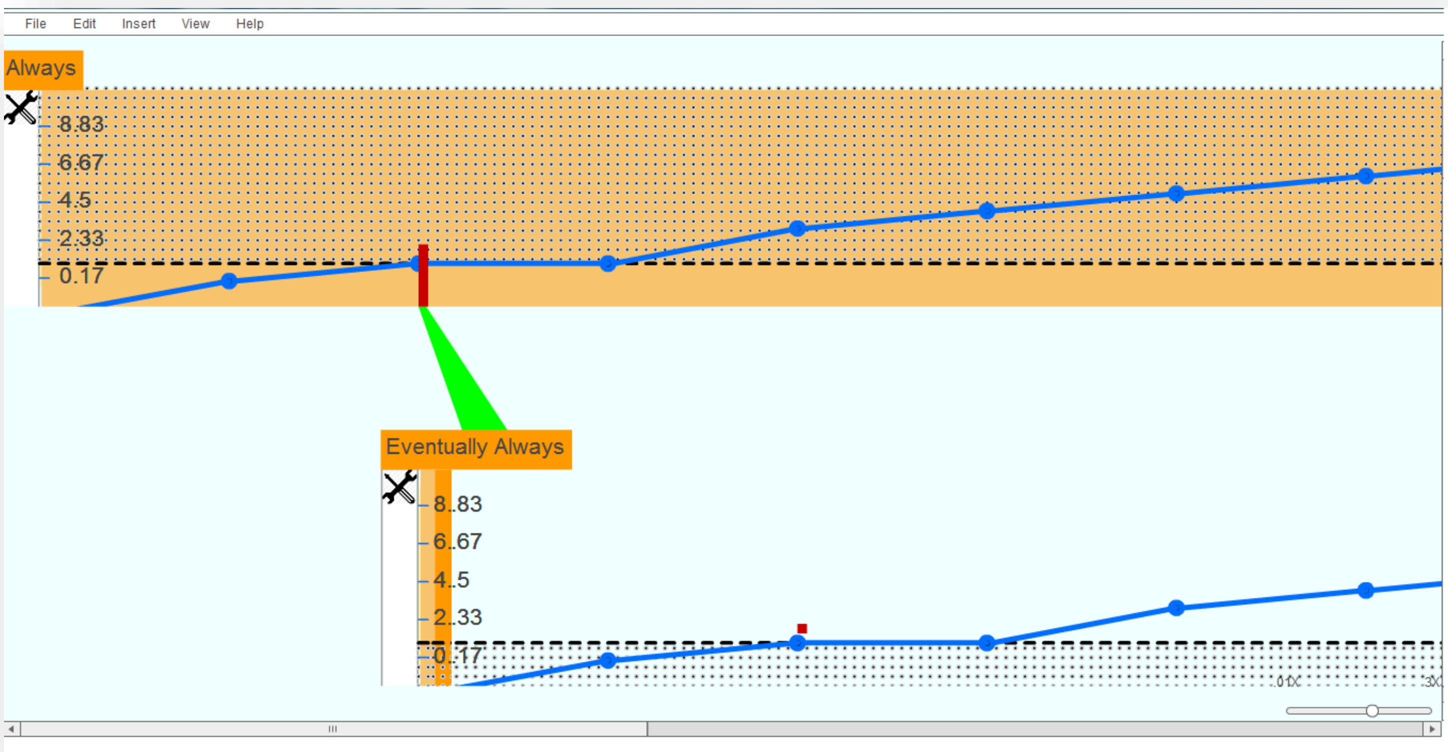


Specification Visualization

Formal Specifications

We have developed a graphical formalism for MTL specification elicitation. Example:

$$\phi_5 = G((\lambda_{diff} > 0.1) \rightarrow F_{[0,1]}G_{[0,1]}(\lambda_{diff} < 0.1))$$



[Hoxha, Bach, Abbas, Dokhanchi, Kobayashi and Fainekos, DIFTS 14]

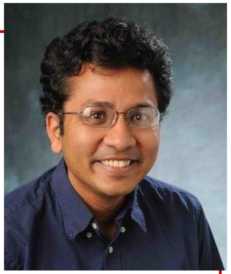
TOYOTA

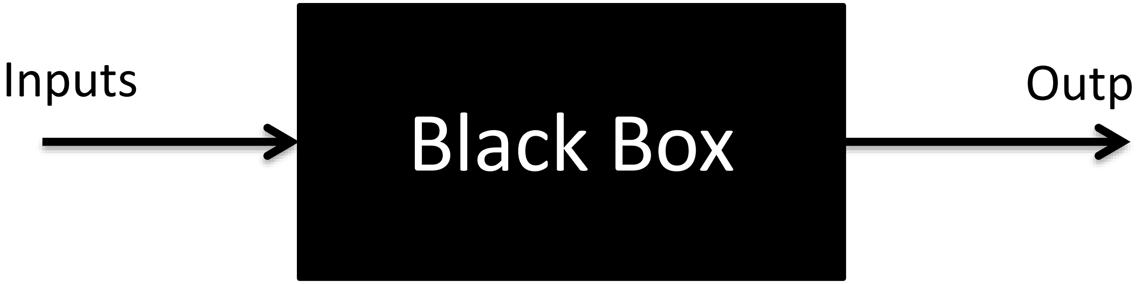
CENTER FOR EMBEDDED SYSTEMS
an NSF Industry/University Cooperative Research Center

Overview

- Motivation
  - Quick intro to control synthesis challenges
  - Model Based Development
- Formal requirements for CPS
- **Requirements driven falsification**
- Autonomous vehicle testi
- Parameter mining in re
- Conformance testing
- Testing based verification
- Vision, Other topics & Futur

Joint work with
S. Sankaranarayanan
CU, Boulder





BLACK BOX TESTING

Temporal Logic falsification as robustness minimization: Example

System:

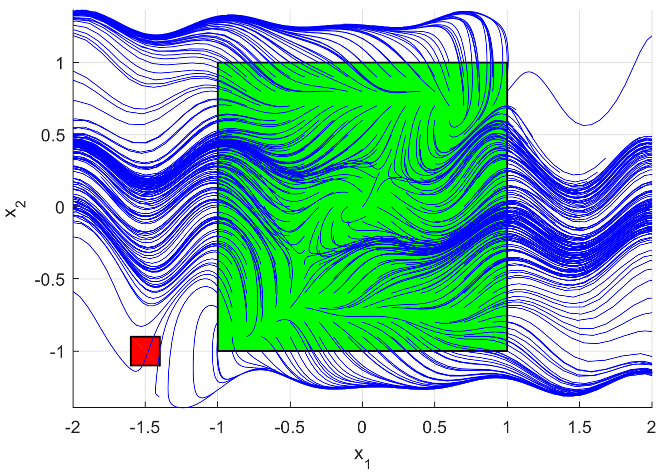
dx/dt = x-y+0.1t

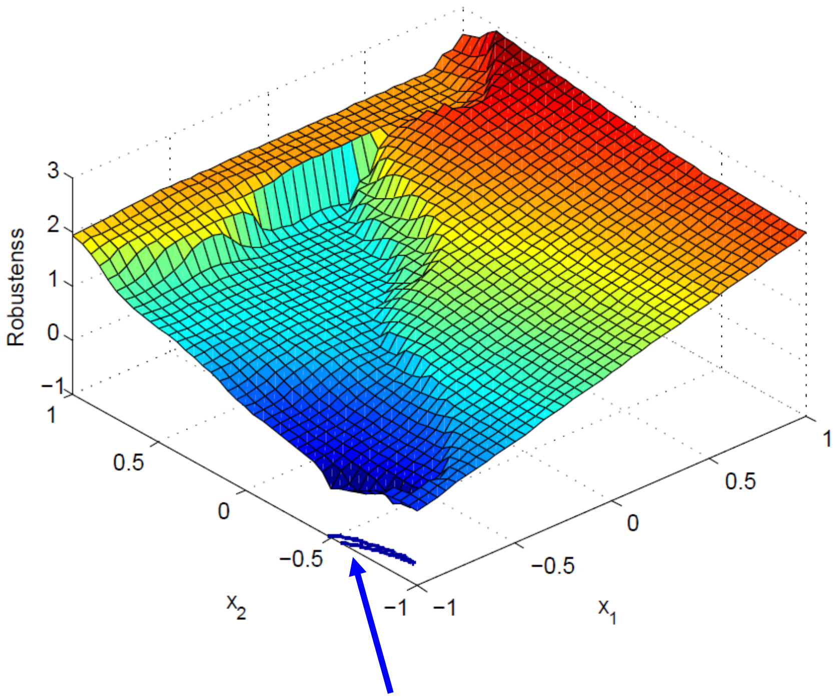
dy/dt = ycos(2πy)-xsin(2πx)+0.1t

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

Specification: $G_{[0,2]}\, \neg a$

where O(a) = [-1.6,-1.4]x[-1.1,-.9]





Zero robustness level set:

Any initial condition within this set will produce a falsifying trajectory.

Temporal Logic falsification as robustness minimization: Example

System:

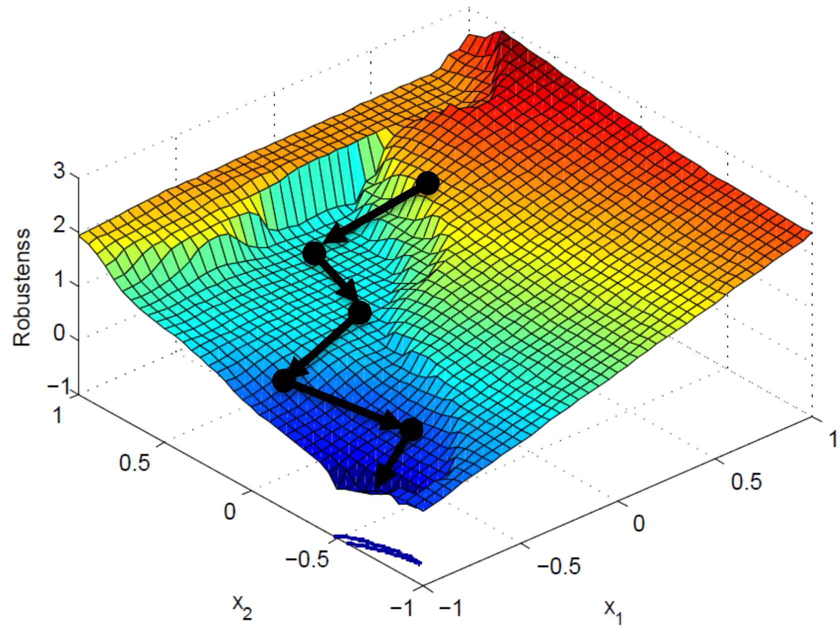
$$dx/dt = x - y + 0.1t$$

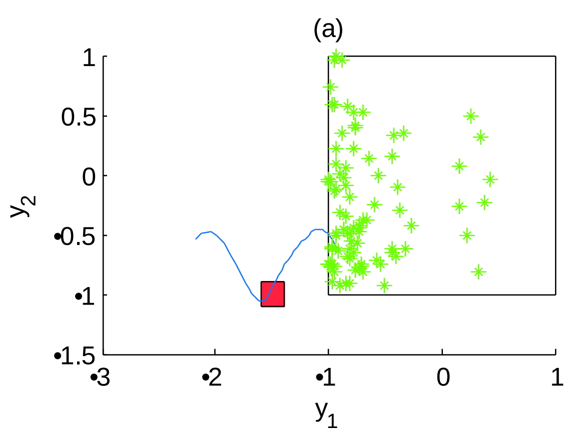
$$dy/dt = y\cos(2\pi y) - x\sin(2\pi x) + 0.1t$$

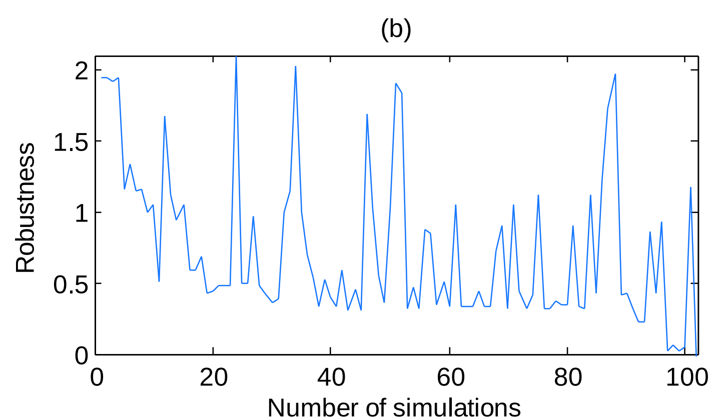
Initial conditions: [-1,1]x[-1,1]

Specification: $G_{[0,2]} \neg a$

where O(a) = [-1.6,-1.4]x[-1.1,-.9]







Minimizing Temporal Logic Robustness

We need to solve an optimization problem:

$\min [\![\varphi]\!](y)$
$y \in Y$ *is the set of all observable trajectories of the hybrid system*

$\min E([\![\varphi]\!](y))$
$y \in Y$ *is the set of all observable trajectories of the stochastic hybrid system*

Challenges:

- Non-linear system dynamics
- Unknown input signals
- Unknown system parameters
- Non-differentiable cost function
  - not known in closed form
  - needs to computed

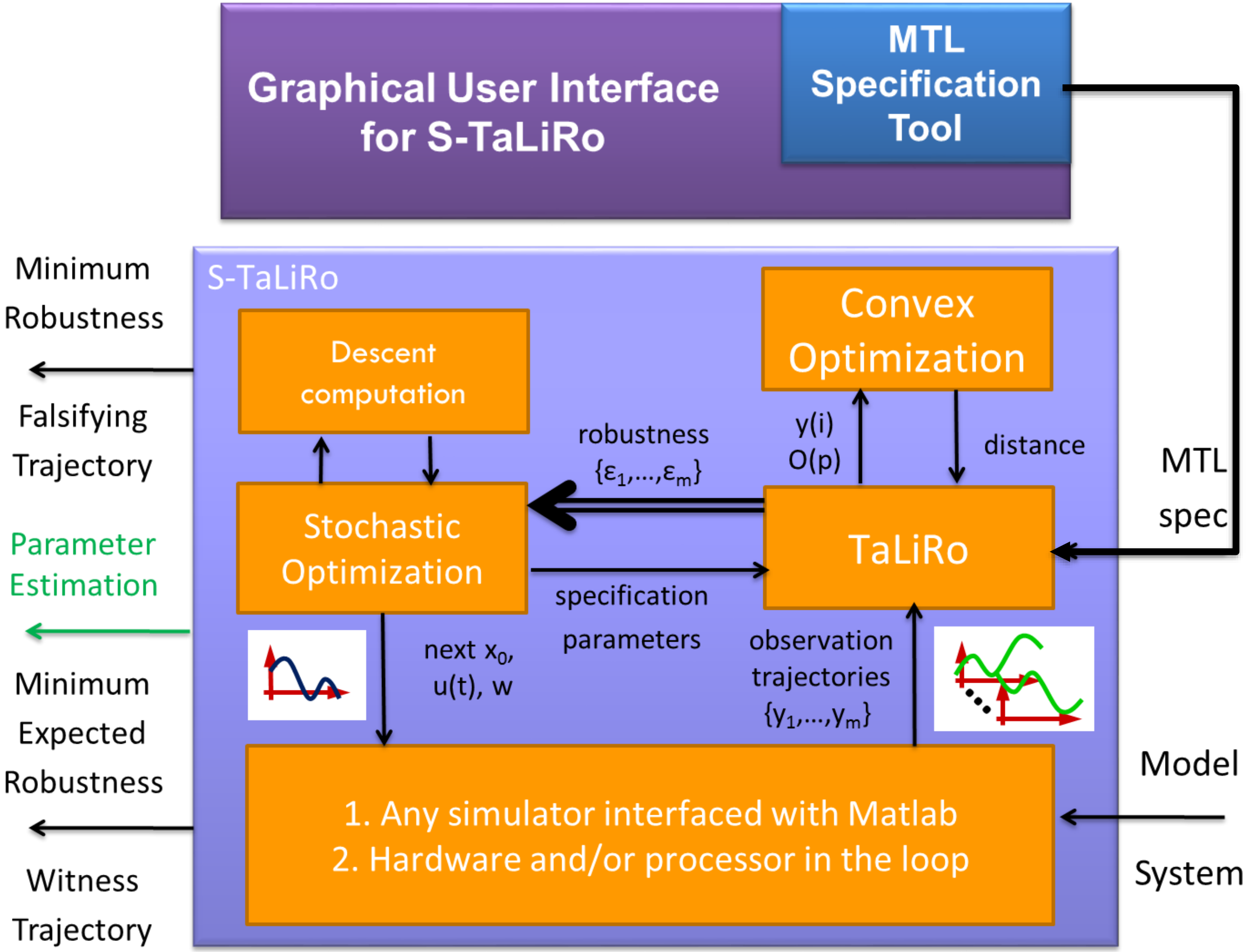
Solution:

- *Stochastic Optimization & Metaheuristics [HSCC 2010]*
- *Gradient Descent [ACC 2013]*

Guarantees:

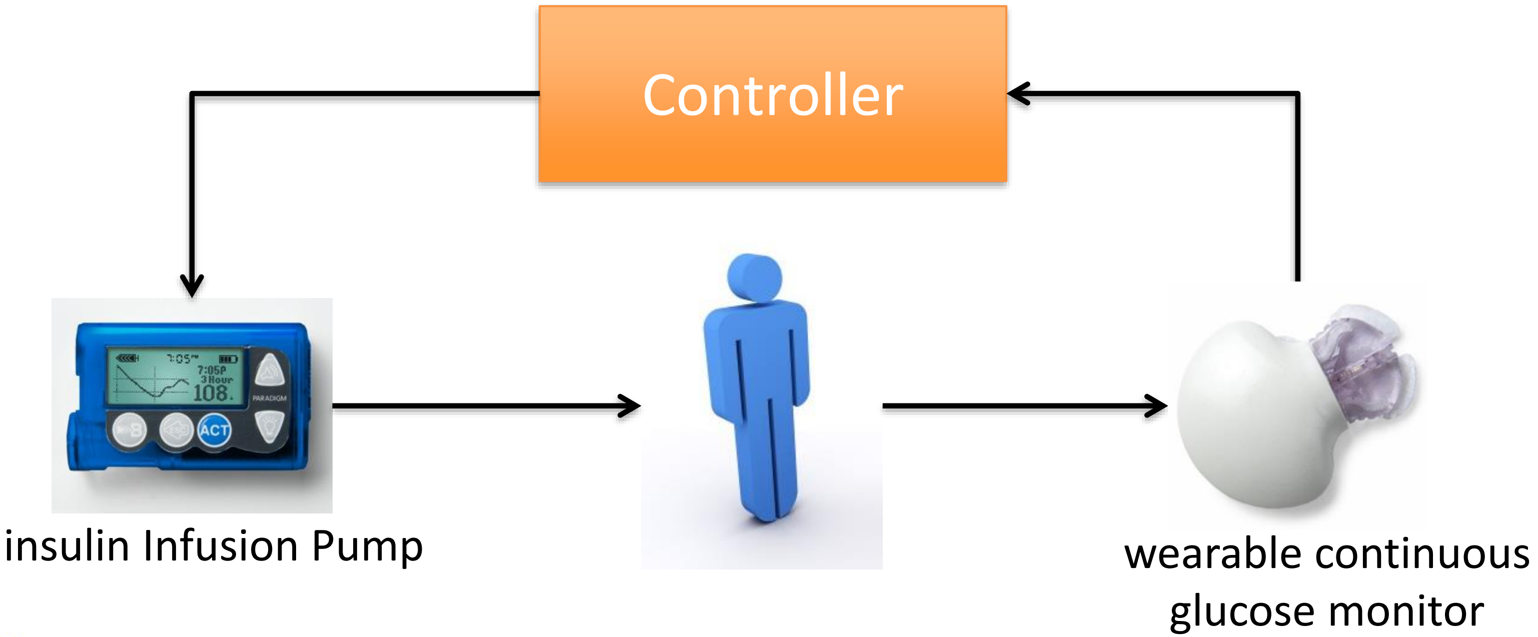
- *Probabilistic convergence if bad behavior is of nonzero measure [Allerton 2012]*
- *Coverage metrics [EMSOFT 2015]*

S-Taliro Architecture



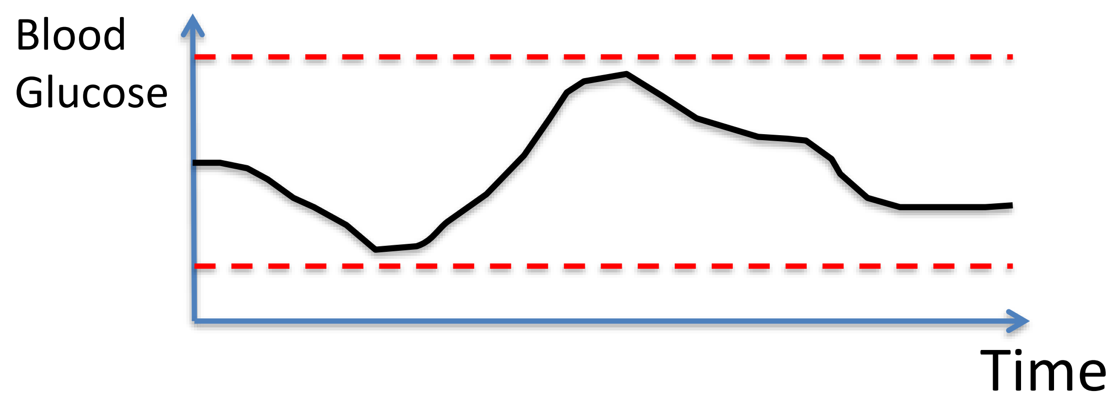
Tools at: https://sites.google.com/a/asu.edu/s-taliro/

Medical Devices: Artificial Pancreas



Awards: 1017074, 1319560, 1350420

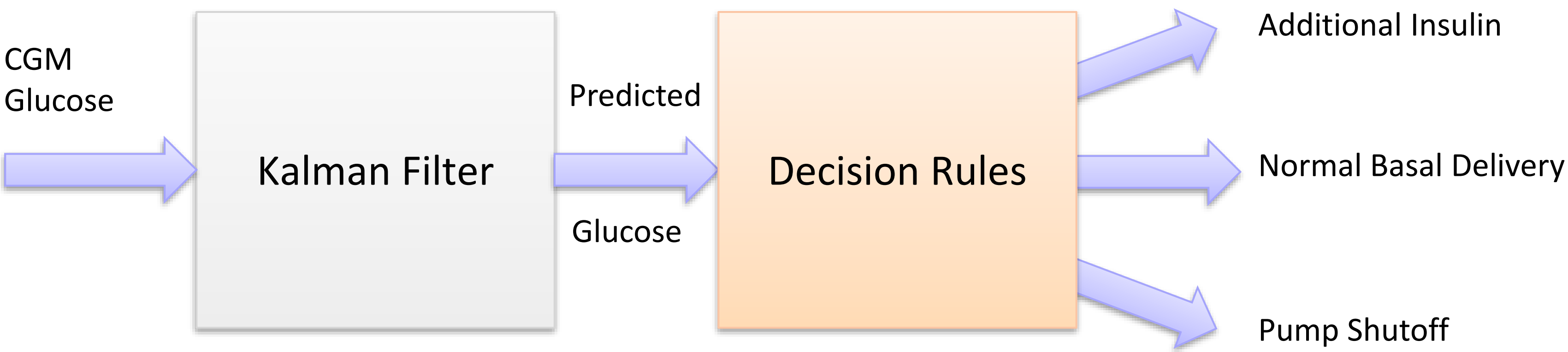
Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.



