S-TaLiRo 1.5 Demonstration & Quick Setup Guide

Georgios Fainekos

School of Computing, Informatics and
Decision System Engineering

Arizona State University

fainekos at asu edu

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

This is an extended version of the slides presented in the Workshop on formal methods for robotics at RSS 2013

Temporal Logic Robustness for Signals

Linear Temporal Logics: Intuition

a – a now