[Sankaranarayanan, Fainekos, CMSB 12]

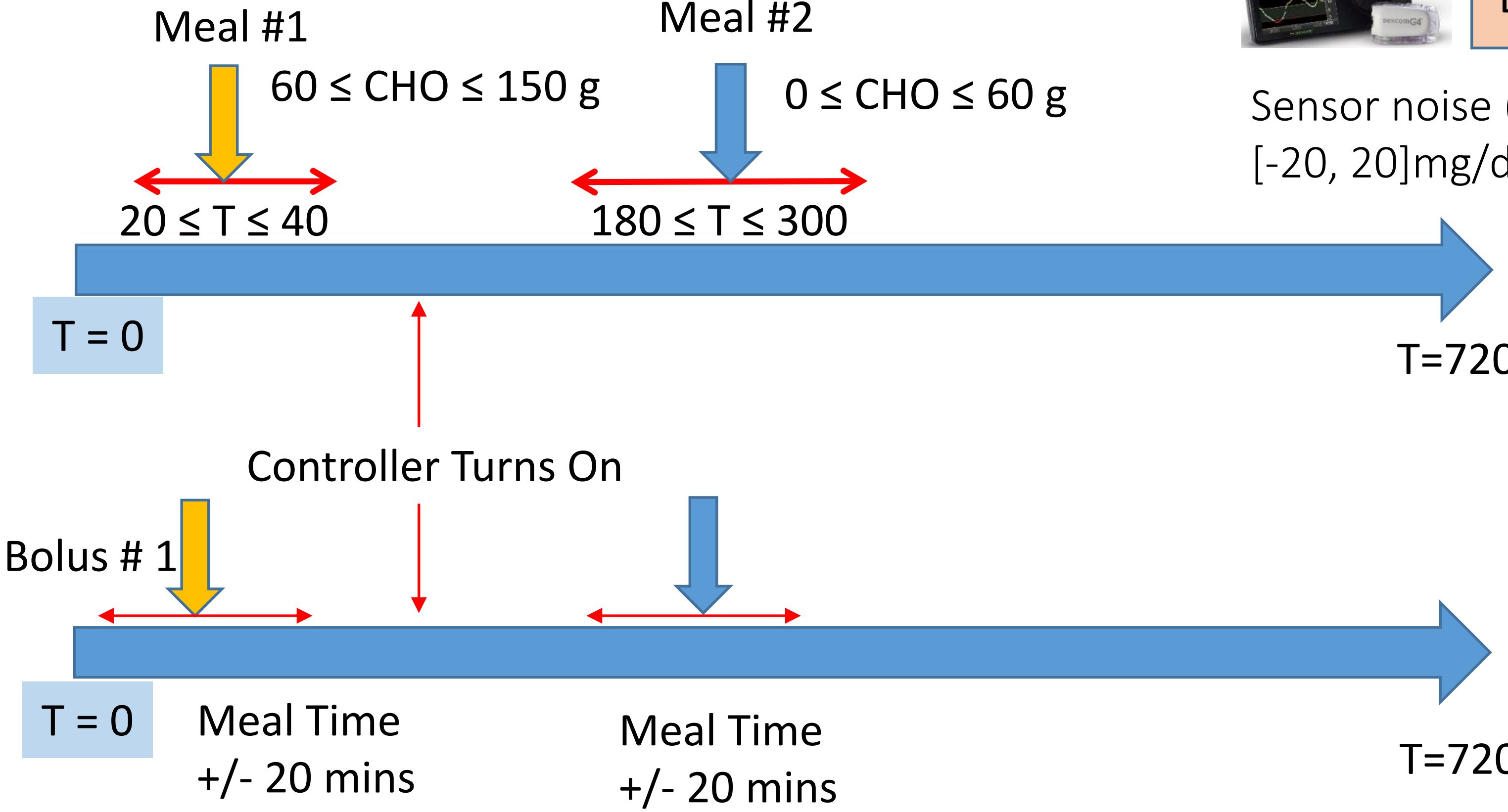
[Sankaranarayanan, Fainekos, HSCC 12]

Case-Study : Kalman Filter Based Hypo/Hyper Mitigation



[Cameron et al. 12, Maahs et al. 15]

In-Silico Study Setup



Disturbances

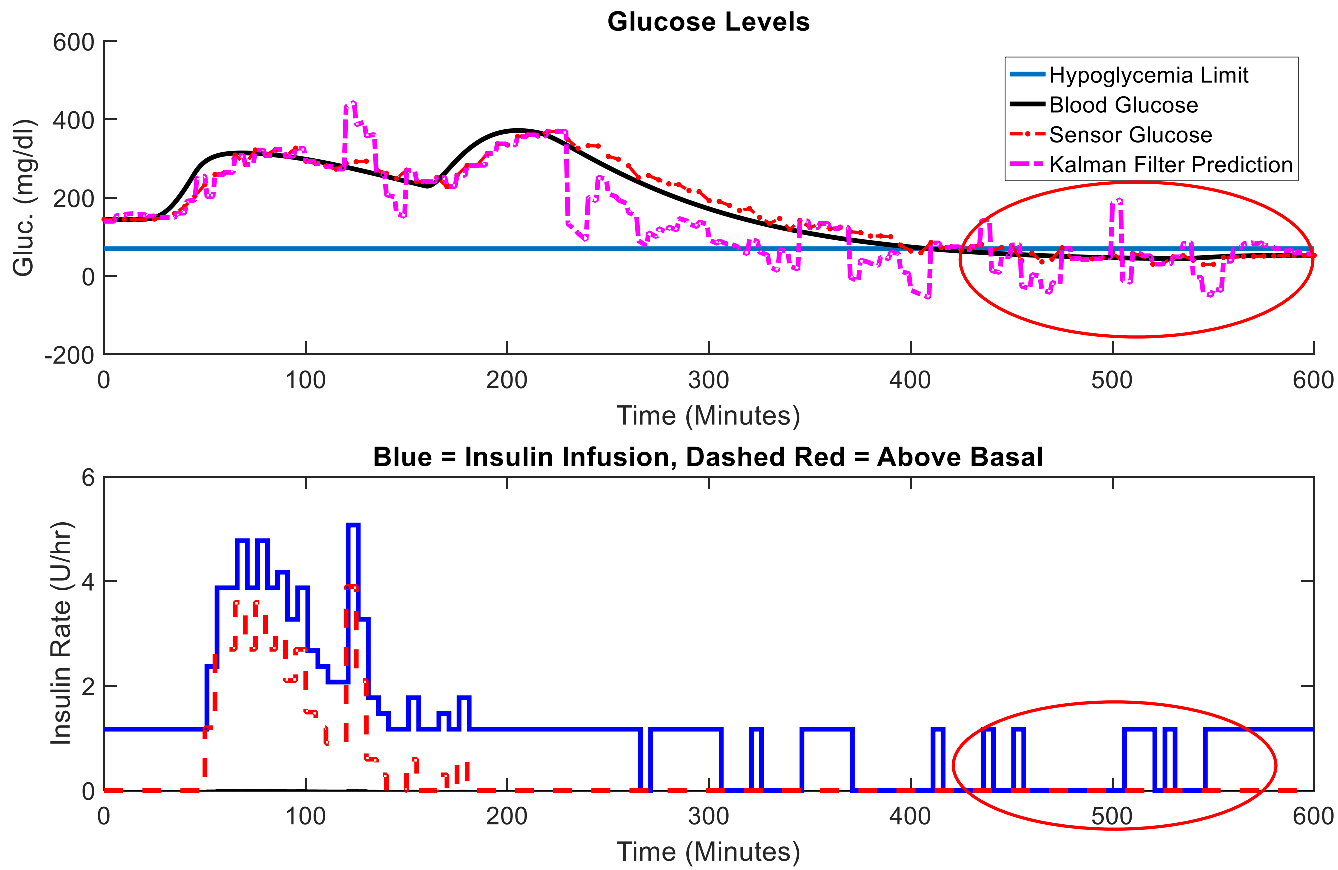
Sensor noise (~ 120 inputs)
[-20, 20]mg/dl

Open loop basal

[0.1, 2]U/hr

P1.1: Can insulin delivery resume under hypoglycemia?

Is it possible for basal insulin to resume when G <= 70 mg/dl while the total shutoff time and the shutoff time within the current time window are still below their upper limits?



S-Taliro ran for nearly 2 hours and 5 minutes and found 5 violations.

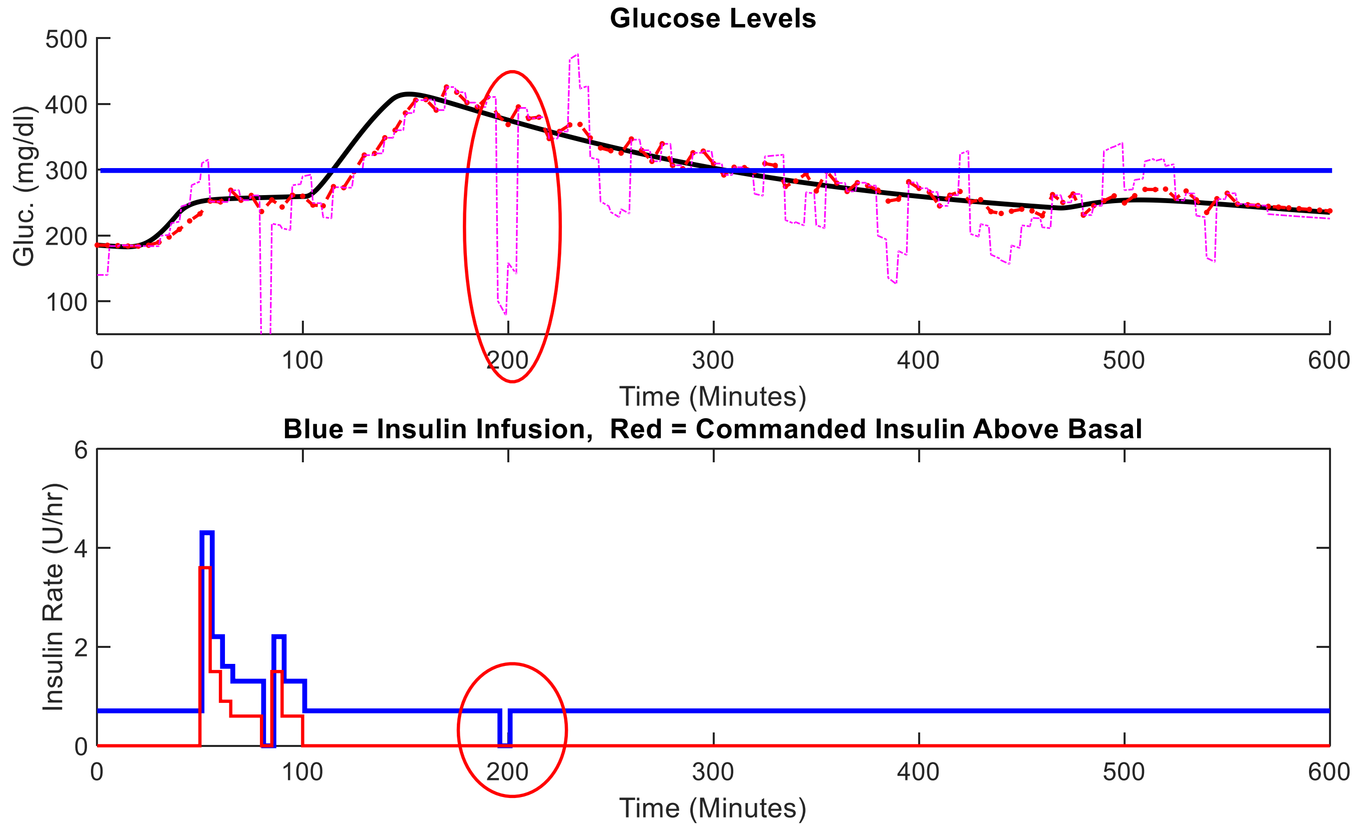
[Sankaranarayanan et al, Medical CPS, 2016]

P2.1-2.3: Safety issues related to hyperglycemia

P2.1 Can the pump be shutoff when G > 300mg/dl?

P2.2 Can the total time under hyperglycemia G>180mg/dl exceed 70% of the total simulation time?

P2.3 Can the total time under hyperglycemia G>300mg/dl exceed 3hrs?



S-Taliro ran for nearly 1 hour and 6 minutes to discover 5 violations for property P2.1.

Trace violates all P2.1-P2.3.

Trial in Actual Control Model (Past defect case)

Detect following defect on SiLS model including all engine control
“monitor value－request value>50” continue over 500msec

There are 75 Control point

Generated input
Gas pedal[%]
Brake[%]
Shift{P,N,D}
Water temp[℃]
Air temp[℃]
Air pressure[kPa]
Air conditioner SW

Defect condition
① Specific logic on
② Engine revolution around 4000rpm
③ Satisfy ①,② and specific accelerator operation

Tried 6 large-scale models,
5 models were falsified.

(Past defect case,intential defect by logic developer)

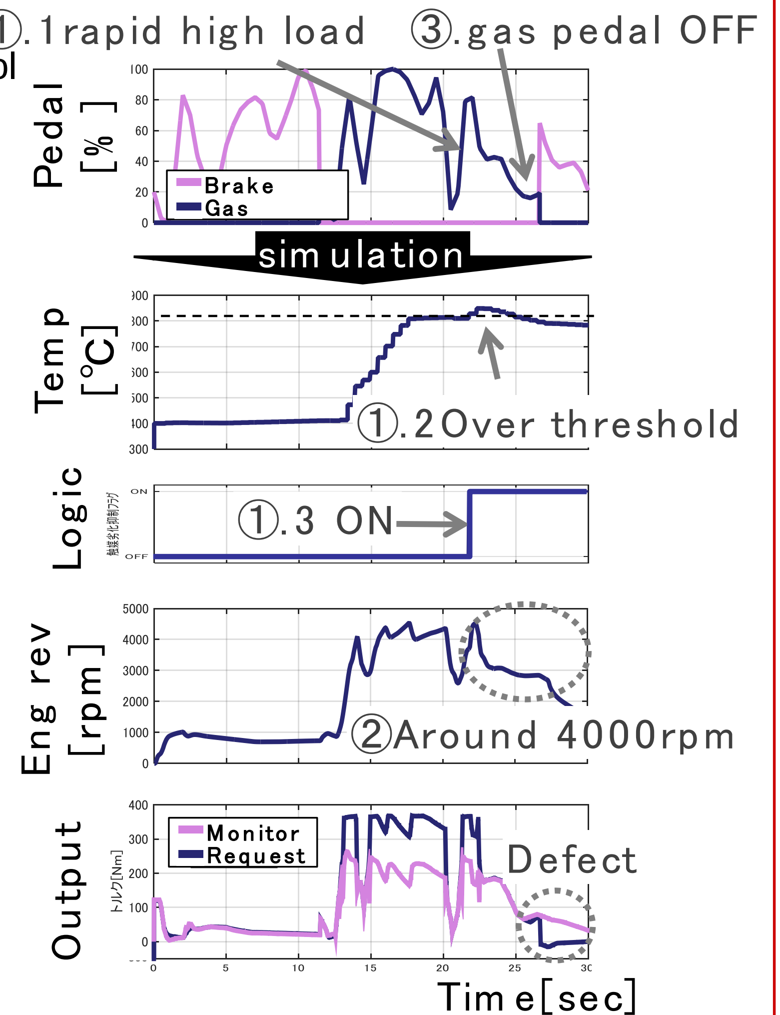
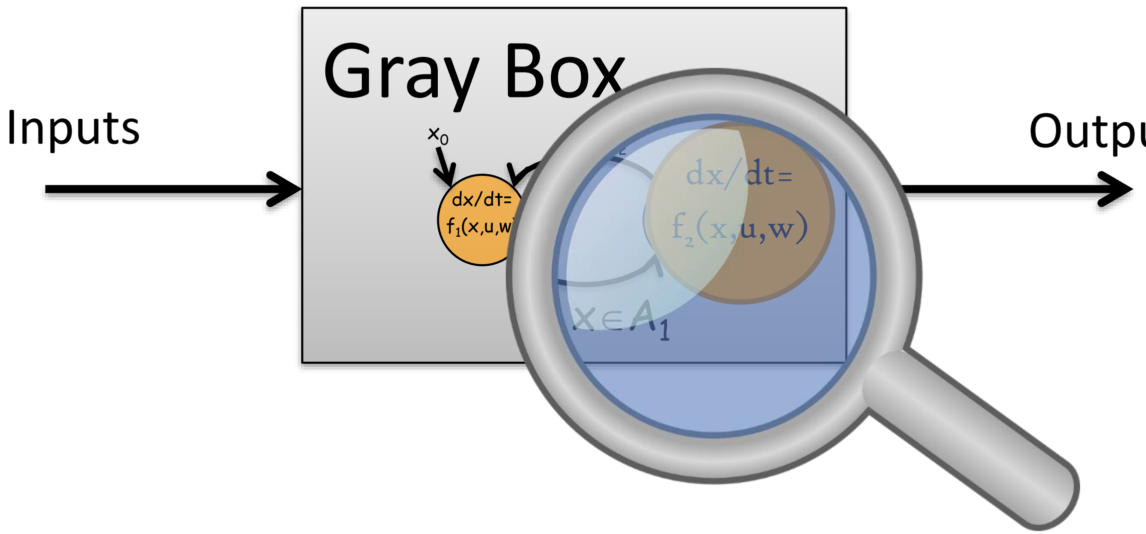


Figure Generated signals automatically

S-Taliro could generate the complicated scenario including the defect



Structural information

GRAY BOX TESTING

Failure Case

Case：Limiting the condition of specific logic to rare case

(ex. temperature threshold up)

Result：After 1000 cycles simulation, defect was not generated.

Objective function change discretely by decision

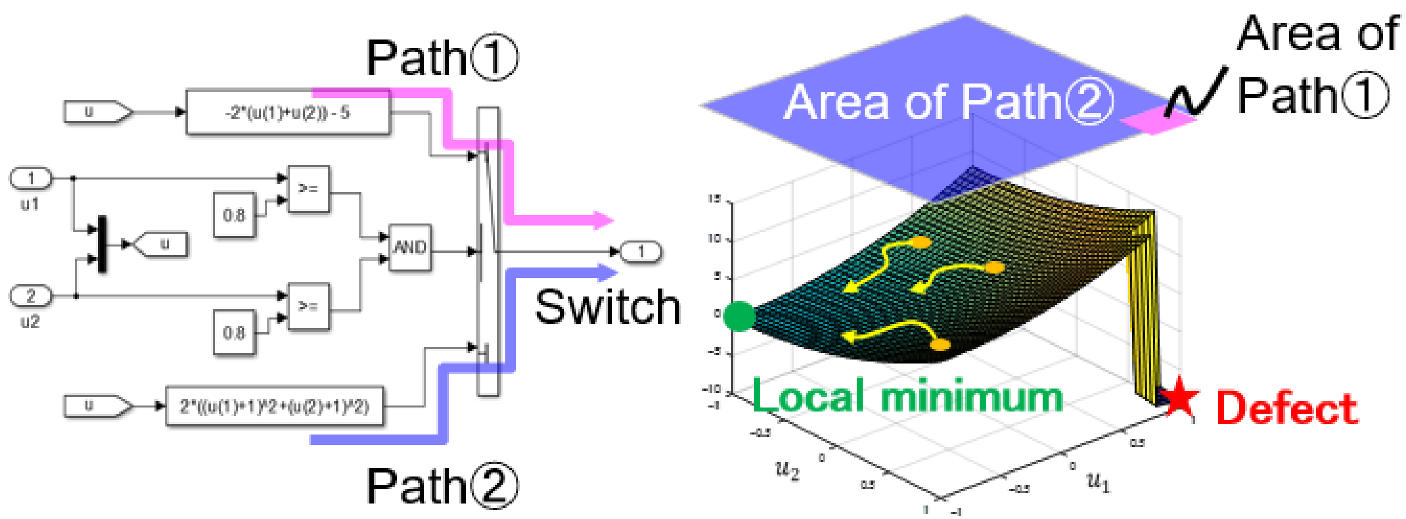
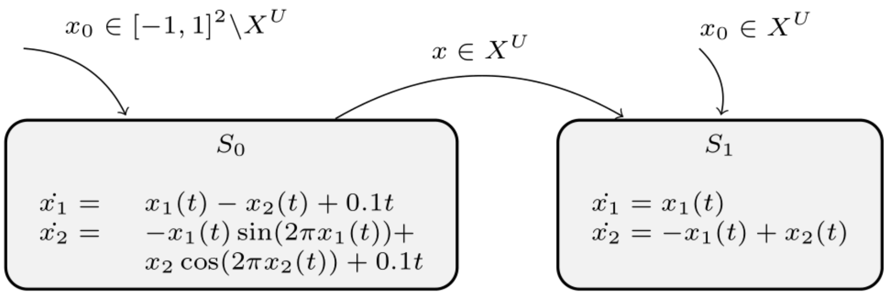


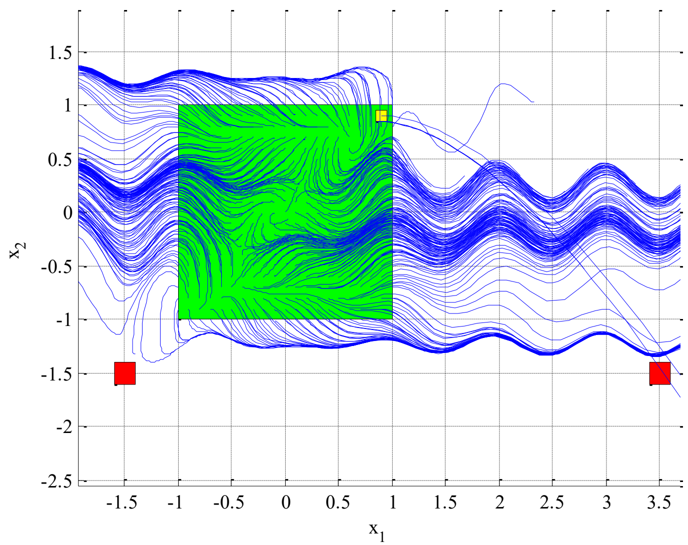
Figure Cause of optimization failure

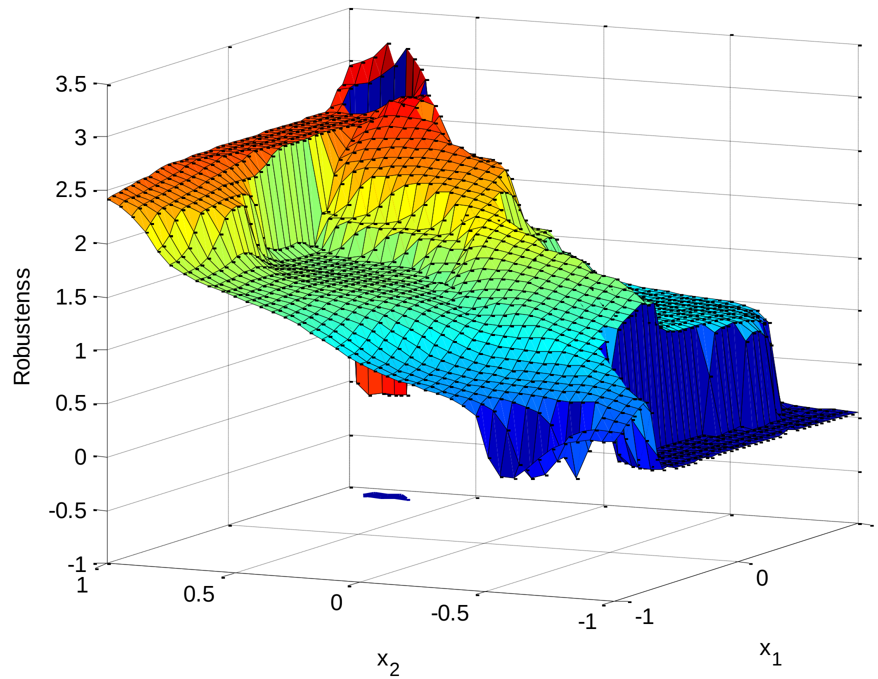
It may overlook defects on rarely exercised paths.

Challenge: Non-Convex Robustness Landscapes



$X^U = [0.85,0.95]^2$ *and* $x_0 \in [-1,1] \times [-1,1]$





Specification:

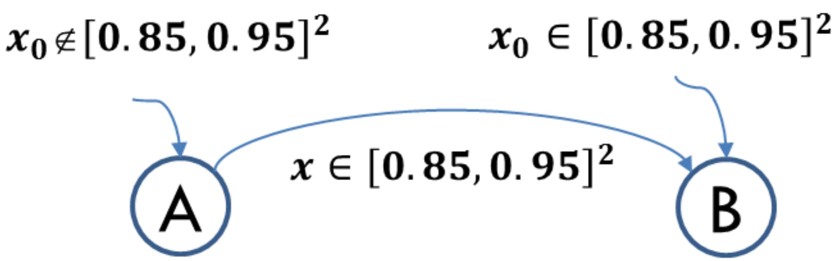
$G_{[0,2]} \neg a \wedge G_{[0,2]} \neg b$

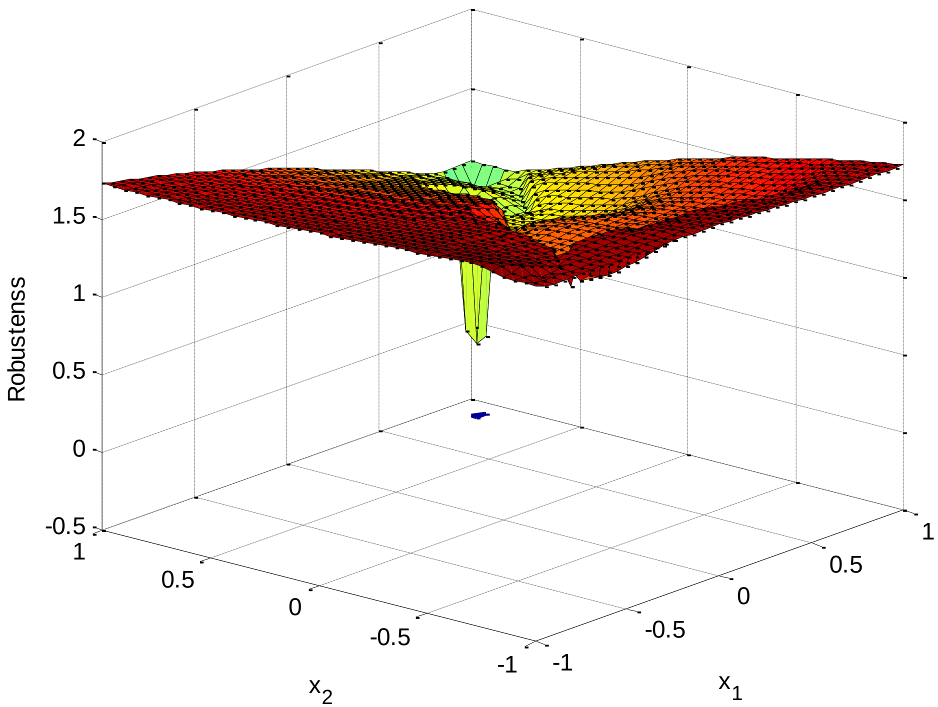
where

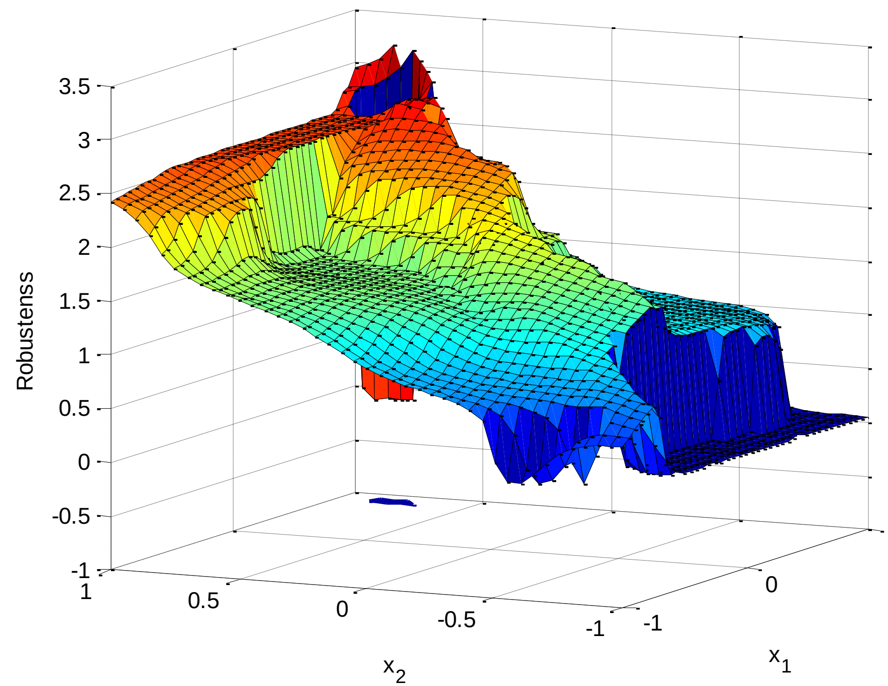
$O(a) = [-1.6,-1.4] \times [-1.6,-1.4]$, $O(b) = [3.4,3.6] \times [-1.6,-1.4]$

Details on how switching conditions can be handled can be found in [EMSOFT 2015]

Observation: What if we knew the "mode of operation" where the error occurs?







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}

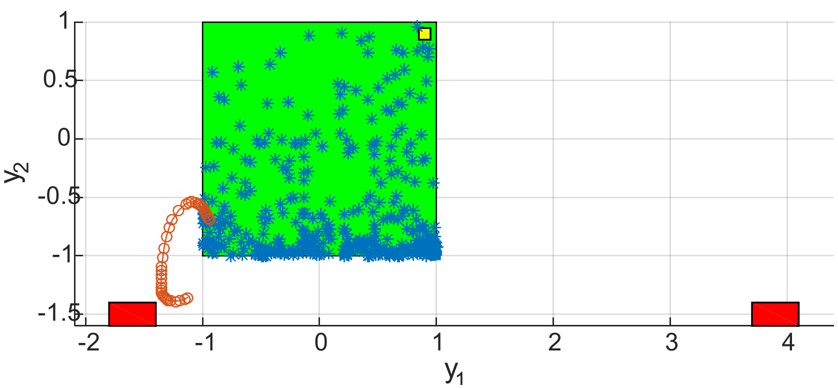
O(b) = [3.4,3.6] x [-1.6,-1.4] x {B}

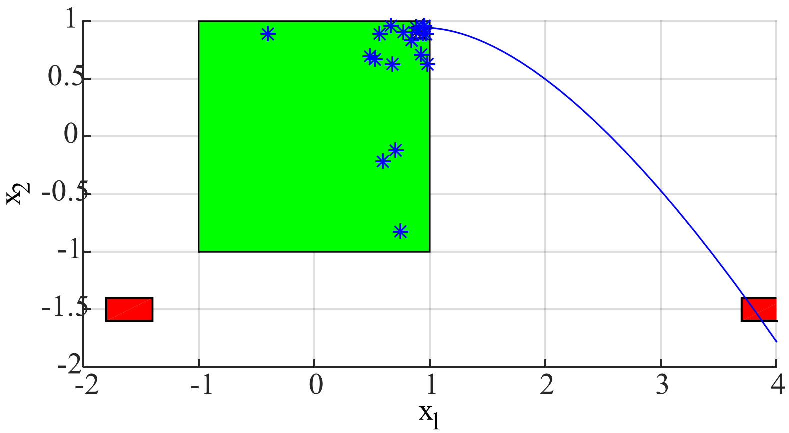
CPS Falsification using software engineering coverage metrics

- **Challenge:** Discrete switching behavior in hybrid systems may hide bugs with low probability of sampling

- **Approach:** Use hybrid distance metrics to bias the search and increase the probability of sampling from the problematic search space

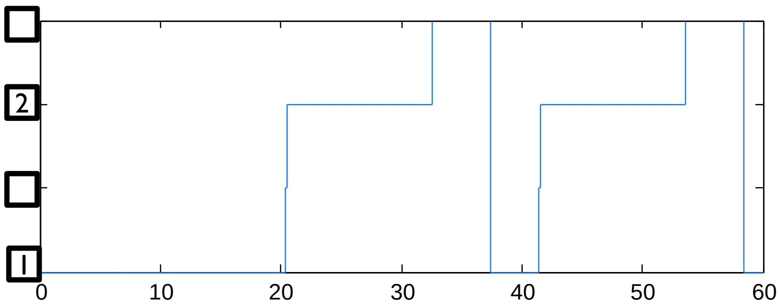
- **Issues to be resolved:**
  1. How to compute hybrid distance metrics in MBD
  2. What coverage metrics to use



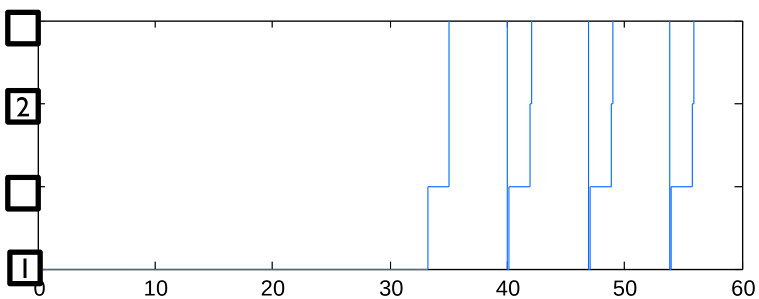


Powertrain Problem (Ford): Falsifications

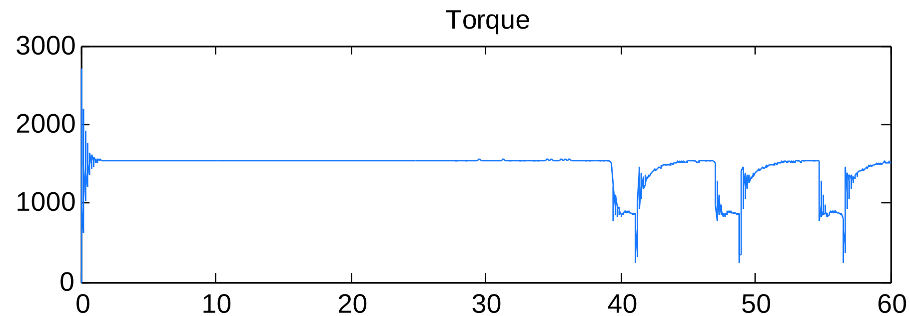
$\varphi_1 = \neg F(gear_2 \wedge F(gear_1 \wedge F gear_2))$

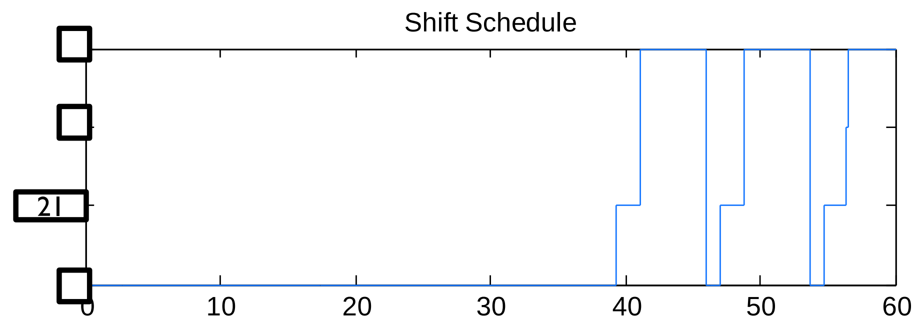


$\varphi_2 = G((\neg gear_1 \wedge X\ gear_1) \rightarrow G_{[0,2.5]}\ \neg gear_2)$



$\varphi_3 = G(gear_{21} \rightarrow |dTs/dt|<450)$

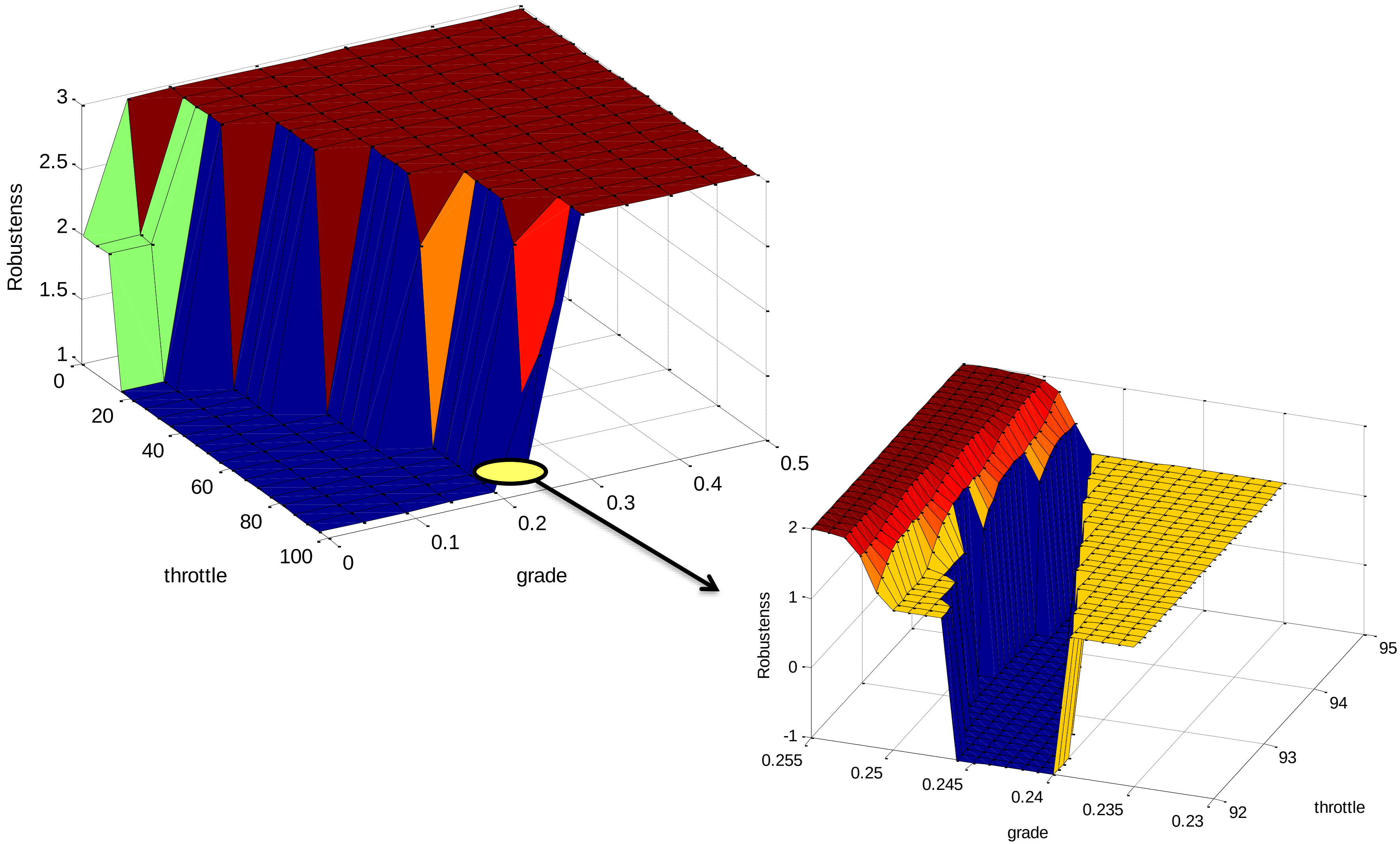




Throttle ≅ 93.9,
Grade ≅ 0.2453

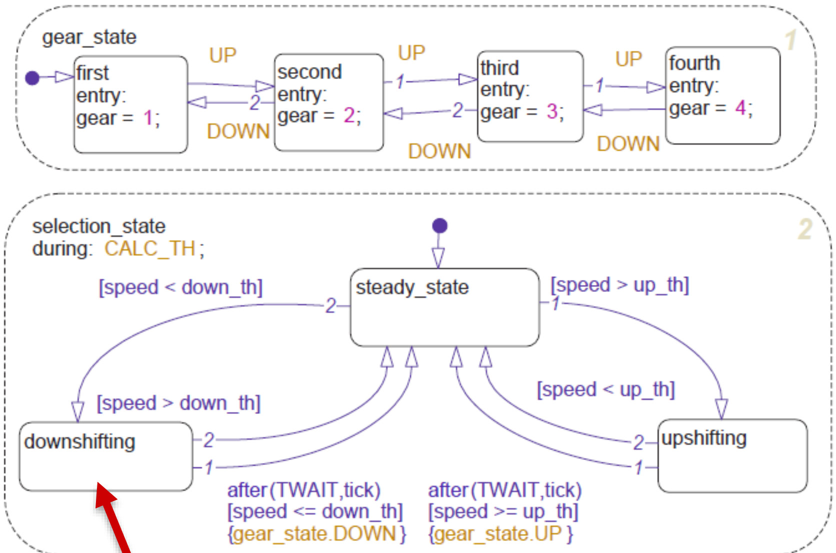
Robustness landscape*

$$\varphi_2 = G((\neg gear_1 \wedge X\, gear_1) \rightarrow G_{[0,2.5]} \neg gear_2)$$

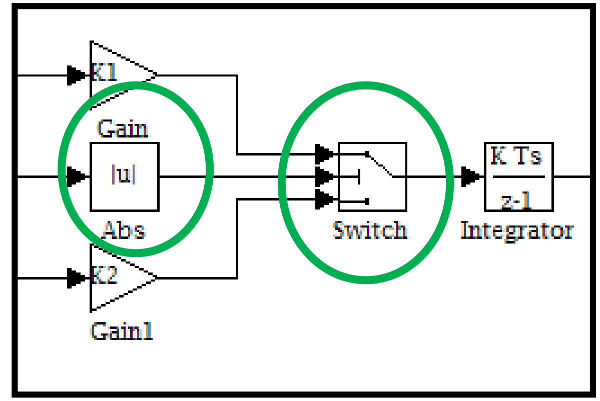


* *Powertrain Challenge Problem by Ford*

Structural Analysis: Extract Global State

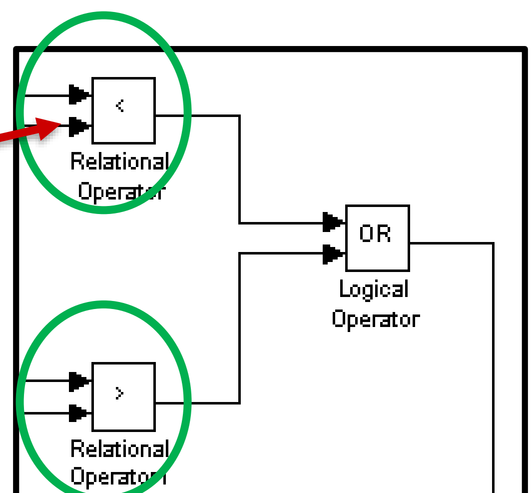


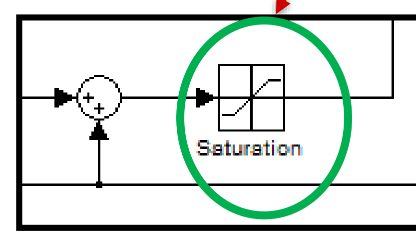
Stateflow Charts



Switch blocks,
Saturation blocks, etc

Assign <u>integer</u> and <u>Boolean</u> variables to identify global state

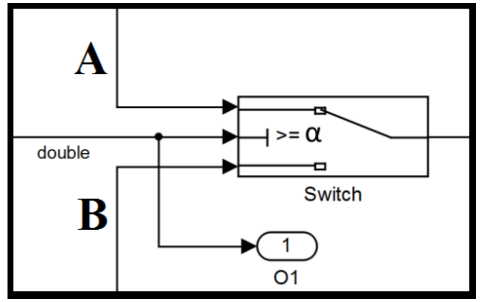


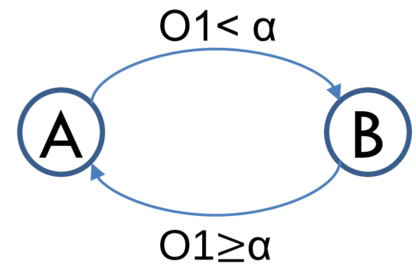


Instrumentation & Coverage Metrics

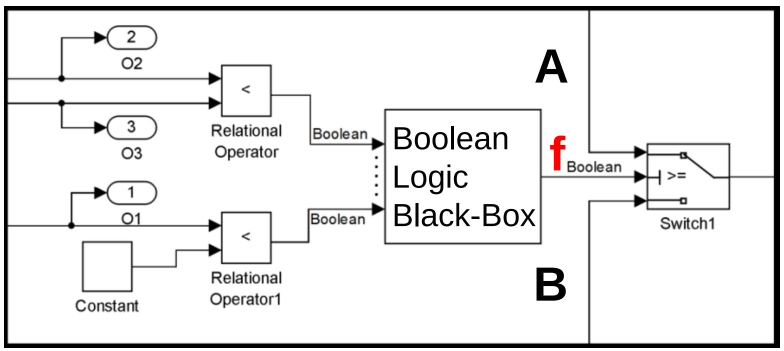
- **State Coverage**

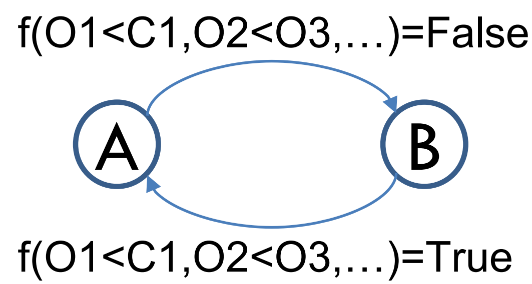
  switch/saturation block



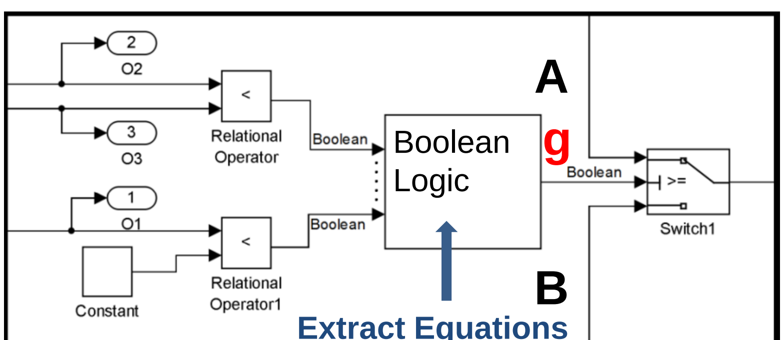


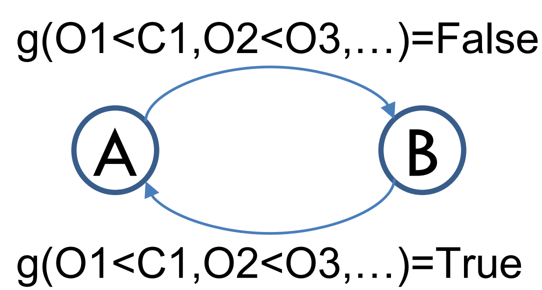
- **Condition Coverage**



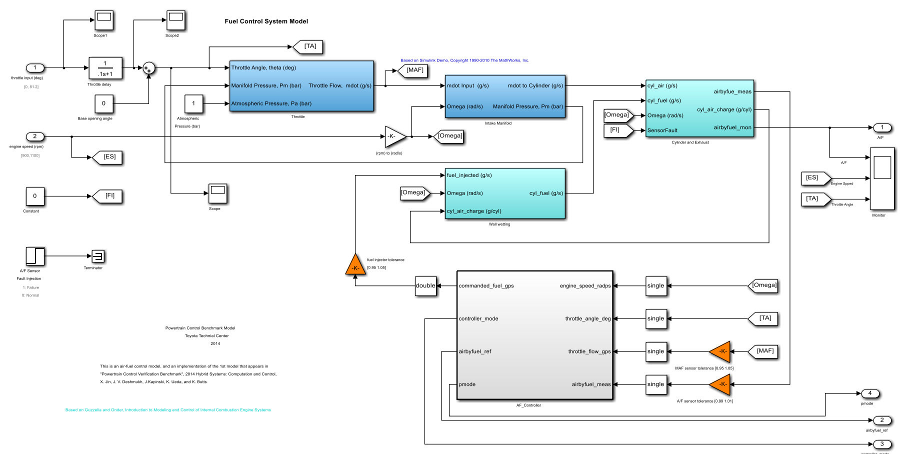


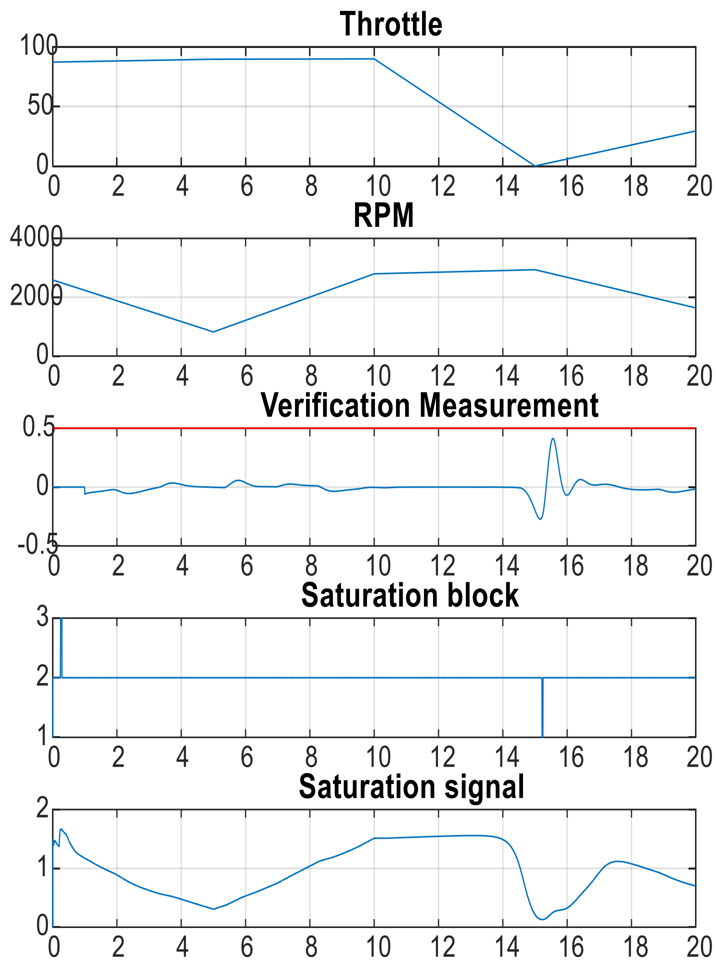
- **Condition coverage that leads to state coverage**





Air Fuel Control Model*





- Logic : 6
- Lookup_n-D : 3
- MinMax : 2
- MultiPortSwitch : 1
- RelationalOperator : 4
- Saturate : 2
- Signum : 1
- Switch : 5
- SwitchCase : 1

} Blocks of interest

** The 1st model that appears in X. Jin et al "Powertrain Control Verification Benchmark", HSCC 2014*

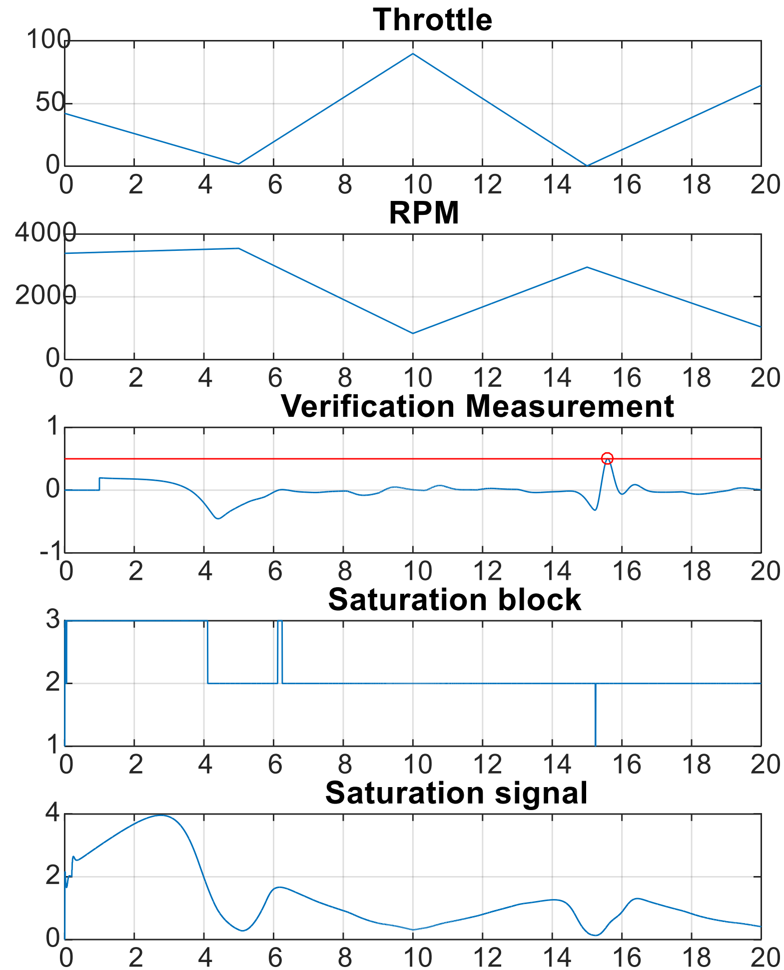
Air Fuel Control Model*

- Uniform random sampling:
  No falsification after 500,000 tests

- S-Taliro with simulated annealing sampling 1 falsification after 4982 tests

- Spec:
  $G_{[0,20]}(VM \leq 0.5) \vee$
  $\neg(F_{[0.1,\infty)} LowerSaturation)$

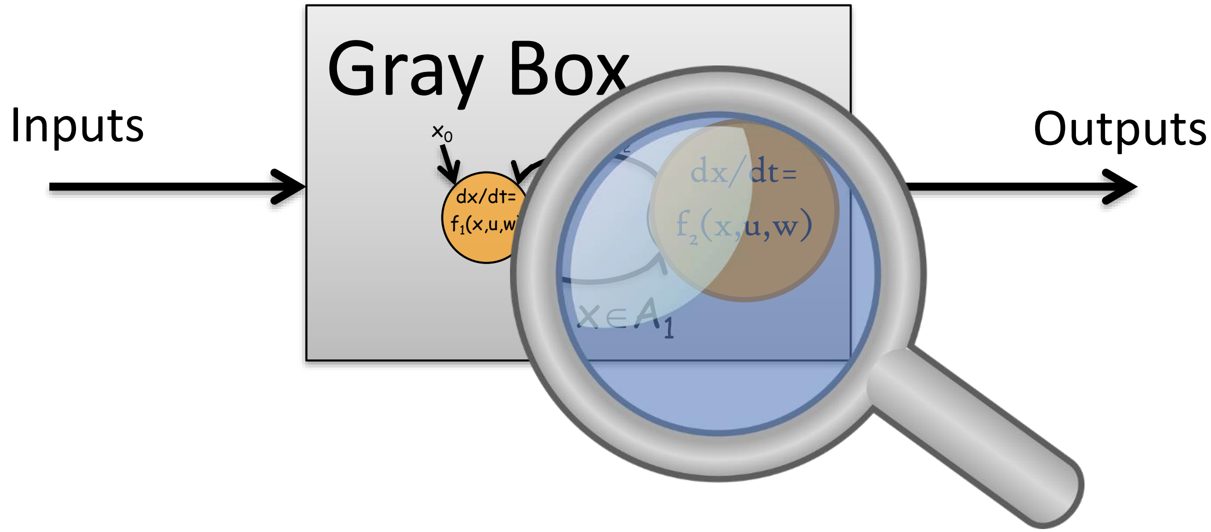
- Value : 0.5000226



*The 1st model that appears in X. Jin et al "Powertrain Control Verification Benchmark", HSCC 2014

Joint work with: H. Abbas (ASU/UPenn)
A. A. Julius (RPI)
S. Yaghoubi (ASU)





System dynamics and structural information

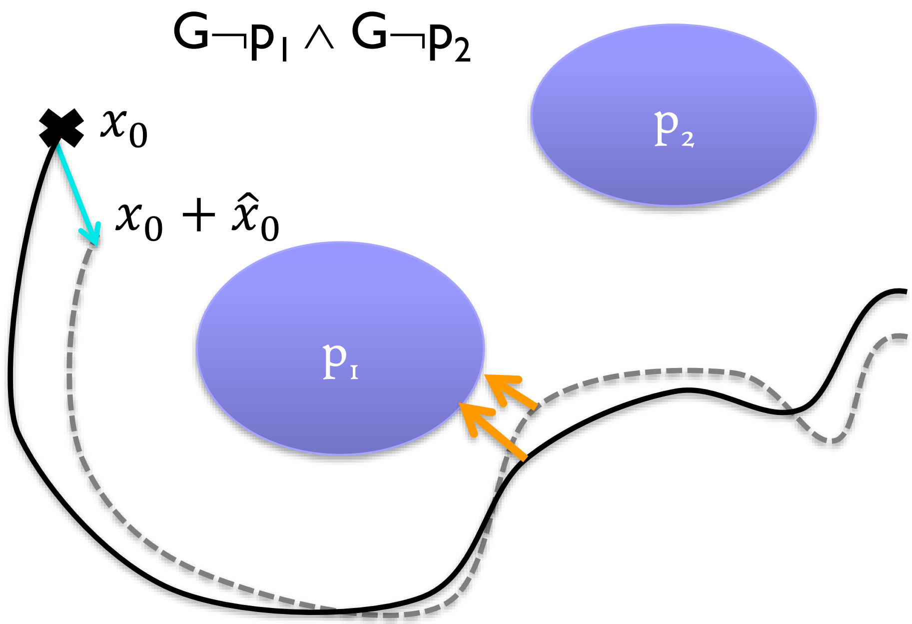
WHITE/GRAY BOX TESTING

Local Descent for Non-Autonomous Smooth Nonlinear Systems

Dynamical system:

$$\dot{x}(t) = F(x(t), t, u(t))$$

Trajectory is uniquely determined by x_0 and $u \in L^2[0,T]$.

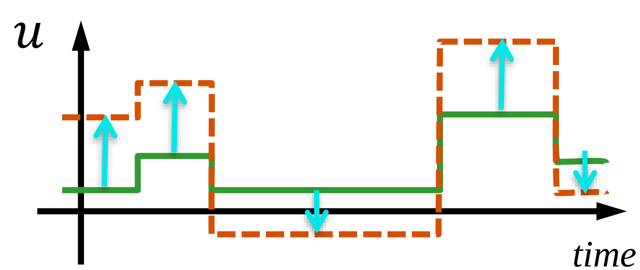


Hence, temporal logic robustness:

$$f_\phi : X_0 \times L^2[0,T] \rightarrow \overline{\mathbb{R}}$$

Our goal is to fine a descent direction s.t.:

$$f_\phi(x_0 + \hat{x}_0, u + \hat{u}) < f(x_0, u)$$



Computation of Gradient Direction in iteration *i*

For $w = (x_0, u)$ Our cost function is given by

$$f_{\phi,i}(w) \triangleq G\left(s_{x_0}(t^*; w)\right) \triangleq \left\|z(t^*; w_i) - s_{x_0}(t^*; w)\right\|$$

Set $\widehat{w} = (\hat{x}_0, \hat{u})$, where

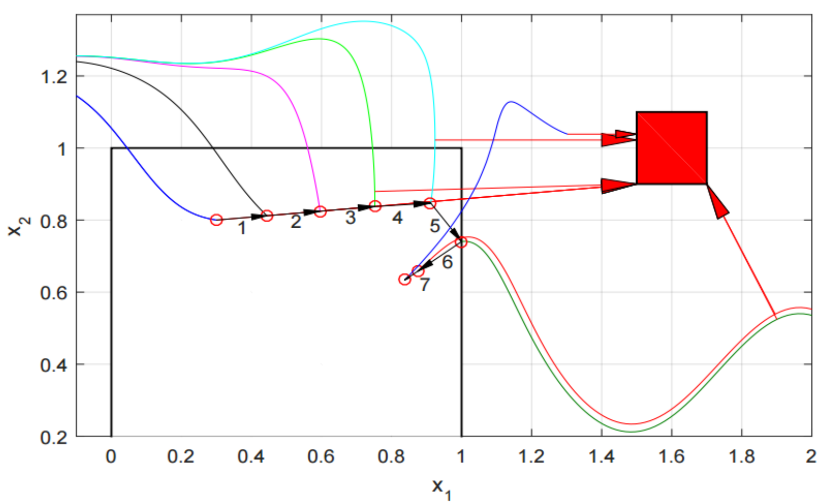
$$\hat{u}(\tau) = -\frac{\partial G}{\partial x} p_u(t^*, \tau),$$

$$\hat{x}_0 = -\frac{\partial G}{\partial x} p_{x_0}(t^*).$$

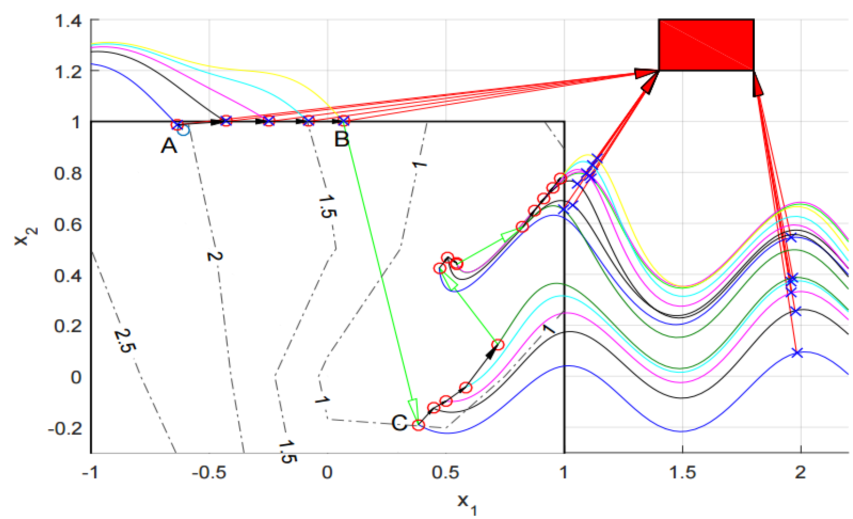
We are guaranteed $f_\phi(w_i + \lambda \widehat{w}) < f_\phi(w_i)$ for a small enough λ

Example: Falsification with descent

$$\dot{x} = \begin{bmatrix} x_1(t) - x_2(t) + 0.1t + u_1(t) \\ x_2(t)\cos(2\pi x_2(t)) - x1(t)\sin(2\pi x_1(t)) + 0.1t + u_2(t) \end{bmatrix}$$

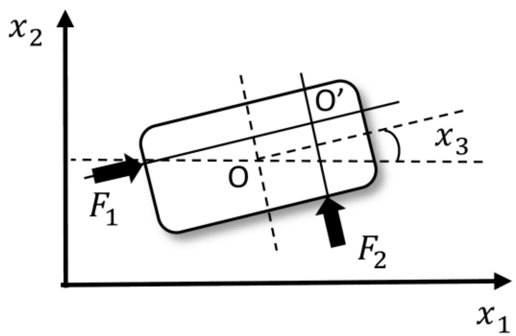


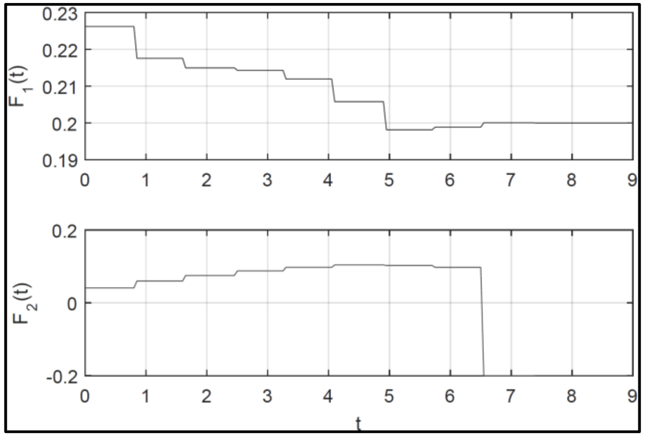
Approximate GD

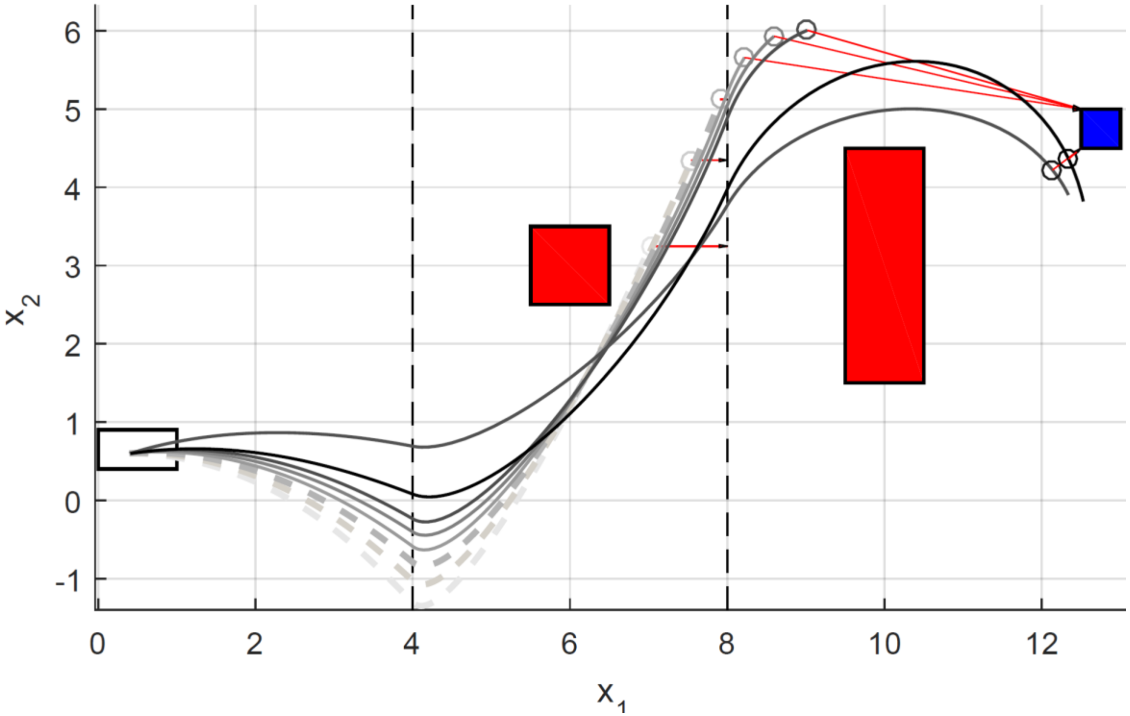


GD+SA

Added benefit of hybrid distance: Local descent for hybrid systems







Overview

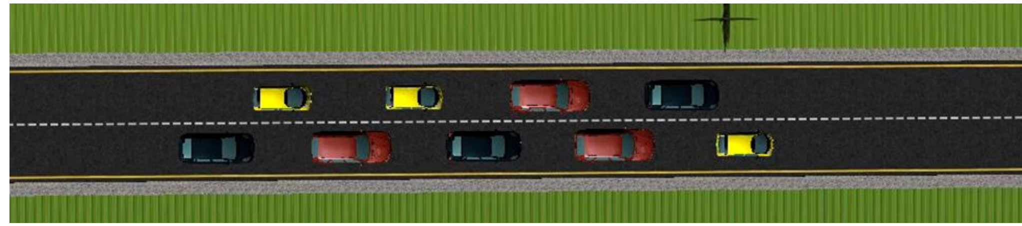
Joint work with
Erkan Tuncali (ASU)
Ted Pavlic (ASU)

- Motivation
  - Quick intro to control synthesis challenges
  - Model Based Development
- Formal requirements for CPS
- Requirements driven falsification
- Autonomous vehicle testing
- Parameter mining in requirements
- Conformance testing
- Testing based verification
- Vision, Other topics & Future work

Tuncali, Pavlic, Fainekos,
Utilizing S-TaLiRo as an Automatic Test Generation Framework for Autonomous Vehicles,
IEEE Intelligent Transportation Systems Conference, 2016

Four vehicles joining a platoon in a distributed and decentralized way

4 vehicles joining a 5 car platoon using a decentralized protocol



High level defined with π-calculus expressions

$\text{Wait}(y) = y.\overline{merge\_done}$

$\text{Align}(y) = \overline{align\_start}.align\_done.\overline{y}.\text{Wait}$

$\text{Rcv\_Ldr}(y, ldr) = y(nldr).\overline{set\_ldr}\langle nldr\rangle.\text{Align}(y)$

$\text{Send\_Ldr}(y) = get\_ldr(ldr).\overline{y}\langle ldr\rangle.\text{Rcv\_Ldr}(y, ldr)$

$\text{Respond}(y, flag) = flag : [True \Rightarrow \text{Send\_Ldr}(y)]$

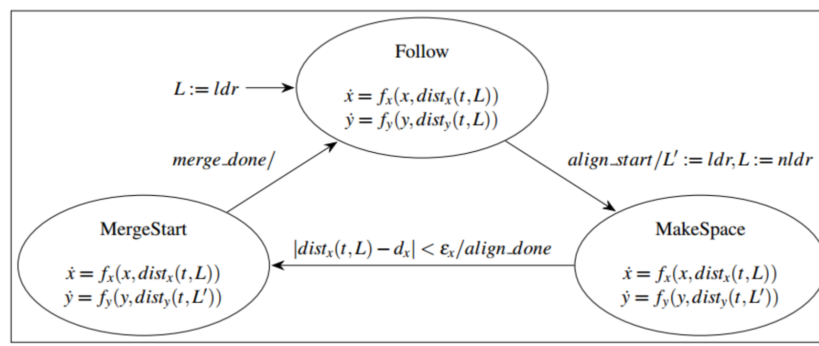
$\text{Ident}(y) = get\_id(id).\overline{y}\langle id\rangle.y(flag).\text{Respond}(y, flag)$

$\text{Cooperate} = !\mathbf{r}(x).(\nu y)(\overline{x}\langle y\rangle.\text{Ident}(y))$

$\text{Follow} = \overline{keep\_dist}.\text{Follow}$

$\text{Follower} = \text{Follow}||\text{Cooperate}$

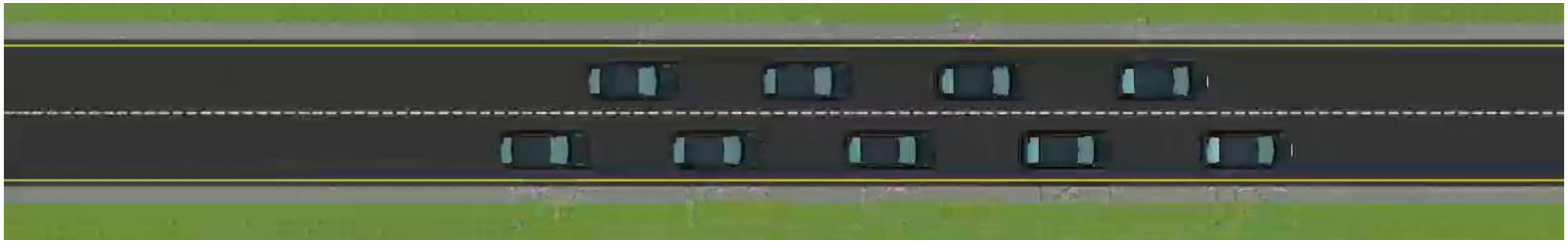
Low level defined with hybrid automata



(b) Follower HA

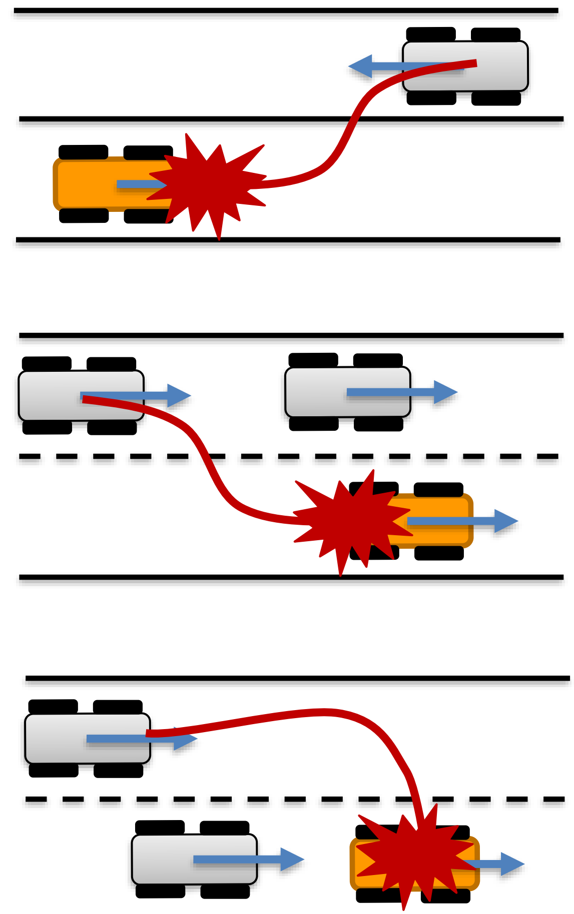
Campbell, Tuncali, Liu, Pavlic, Ozguner, Fainekos, *Modeling Concurrency and Reconfiguration in Vehicular Systems: A π-calculus Approach*, IEEE CASE, 2016

What can go wrong?



There are always worst case scenarios that we cannot avoid …

Where is the boundary between safe and unsafe scenarios?



Our claim:

We need to detect and robustify "boundary" situations.

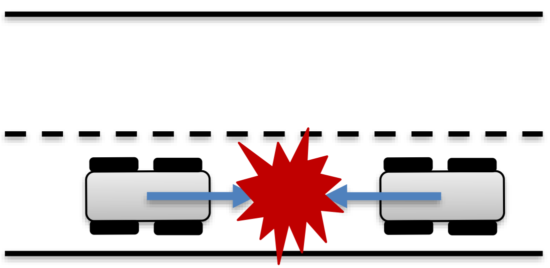
Robustness Metric

Used for guiding the tests to the boundaries of safe and unsafe scenarios.

Collision:

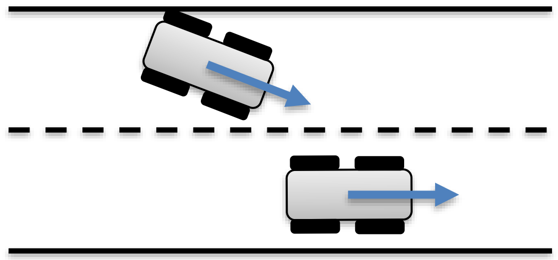
Severity of collision

(relative speed at the collision)



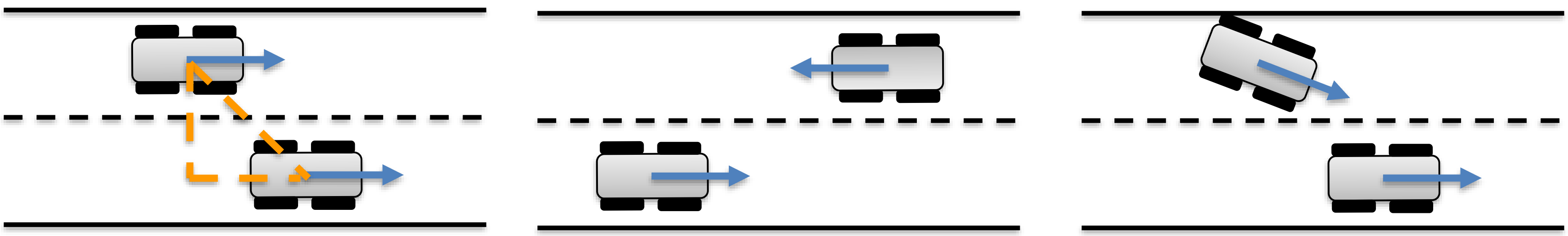
No collision:

Risk of collision



How do we measure robustness?

aka How can we tell that we are approaching a problematic behavior?



Euclidean distance or other norms do not work

Lane dependent measures also do not work

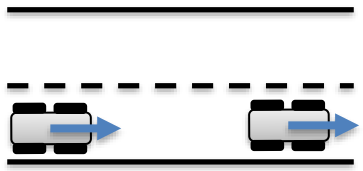
TTC: ∞

TTC: small number

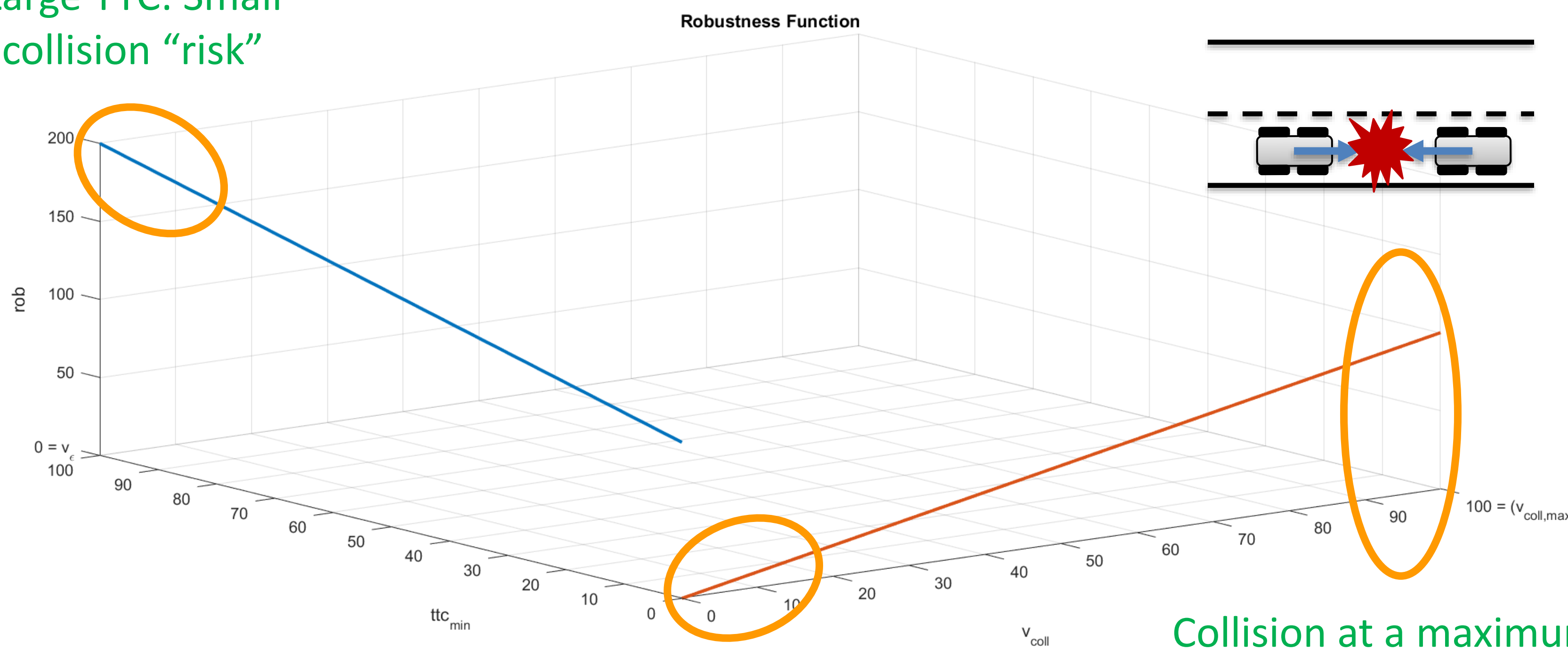
Time-to-Collision (TTC*) :
Time required to collision with current heading and velocity

*J. C. Hayward, "Near-miss determination through use of a scale of danger," Highway Research Record, no. 384, 1972.

Robustness Metric

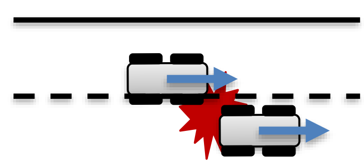


Large TTC: Small collision “risk”



Collision at a small relative velocity

Collision at a maximum relative velocity



An example robustness function.

Maximum possible collision speed = 100

Robustness Metric

Goal is to find boundaries between safe and unsafe behavior!

Minimizing the robustness function should guide the search towards the boundary

Time-to-Collision (TTC)*: Time required to collision with current motion

$$\mathcal{R}(y) = \begin{cases} v_{coll,y} - v_{\epsilon} & \text{, collision detected in } y \\ ttc_{min,y} + v_{coll,max} & \text{, otherwise.} \end{cases}$$

Robustness function: $\mathcal{R}$

Simulation trajectory: y

Collision speed (relative): $v_{coll,y}$

Minimum collision severity: v_{ϵ}

Minimum TTC in the trajectory: $ttc_{min,y}$

Maximum possible collision speed: $v_{coll,max}$

*J. C. Hayward, "Near-miss determination through use of a scale of danger," Highway Research Record, no. 384, 1972.

Case Study

Simulation Configuration:

- Two vehicles under test
- One dummy vehicle
- Two-lane straight road

Simulation Engine:

- Simulates VUT using a vehicle dynamic model
- Simulates dummy vehicles using a kinematic model
- Implemented in MATLAB (Can be changed to another platform)

Initial Conditions:

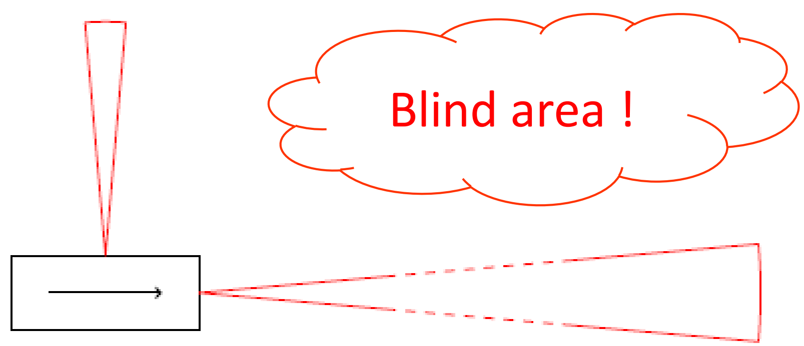
- Both VUT on the right lane separated by a distance
- The dummy vehicle is on the left lane next to one of the VUT

Case Study

Vehicle Configuration:

- Critical points define the vehicle (corners, sensor positions etc.)
- Sensor locations, orientation and ranges are defined
- VUT controlled by a Model Predictive Controller
- Dummy vehicles are controlled by a PID controller

A side sensor with 5m range and 10° sensing angle

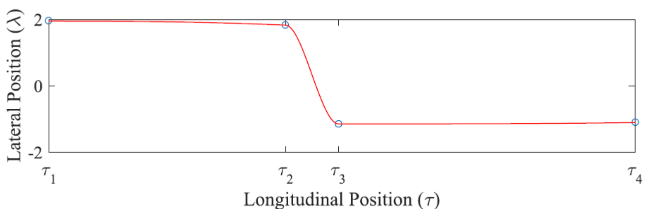


A front sensor with 10m range and 10° sensing angle

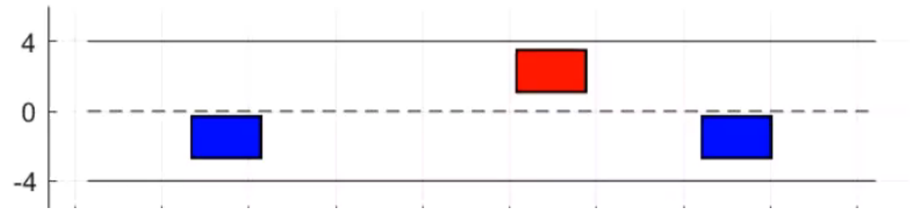
Case Study

Experiment Results:

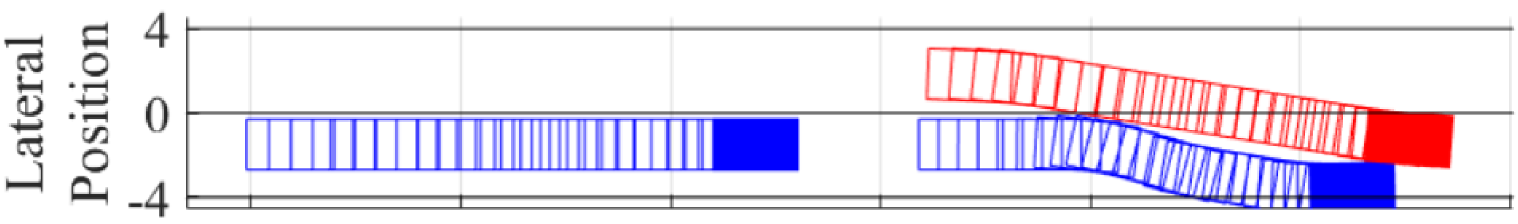
Seeking a trajectory for the dummy vehicle which causes a behavior at the boundary between collision and no-collision operations. (A very slow speed collision or a very near miss)



(The trajectory for the dummy vehicle)



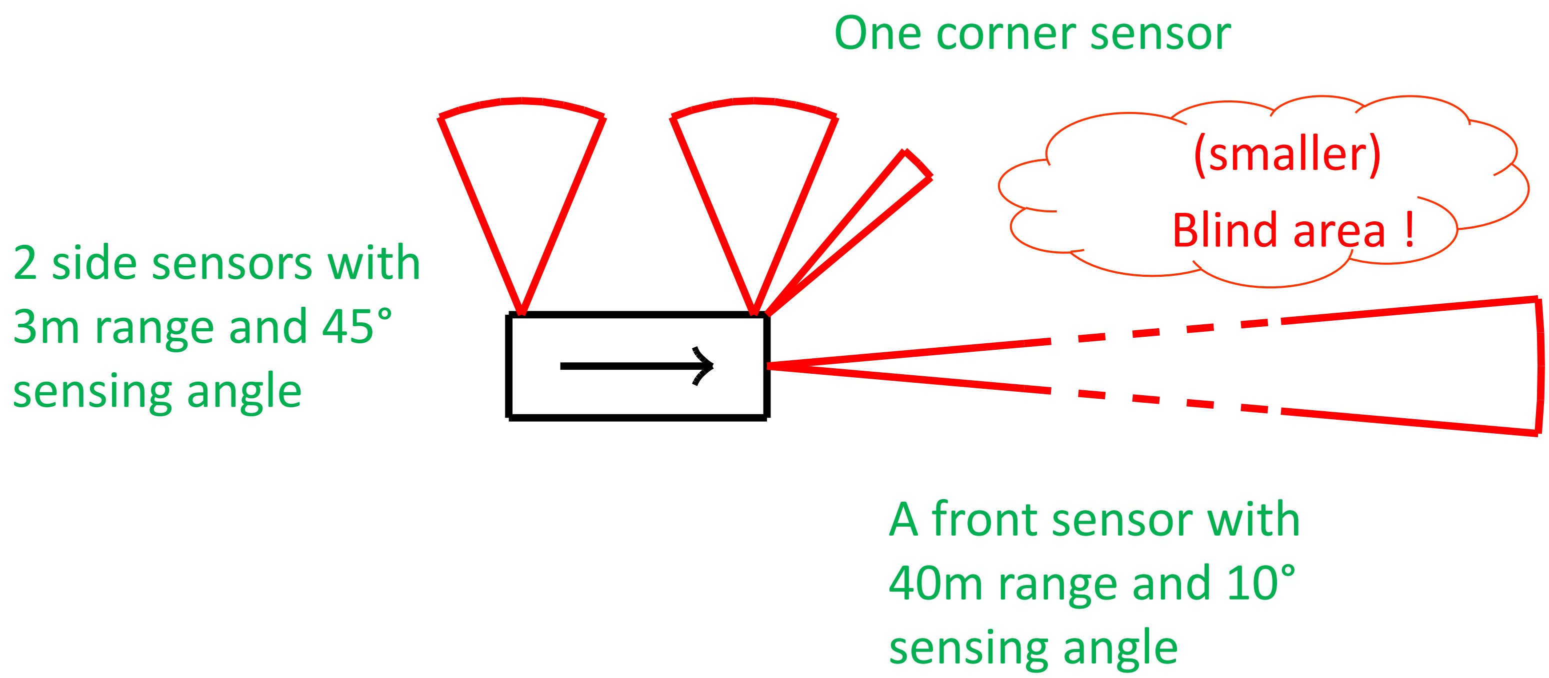
(A front collision right after avoiding a side collision)



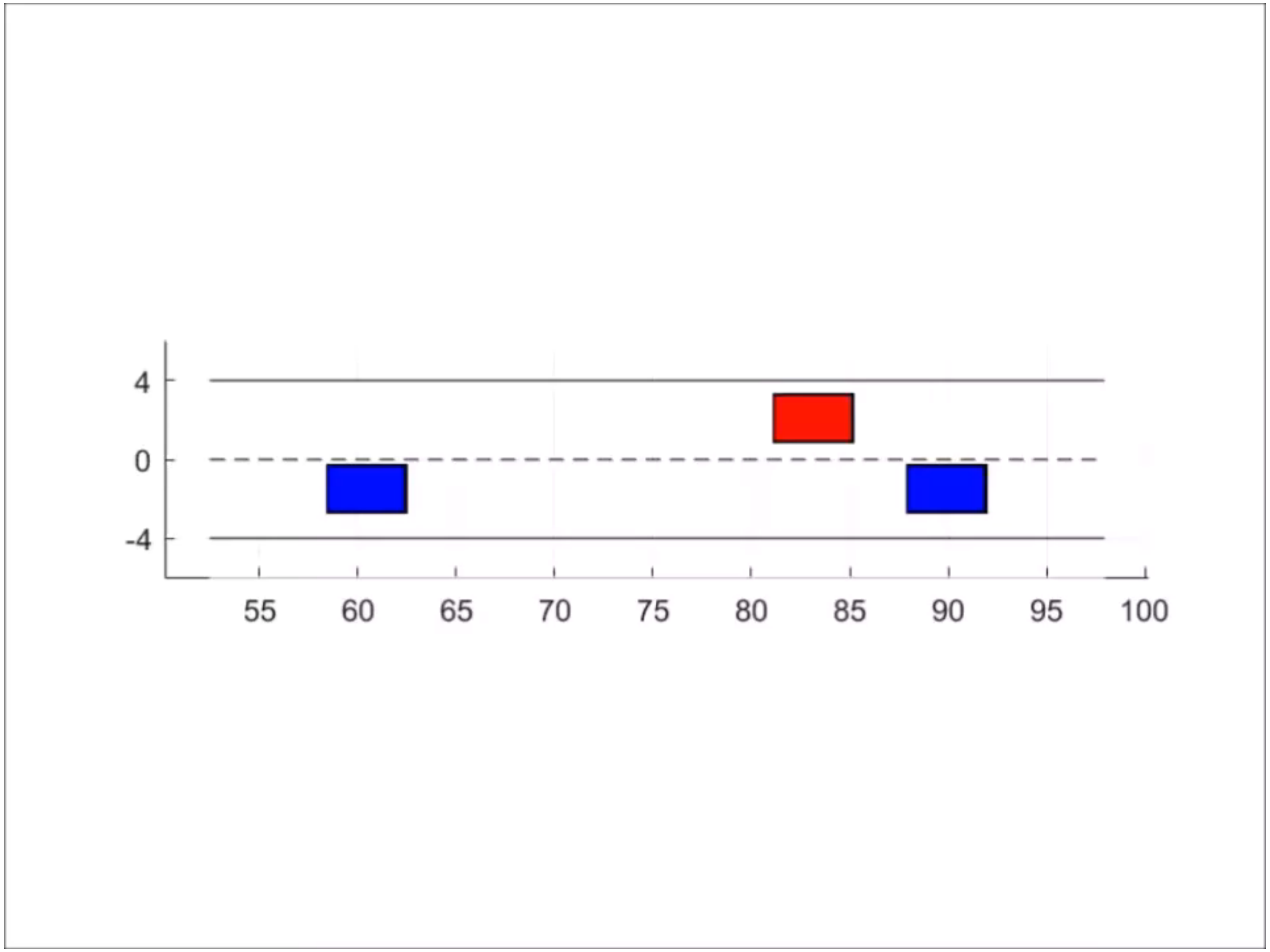
Case Study - Updated

Updated Sensor Setup and Controller:

- Better side and corner coverage
- Better detection of vehicle on the side
- Speed up / slow down based on the vehicle on the side



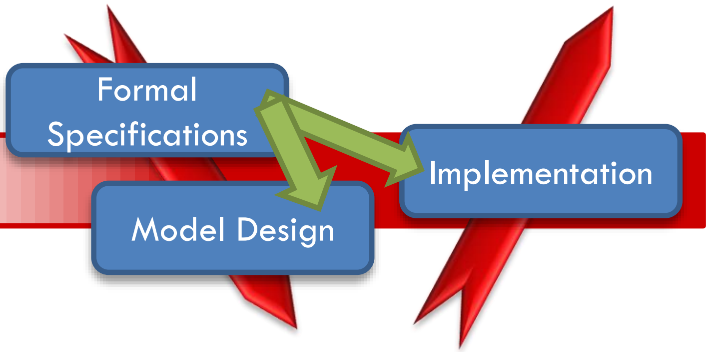
Case Study - Updated



Overview


Joint work with
Bardh Hoxha (ASU)
Adel Dokhanchi (ASU)

- Motivation
  - Quick intro to control synthesis challenges
  - Model Based Development
- Formal requirements for CPS
- Requirements driven falsification
- Autonomous vehicle testing
- **Parameter mining in requirements**
- Conformance testing
- Testing based verification
- Vision, Other topics & Future work



Hoxha, Dokhanchi, Fainekos, *Mining Parametric Temporal Logic Properties in Model Based Design for Cyber-Physical Systems*, To Appear in STTT

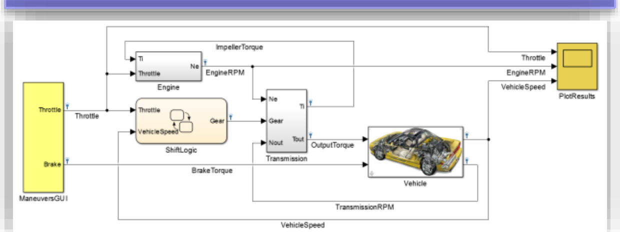
Parameter Mining

What is the <u>shortest time</u> that the engine speed can exceed 3200RPM?

The vehicle speed is always less than <u>parameter θ_1</u> and the engine speed is always <u>less than θ_2</u>.

System Σ

$y = \Delta(x_0, u)$ $\quad x_0 \in X_0$ $\quad u \in \mathbb{U}$



If I <u>increase/decrease θ_1</u> by a specific amount, how much do I have to <u>increase/decrease θ_2</u> so that the system satisfies the specification?"

Parameter Mining

The vehicle speed is always less than parameter θ_1 and the engine speed is always less than θ_2.

Parametric MTL: $\phi_1\left[\vec{\theta}\right] = Always((v \leq \theta_1) \wedge (\omega \leq \theta_2))$

PMTL formulas may contain state and/or timing parameters

Ex. $\phi_2\left[\vec{\theta}\right] = \neg(Eventually_{[0,\theta_1]}(v > 100) \wedge (\omega \leq \theta_2))$

Timing

State

Parameter Mining

Parameter Mining Problem:

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

Approximation possible ☺

Question:

Why don't we search for the set of parameters for which the system satisfies the specification?

Problem is undecidable [AL94] ☹.

[AL94]: Alur, Rajeev, et al. "The algorithmic analysis of hybrid systems." *11th International Conference on Analysis and Optimization of Systems Discrete Event Systems*. Springer Berlin Heidelberg, 1994.

Parameter Mining

Testing framework based on the theory of robustness of MTL formulas

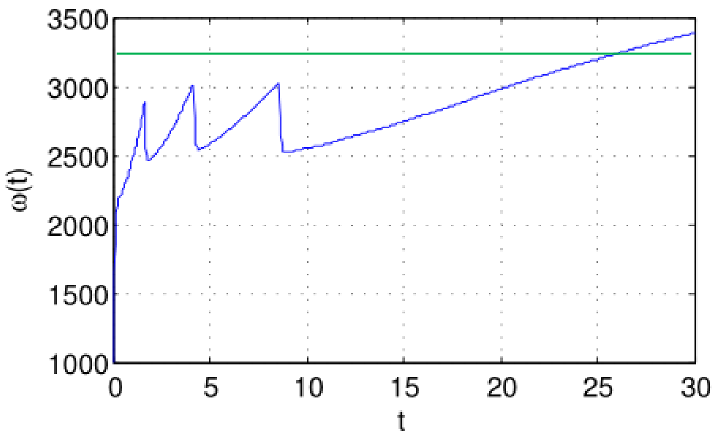
Monotonicity properties of parametric MTL formulas.

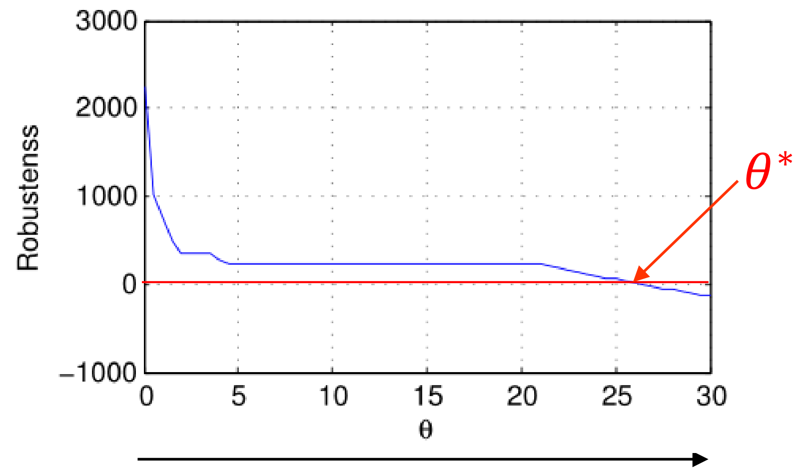
Parameter mining $\Rightarrow$ *Multi-parametric optimization problem*

Monotonicity of parametric MTL specifications

NL: Always, from 0 to θ, the engine speed is less than 3250

$$\phi[\theta] = Always_{[0,\theta]}(\omega \leq 3250)$$





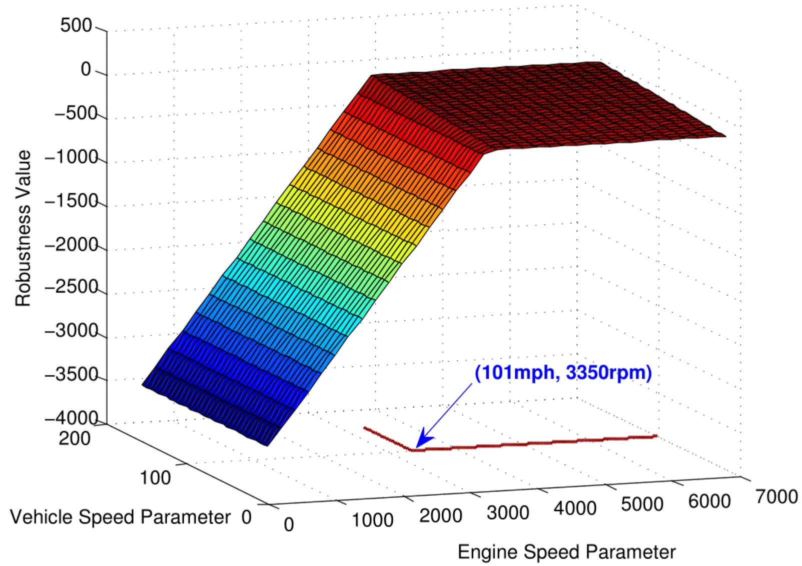
As we increase θ, we can only increase the opportunity to find falsifying system behavior

Non-Increasing robustness with respect to θ

Monotonicity of parametric MTL specifications

NL: Always, vehicle speed is less than θ_1 and engine speed is less than θ_2

$$\phi_1[\theta] = Always((v \leq \theta_1) \wedge (\omega \leq \theta_2))$$



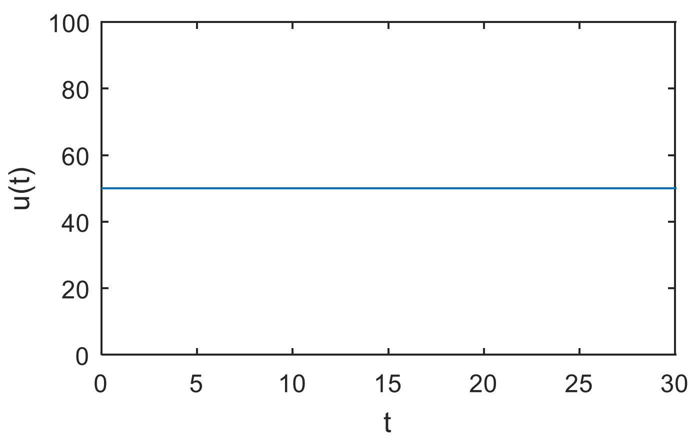
As we increase θ_1 and θ_2, we can only decrease the opportunity to find a falsifying system behavior

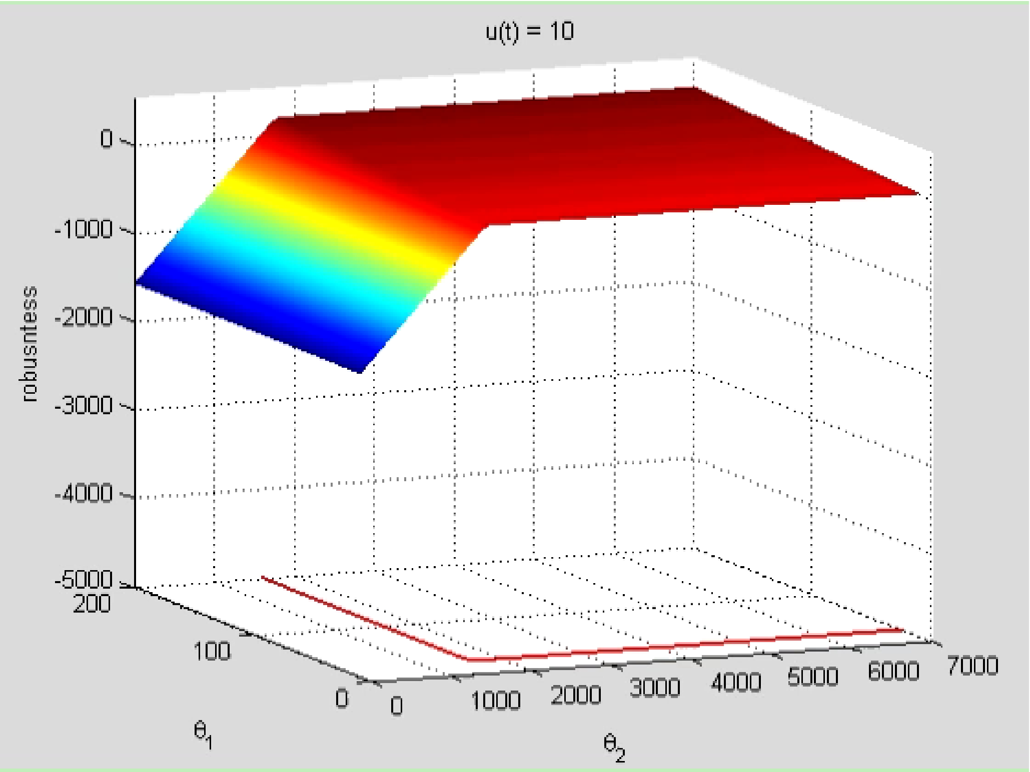
Non-Decreasing robustness with respect to $f(\vec{\theta})$

Monotonicity of parametric MTL specifications

$$\phi_1[\theta] = Always((v \leq \theta_1) \wedge (\omega \leq \theta_2))$$

Example: Searching over constant input signals to the system





Minimizing Temporal Logic Robustness

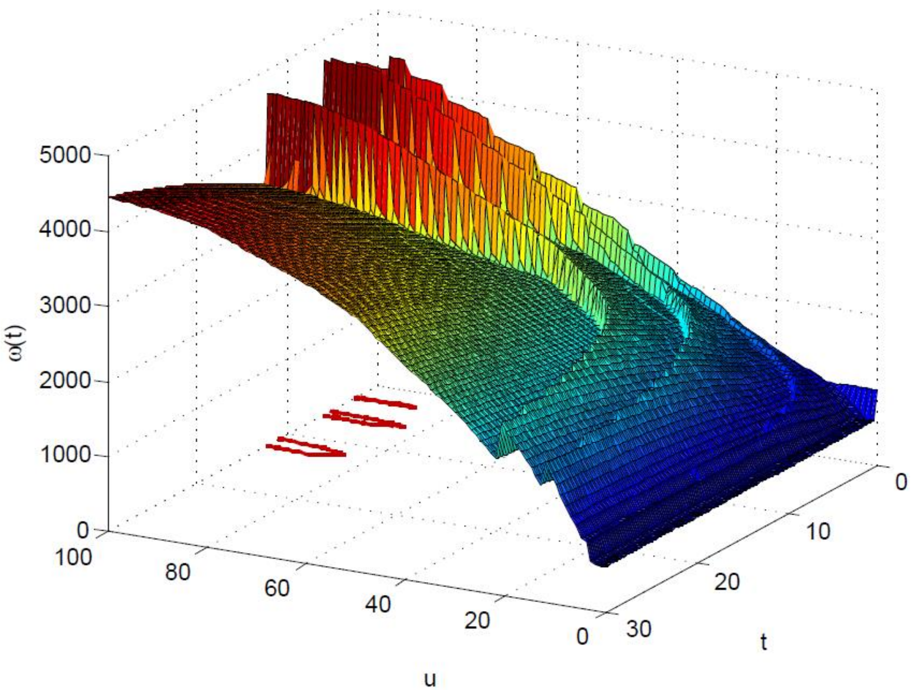
- We need to solve an optimization problem:

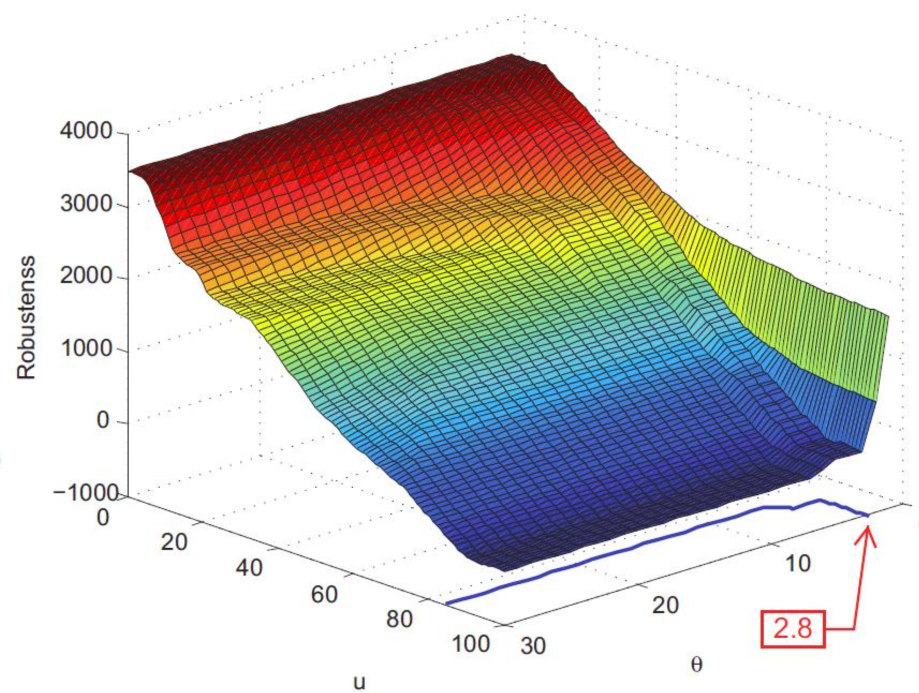
$$\begin{array}{ll} \text{optimize} & \theta \\ \text{subject to} & \theta \in \Theta \text{ and } [\![\phi[\theta]]\!](\Sigma) = \min_{\mu \in \mathcal{L}_\tau(\Sigma)} [\![\phi[\theta]]\!](\mu) \leq 0 \end{array}$$

- **Challenges:**
  - Non-linear system dynamics
  - Unknown input signals
  - Unknown system parameters
  - Non-differentiable cost function
    - not known in closed form
    - needs to computed
  - **When multiple parameters: Pareto front**

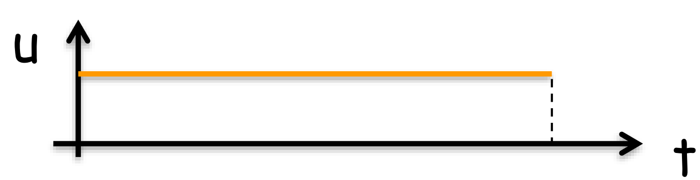
How does our cost function look like?

Spec: $G_{[0,\theta]}(\omega<4500RPM))$

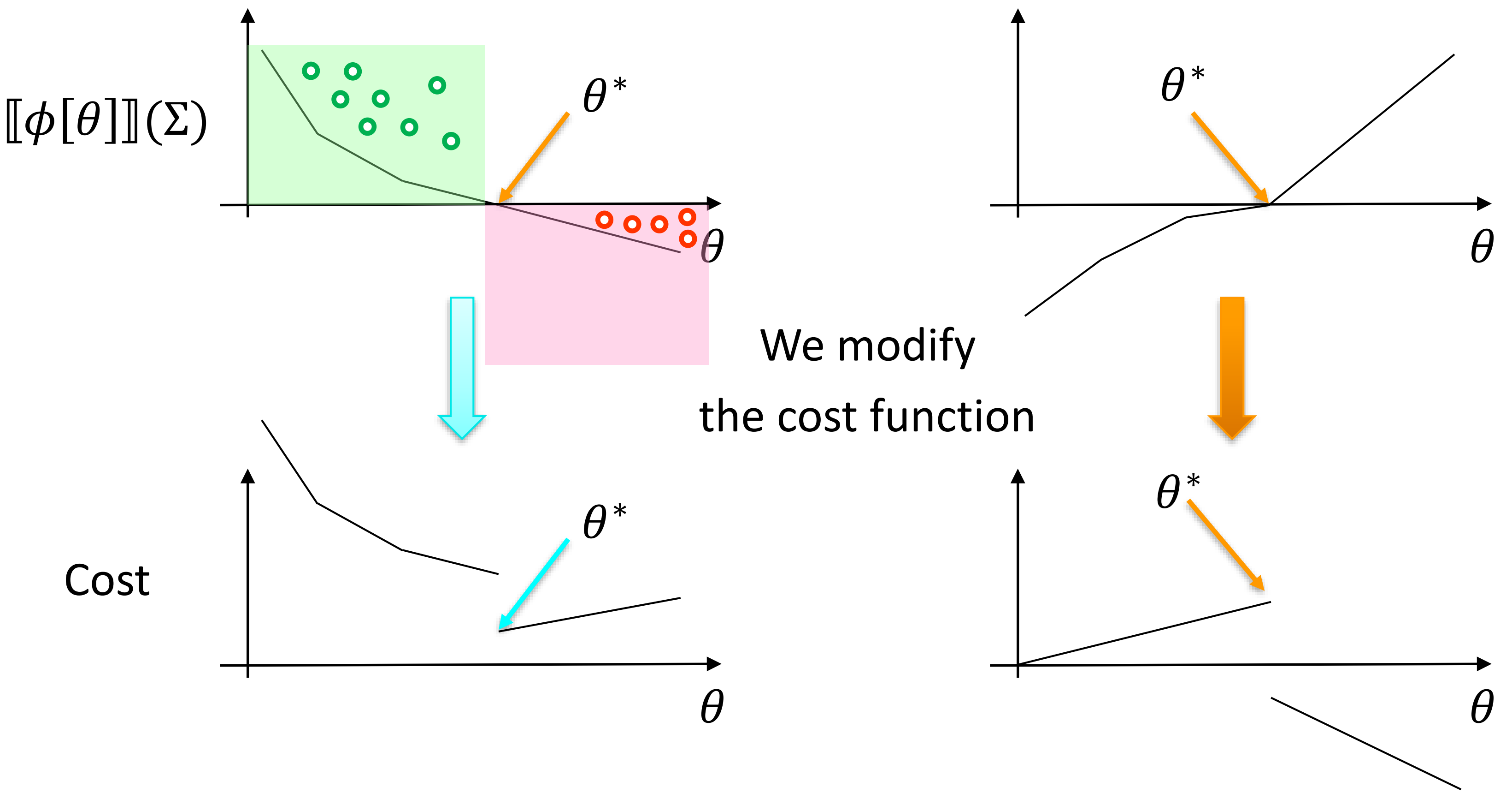




Throttle % parameterization with 1 variable



Parameter Bound Computation



Non-Increasing robustness with respect to θ

Minimize

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

Non-Decreasing robustness with respect to θ

Maximize

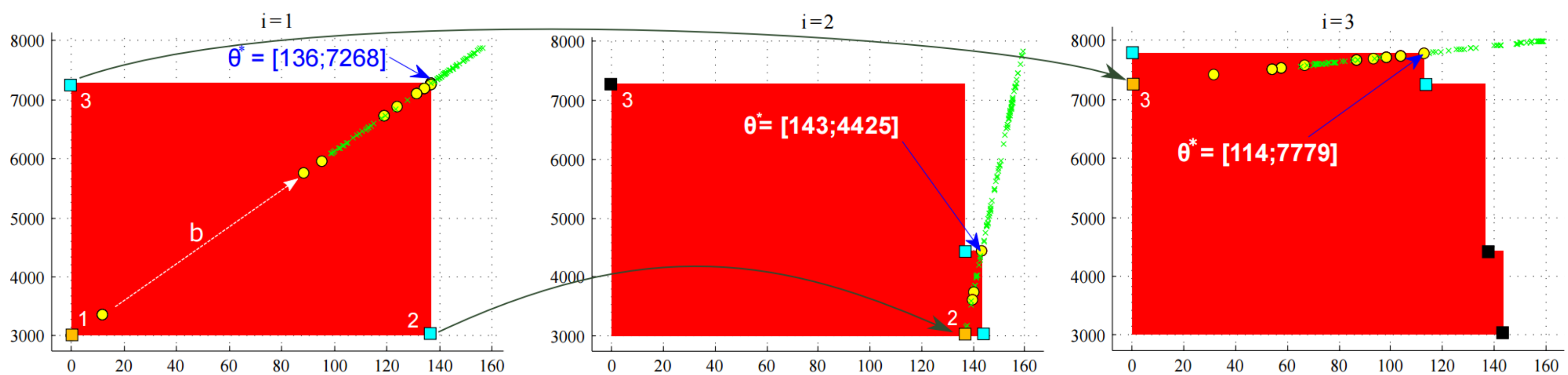
$$\max_{\theta \in \Theta} \max_{\mu \in \mathcal{L}_\tau(\Sigma)} \left(f(\theta) + \begin{cases} \gamma - [\![\phi[\theta]]\!](\mu) & \text{if } [\![\phi[\theta]]\!](\mu) \geq 0 \\ 0 & \text{otherwise} \end{cases} \right)$$

Parameter Falsification Domain

Alg: Structured Parameter Falsification Domain Algorithm

$$\phi[\theta] = Always((v \leq \theta_1) \wedge (\omega \leq \theta_2))$$

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



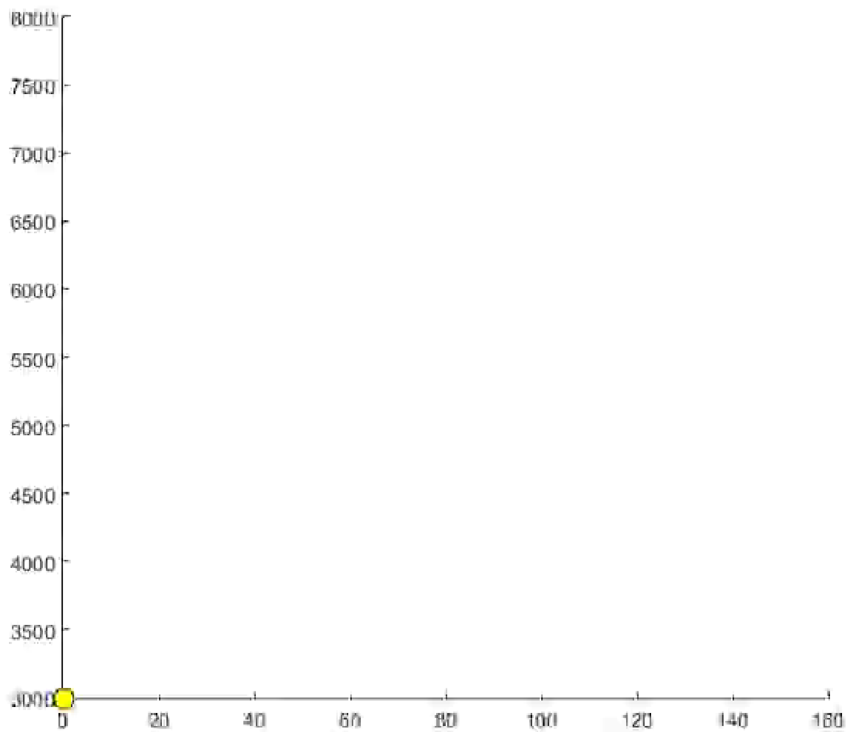
$$\max_{\theta \in \Theta} \max_{\mu \in \mathcal{L}_\tau(\Sigma)} \left(f(\theta) + \begin{cases} \gamma - [\![\phi[\theta]]\!](\mu) & \text{if } [\![\phi[\theta]]\!](\mu) \geq 0 \\ 0 & \text{otherwise} \end{cases} \right)$$

Parameter Falsification Domain

Alg 2: Structured Parameter Falsification Domain Algorithm

$$\phi[\theta] = Always((v \leq \theta_1) \wedge (\omega \leq \theta_2))$$

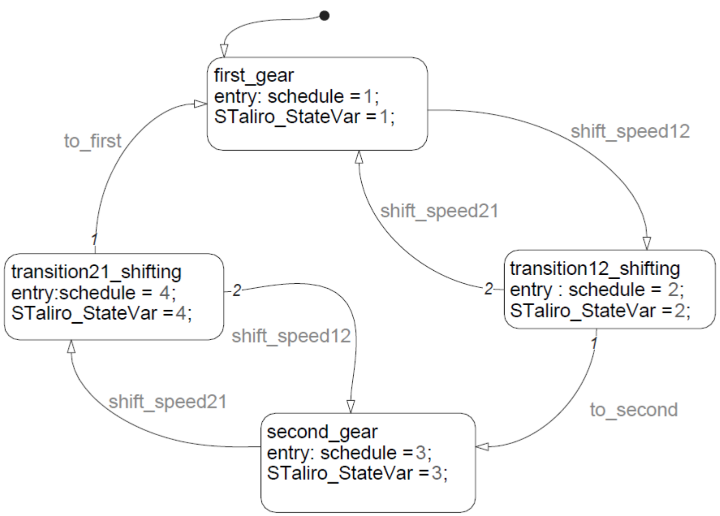
Non-Decreasing robustness with respect to $f(\vec{\theta})$

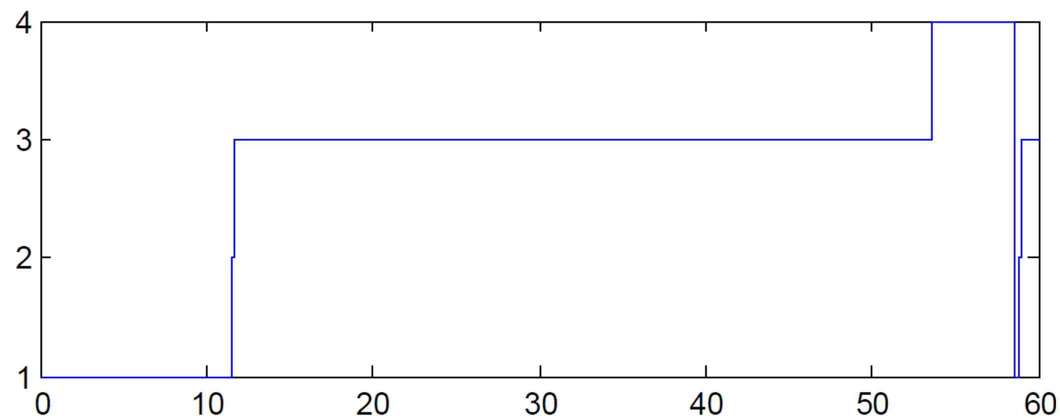


Powertrain Example: Parameter querying

$$\varphi_2 = G((\neg gear_1 \wedge X\ gear_1) \rightarrow G_{[0,2.5]}\ \neg gear_2)$$

$$\varphi_2 = G((\neg gear_1 \wedge X\ gear_1) \rightarrow G_{[0,?]}\ \neg gear_2)$$

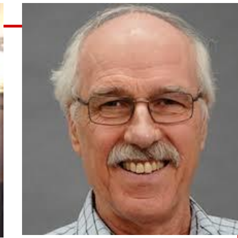




? = 0.4273

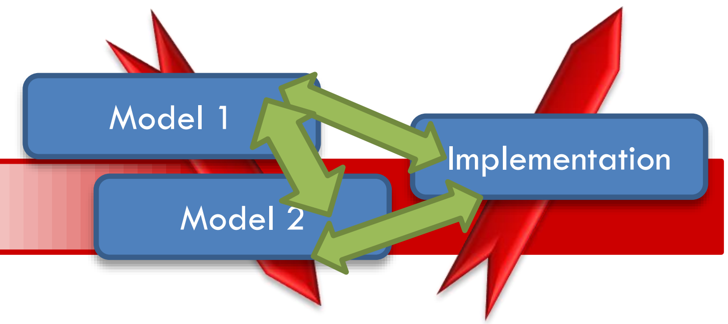
I.e. for any parameter ≥ 0.4273, it is guaranteed that the system does not satisfy φ_2.

Overview



Joint work with
Houssam Abbas (UPenn)
Hans Mittelmann (ASU)

- Motivation
  - Quick intro to control synthesis challenges
  - Model Based Development
- Formal requirements for CPS
- Requirements driven falsification
- Autonomous vehicle testing
- Parameter mining in requirements
- **Conformance testing**
- Testing based verification
- Vision, Other topics & Future work

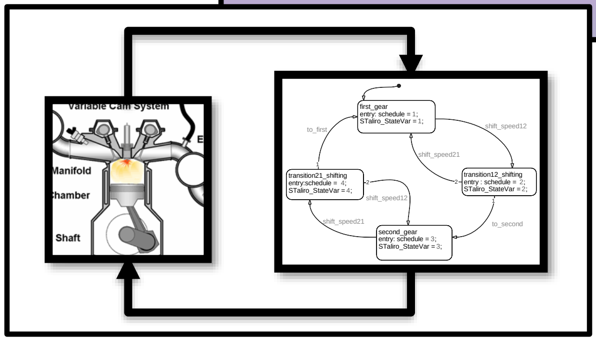


Abbas, Mittelmann and Fainekos, *Formal property verification in a conformance testing framework*, MEMOCODE 2014

Conformance Problem

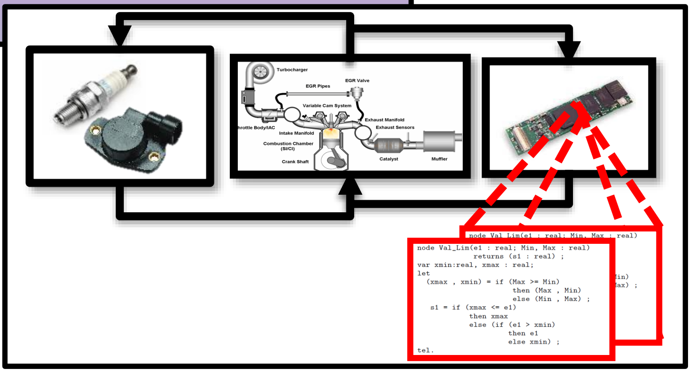
Model 1

$\sigma_1 = \Delta_1(x_0, u)$ $\quad x_0 \in X_0$, $u \in \mathbb{U}$



Model 2 / Implementation

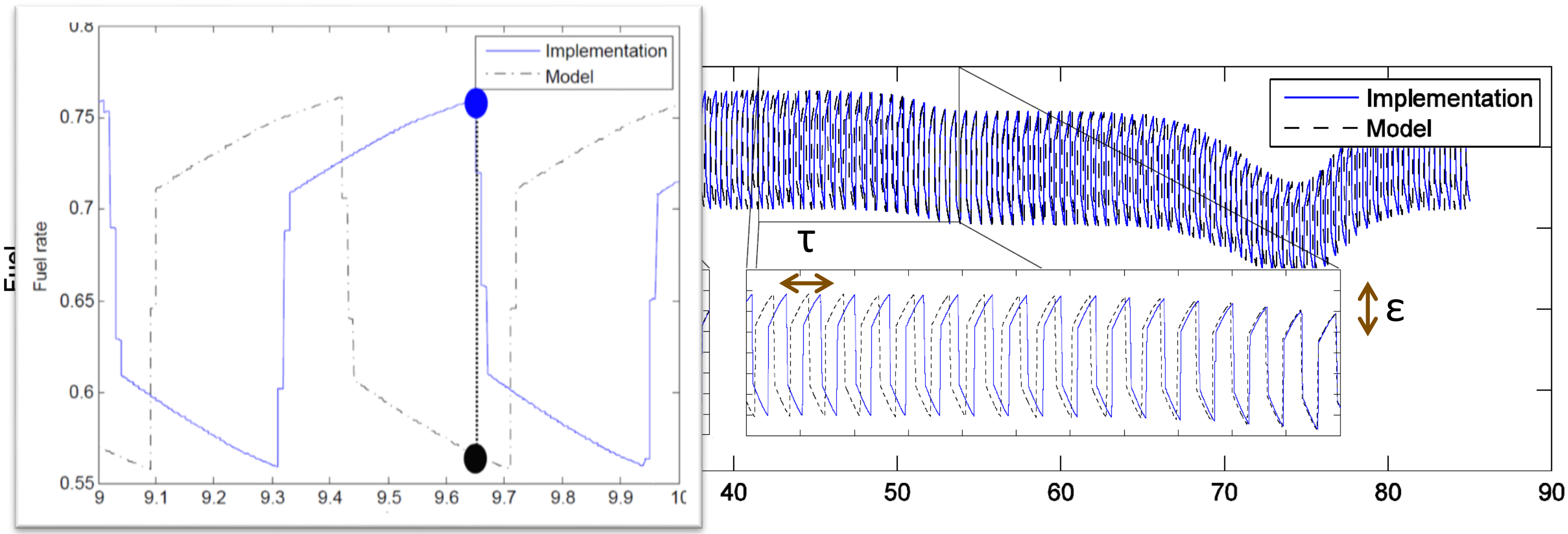
$\sigma_2 = \Delta_2(x_0, u)$ $\quad x_0 \in X_0$, $u \in \mathbb{U}$



Does the implementation conform to the model?

- System 1 is deterministic (or maybe stochastic) model. Not an abstraction!
- Thus, we need to talk about "distance" between the system behaviors.
- What is an appropriate notion of distance?

Conformance Notion for CPS?



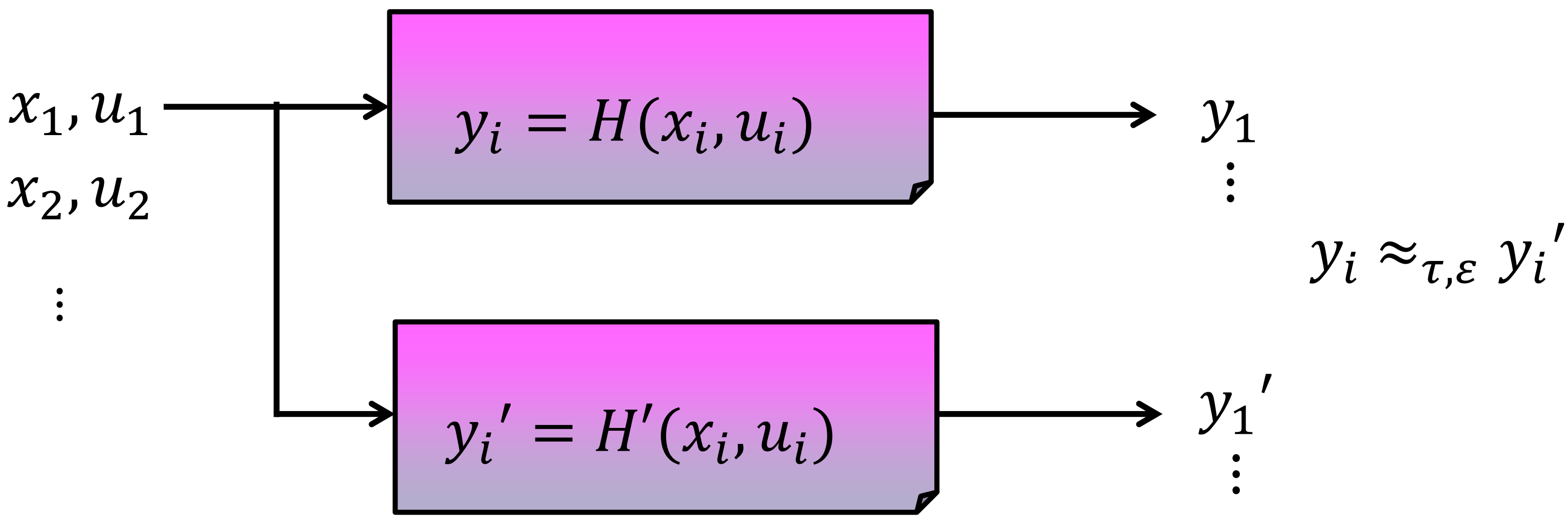
Consider two trajectories $\boldsymbol{y}$, and $\boldsymbol{y}'$ of Σ and Σ', respectively. Given $T > 0$, $J > 0, \tau > 0$, and $\varepsilon > 0$, we say $\boldsymbol{y}$ and $\boldsymbol{y}'$ are $(T, J, \tau, \varepsilon)$–close if:

a) For all (t, j) in the support of $\boldsymbol{y}$ s.t. $t \leq T$ and $j \leq J$, there exists (s, j) in the support of $\boldsymbol{y}'$, such that $|t - s| < \tau$ and $\left\| y(t, j) - y'(s, j) \right\| < \varepsilon$

b) For all (t, j) in the support of $\boldsymbol{y}'$, s.t. $t \leq T$ and $j \leq J$, there exists (s, j) in the support of $\boldsymbol{y}$, such that $|t - s| < \tau$ and $\left\| y'(t, j) - y(s, j) \right\| < \varepsilon$

Conformance Between Systems

A system is a map $H\colon X_0 \times U \rightarrow (E \rightarrow \mathbb{R}^n)$, E = time domain



Write this as $H \preccurlyeq_{\tau,\varepsilon} H'$

The smallest ε s.t. $H \preccurlyeq_{\tau,\varepsilon} H'$ is the **conformance degree given** τ.

Property Preservation?

Model 1 (M_1)

$\sigma_1 = \Delta_1(x_0, u)$ $\quad x_0 \in X_0$, $u \in \mathbb{U}$

Model 2 (M_2)

$\sigma_2 = \Delta_2(x_0, u)$ $\quad x_0 \in X_0$, $u \in \mathbb{U}$

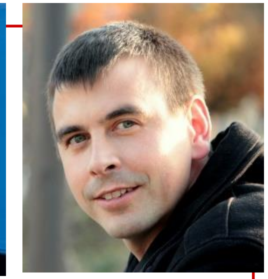
$M_1 \models \varphi$

$M_2 \models \varphi'$??

Theorem: Let H_1 and H_2 be two hybrid systems, and φ be an MTL formula. If $H_1 \preceq_{(\tau,\varepsilon)} H_2$ and $H_2 \models_O \varphi$, then $H_1^\tau \models_{O\varepsilon} \varphi_\tau$.

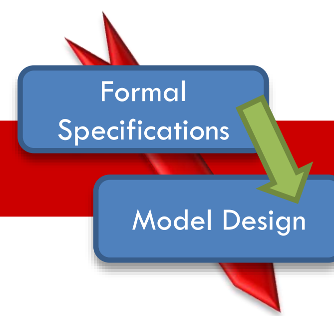
Abbas, Mittelmann and Fainekos, *Formal property verification in a conformance testing framework*, MEMOCODE 2014

Overview

Joint work with
George Pappas (UPenn)
Antoine Girard (CNRS)

- Motivation
  - Quick intro to control synthesis challenges
  - Model Based Development
- Formal requirements for CPS
- Requirements driven falsification
- Autonomous vehicle testing
- Parameter mining in requirements
- Conformance testing
- **Testing based verification**
- Vision, Other topics & Future work



Temporal Logic Testing Based Verification

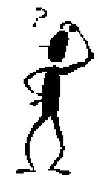
Closed-loop system Σ:

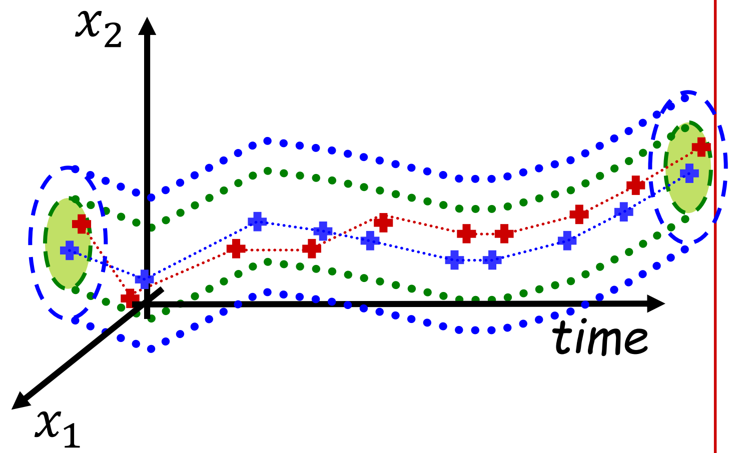
$$\dot{x} = f(x)$$
$$y = g(x)$$

$X_0 \subseteq X$

Specification Φ

$$\mathcal{L}(\Sigma) \subseteq \mathcal{L}(\Phi)$$





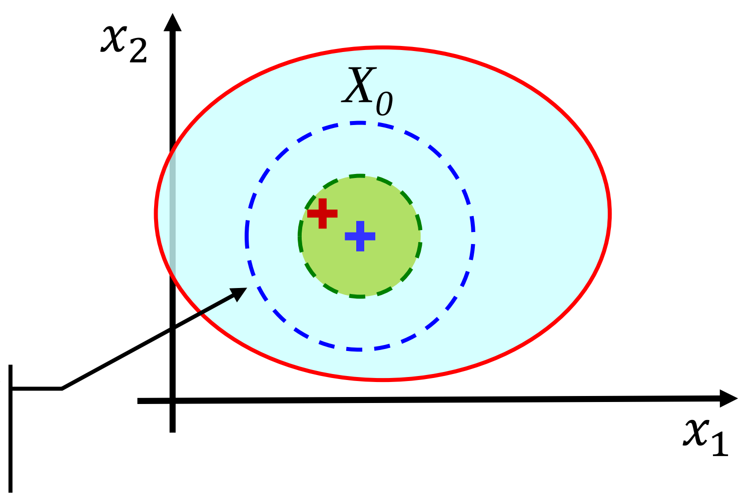
Green tube of system trajectories

⊆

Blue tube of trajectories that satisfy the specification Φ

ε robustness parameter

$B_\rho(\sigma, |\varepsilon|)$



Property: Any trajectory inside the blue tube satisfies the same specification as the blue trajectory.

Property: If a trajectory starts inside the green ball in the initial conditions, then it stays in the green tube for all time.

Temporal Logic Testing Based Verification

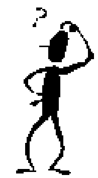
Closed-loop system Σ:

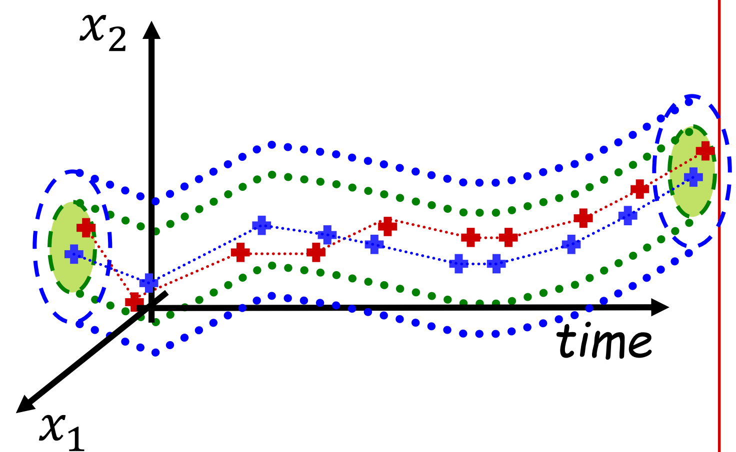
$$\dot{x} = f(x)$$
$$y = g(x)$$

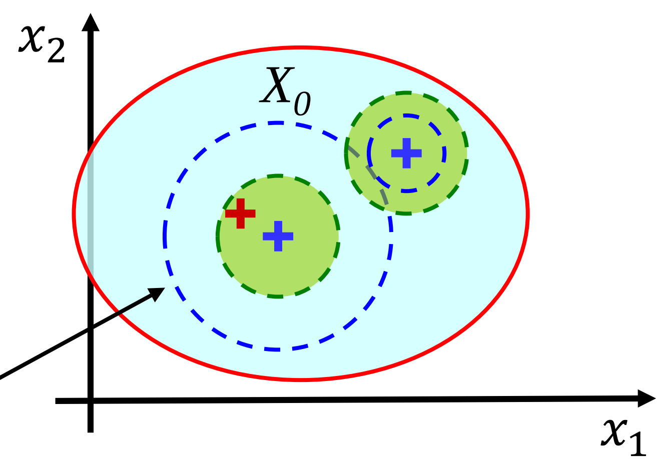
$X_0 \subseteq X$

Specification Φ

$$\mathcal{L}(\Sigma) \subseteq \mathcal{L}(\Phi)$$







ε robustness parameter

$B_\rho(\sigma, |\varepsilon|)$

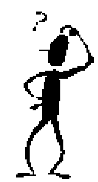
Temporal Logic Testing Based Verification

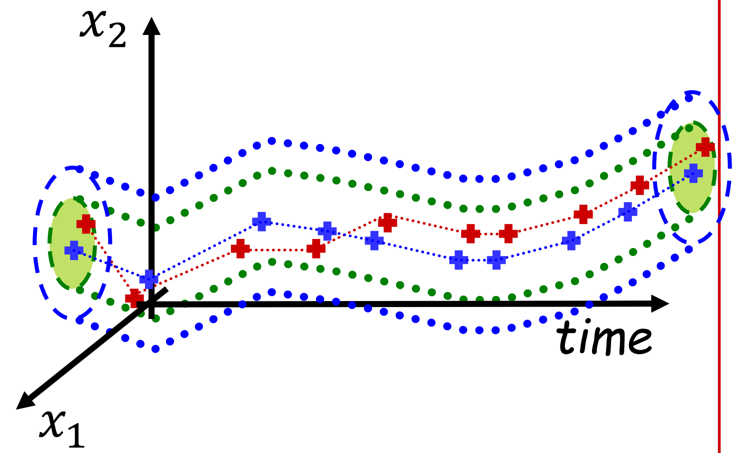
Closed-loop system Σ:

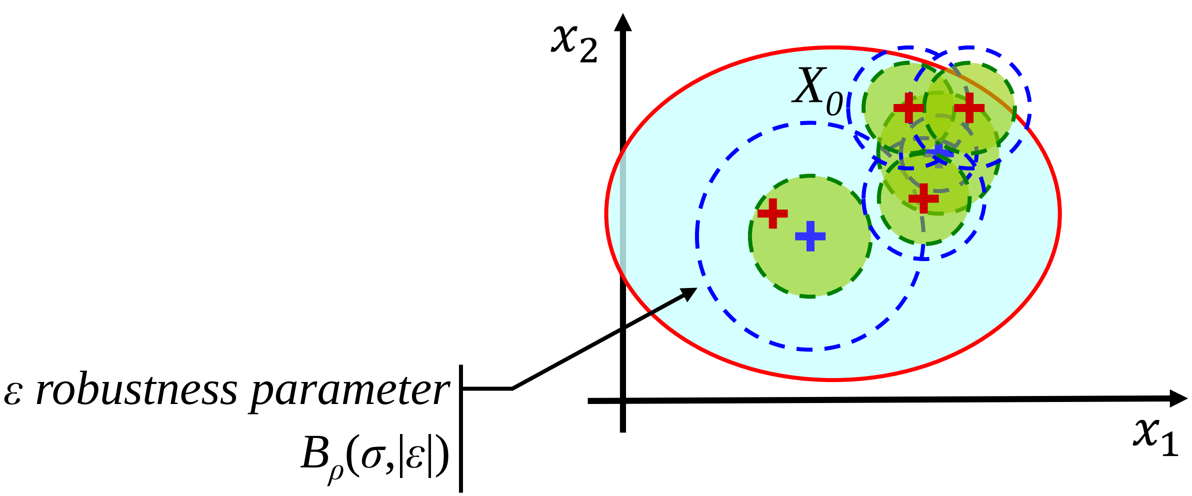
$$\dot{x} = f(x) \qquad X_0 \subseteq X$$
$$y = g(x)$$

Specification Φ

$$\mathcal{L}(\Sigma) \subseteq \mathcal{L}(\Phi)$$







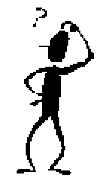
Achieving coverage!

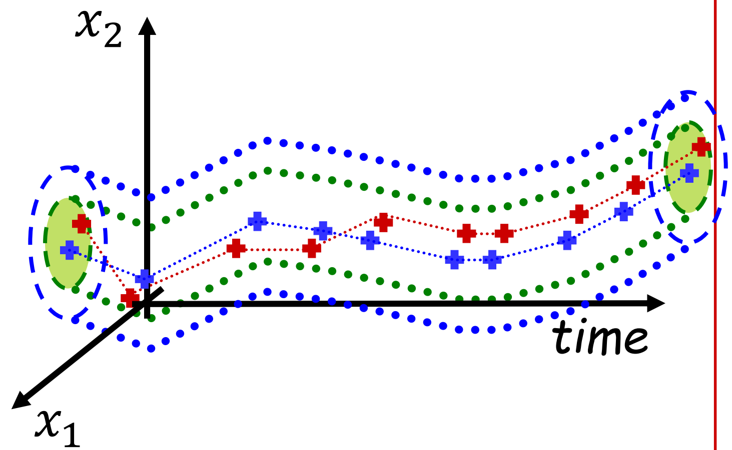
Closed-loop system Σ:

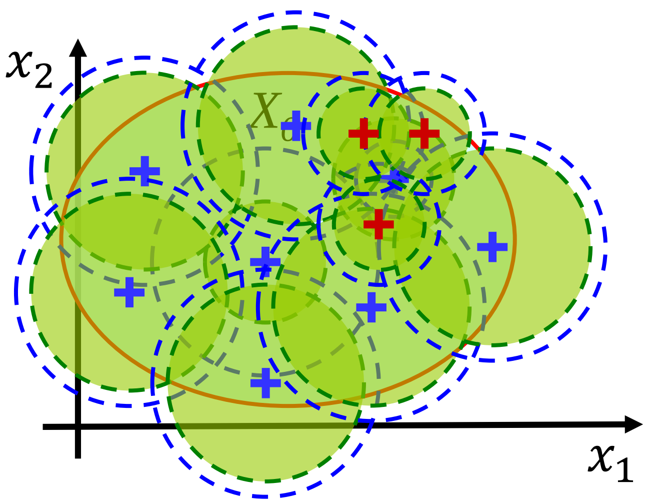
$$\dot{x} = f(x) \qquad X_0 \subseteq X$$
$$y = g(x)$$

Specification Φ

$$\mathcal{L}(\Sigma) \subseteq \mathcal{L}(\Phi)$$







Good news!

Coverage with a finite number of simulations

Computing bisimulation functions

Quadratic Bisimulation Functions for Deterministic Linear Systems

$$\dot{x} = f(x)$$
$$y = g(x)$$

y

Proceedings of the
44th IEEE Conference on Decision and Control, and
the European Control Conference 2005
Seville, Spain, December 12-15, 2005

WeA16.4

Approximate Bisimulations for Constrained Linear Systems

Antoine Girard and George J. Pappas

$$V(x) = \sqrt{x^T M x}$$

is a bisimulation function if

$$M \geq C^T C$$

$$A^T M + MA \leq 0$$

Bisimulation Functions using Sum Of Squares Relaxation

$$\dot{x} = f(x)$$
$$y = g(x)$$

y

Proceedings of the
44th IEEE Conference on Decision and Control, and
the European Control Conference 2005
Seville, Spain, December 12-15, 2005

MoB01.3

Approximate Bisimulations for Nonlinear Dynamical Systems

Antoine Girard and George J. Pappas

$$V(x_1, x_2) = \sqrt{q(x_1, x_2)}$$

is a bisimulation function if

$$q(x_1, x_2) - \|g_1(x_1) - g_2(x_2)\|^2 \quad \text{is SOS}$$

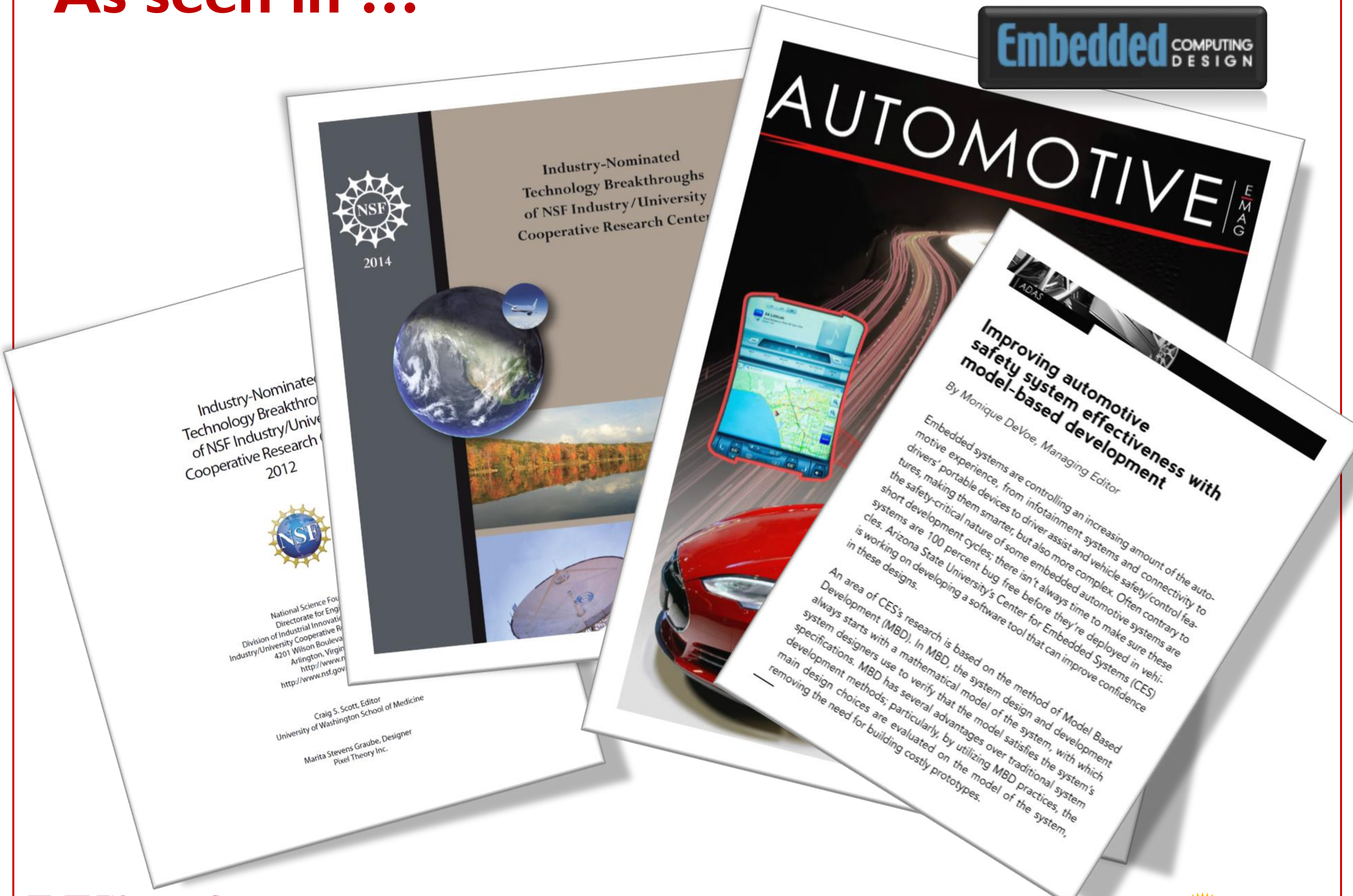
$$-\frac{\partial q(x_1, x_2)}{\partial x_1} f_1(x_1) - \frac{\partial q(x_1, x_2)}{\partial x_2} f_2(x_2) \quad \text{is SOS}$$

[For more details and possibilities see Tabuada 2009]

Overview

- Motivation
  - Quick intro to control synthesis challenges
  - Model Based Development
- Formal requirements for CPS
- Requirements driven falsification
- Autonomous vehicle testing
- Parameter mining in requirements
- Conformance testing
- Testing based verification
- Vision, Other topics & Future work

As seen in …



Industry-Nominated
Technology Breakthroughs
of NSF Industry/University
2012
NSF
2014
Industry-Nominated
Technology Breakthroughs
of NSF Industry/University
Craig S. Scott, Editor
University of Washington School of Medicine
Marita Stevens Graube, Designer
Pixel Theory Inc.
Embedded COMPUTING DESIGN
AUTOMOTIVE
EMAG
ADAS
Improving automotive
safety system effectiveness with
model-based development
By Monique DeVoe, Managing Editor
Embedded systems are controlling an increasing amount of the automotive experience, from infotainment systems and connectivity to drivers' portable devices to driver assist and vehicle safety/control features, making them smarter, but also more complex. Often contrary to the safety-critical nature of some embedded automotive systems are short development cycles; there isn't always time to make sure these systems are 100 percent bug free before they're deployed in vehicles. Arizona State University's Center for Embedded Systems (CES) is working on developing a software tool that can improve confidence in these designs.
An area of CES's research is based on the method of Model Based Development (MBD). In MBD, the system design and development always starts with a mathematical model of the system, with which system designers use to verify that the model satisfies the system's specifications. MBD has several advantages over traditional system development methods; particularly, by utilizing MBD practices, the main design choices are evaluated on the model of the system, removing the need for building costly prototypes.

Vision: *a complete theory for MBD for CPS*

Transparent from the user perspective:

1. Automated synthesis
2. Testing and verification support with guarantees

Informal Requirements

Formal Specifications

Model Design

Autocode Generation (with multi-core in mind)

Processor In the Loop (PIL)

Hardware In the Loop (HIL)

System Calibration

System Deployment

Awards:
1017074, 1116136,
1319560, 1350420, 1446730

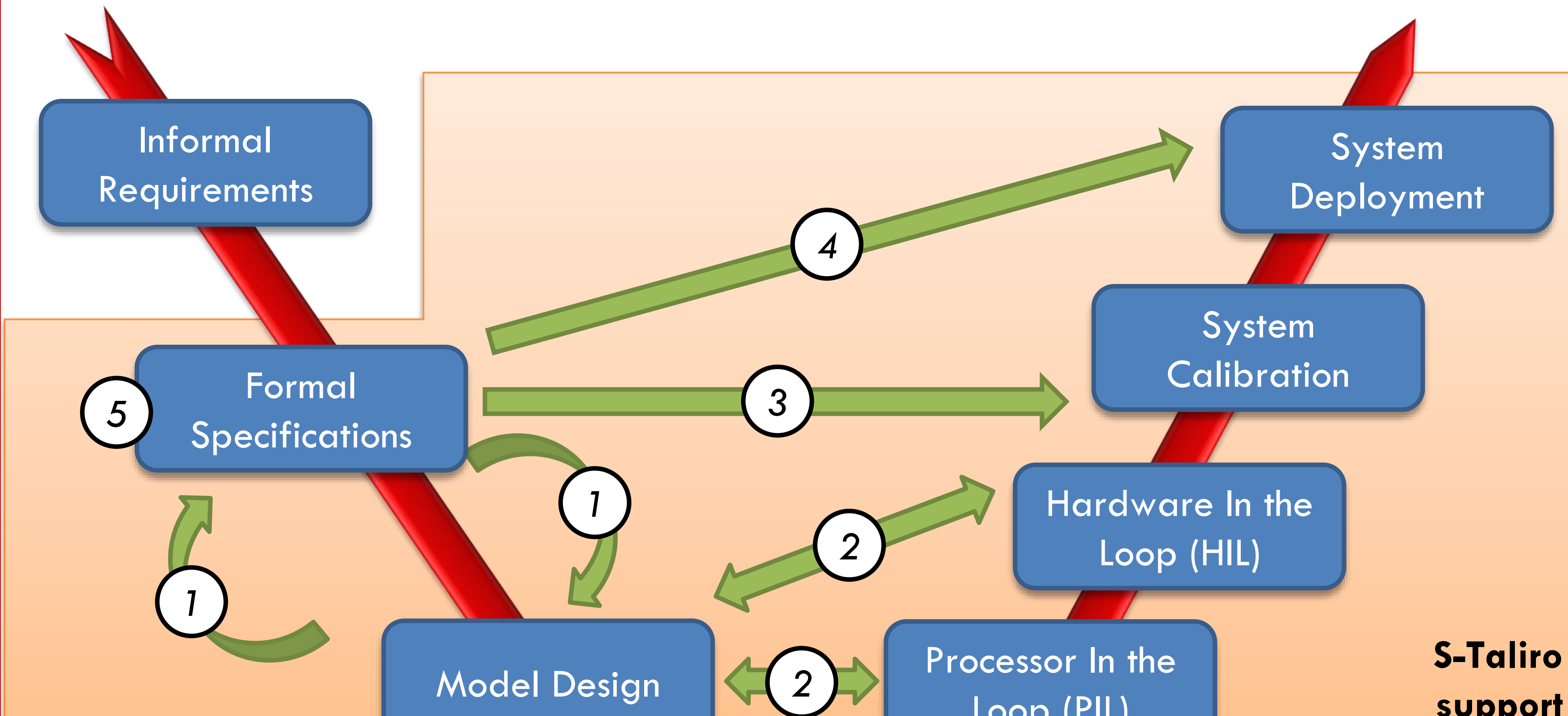
Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

TOYOTA

BOSCH

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

S-Taliro support in the V-process



1. Testing formal specifications and specification mining [TECS 2013, ICTSS 2012, …]
2. Conformance testing: models, HIL/PIL or tuned/calibrated model [MEMOCODE 2014]
3. Testing formal specifications on the HIL/PIL calibrated system [TECS 2013, …]
4. Runtime monitoring of formal requirements [RV 2014]
5. Specification visualization [IROS 2015] & Debugging [MEMOCODE 2015]

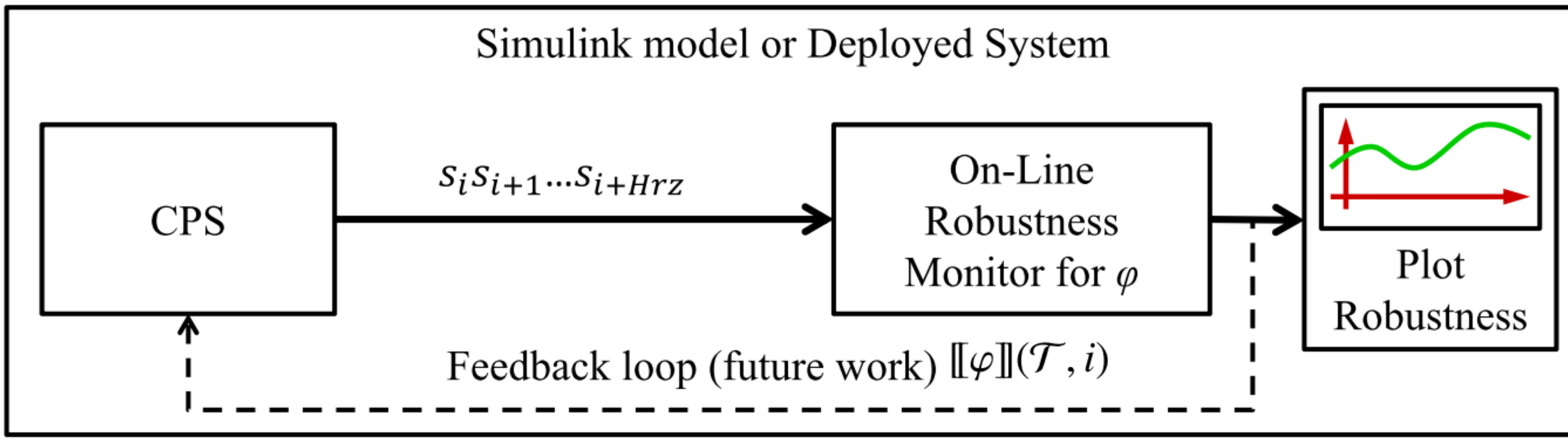
Current S-Taliro Functionality

Falsification	Utilizes stochastic optimization algorithms with the theory of the robustness over MTL specifications to find system behaviors that falsify the specification.
Parameter Mining	Given a parametric MTL specification, with unknown state and/or timing parameters, find the parameter range for which the system falsifies the specification.
Runtime Verification	Enables on-line monitoring of MTL specifications through a Simulink block that can run as an integrated module in the simulation process.
Conformance Testing	Test the conformance between a model and implementation.
Worst Expected Robustness for Stochastic Systems	The method searches for a global minimizer for the expected temporal logic robustness of SCPS.
Elicitation of Formal Requirements	Enables the elicitation of formal requirements through the tool ViSpec.
Debugging of Formal Requirements	Enables the debugging of formal requirements.

On-Line Monitoring problem

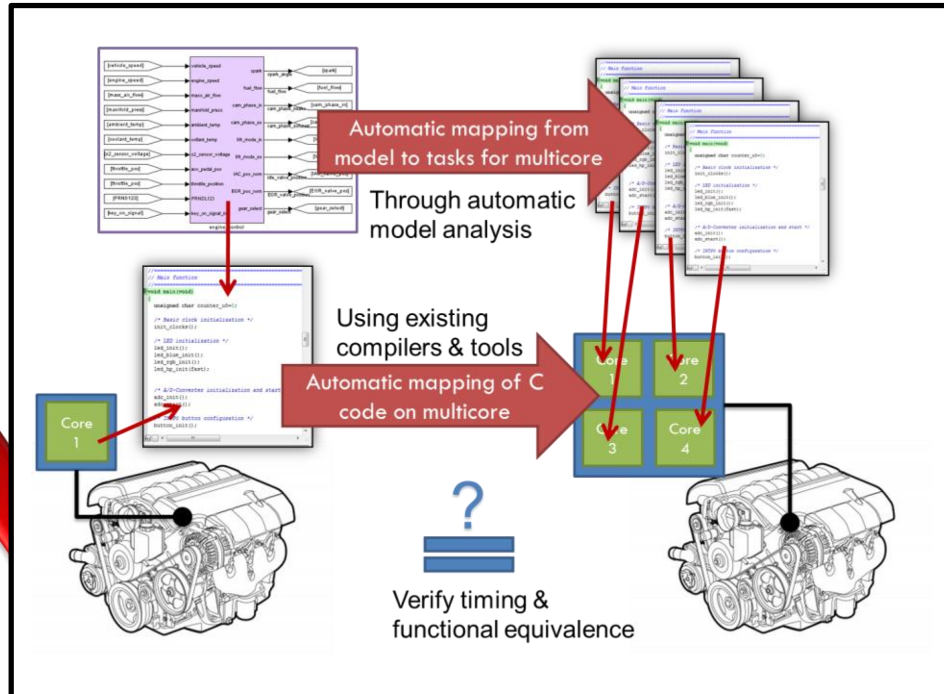
Award: 1319560

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.



[Dokhanchi, Hoxha , Fainekos, RV 14]

Automating Model to HIL on Multicore



Model Design

Hardware In the Loop (HIL)

Autocode Generation
(with multi-core in mind)

TOYOTA

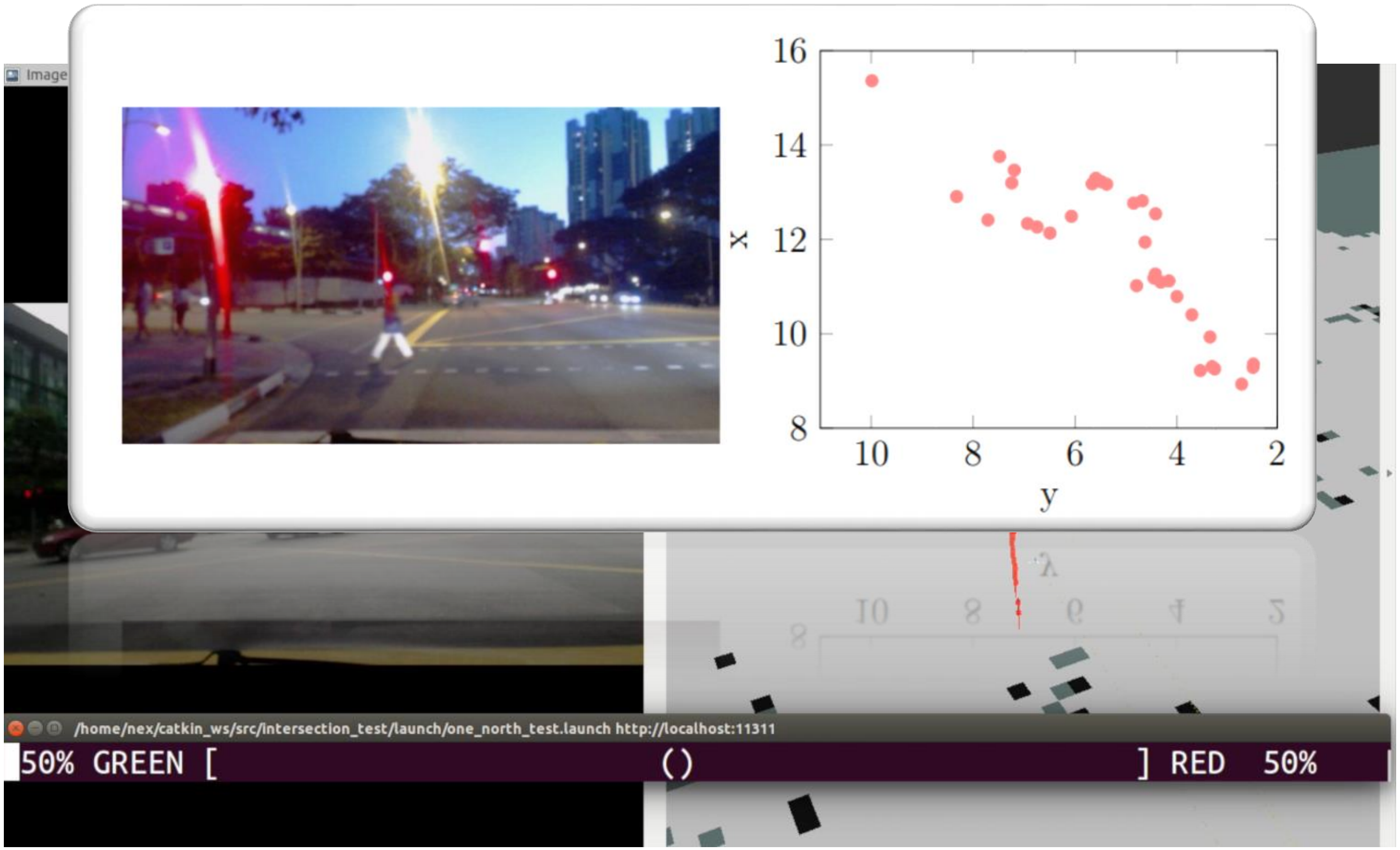
CENTER FOR EMBEDDED SYSTEMS
an NSF Industry/University Cooperative Research Center

Tuncali, Fainekos, Lee, *Automatic Parallelization of Multi-rate Block Diagrams of Control Systems on Multi-core Platforms*, ACM TECS, 2016, V16, Article No 15

Where are we going with this?

Testing sensing and perception algorithms as part of the system.

Typical Example: *Traffic Light Status Detection Using Movement Patterns of Vehicles**



* Campbell et al *Traffic Light Status Detection Using Movement Patterns of Vehicles*, IEEE Intelligent Transportation Systems Conference, 2016

Acknowledgements

(Main contributors to the S-TaLiRo project)

Current Students

- Adel Dokhanchi – PhD
- Bardh Hoxha – PhD
- C. Erkan Tuncali – PhD
- Shakiba Yaghoubi – PhD

Former Students

- Houssam Abbas - PhD
- Y. Annapureddy - MS
- Rahul T. Srinivasa - MS
- Hengyi Yang – MS
- Hoang Bach – BS
- Jorge Mendoza – BS

Main collaborator

- CU, Boulder: S. Sankaranarayanan

Other collaborators

- ASU: Y. Kobayashi, Y-H Lee, H. Mittelmann
- NEC Labs: A. Gupta (now in Princeton), F. Ivancic (now in Google)
- RPI: Agung Julius
- Toyota: J. V. Deshmukh, J. Kapinski, K. Ueda, H. Yazarel (now in CareFusion), X. Jin

We build systems you can trust your life on!

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.

NSF

BOSCH

MBD
TOYOTA TECHNICAL CENTER

Special Thanks: S. Vrudhula (ASU)

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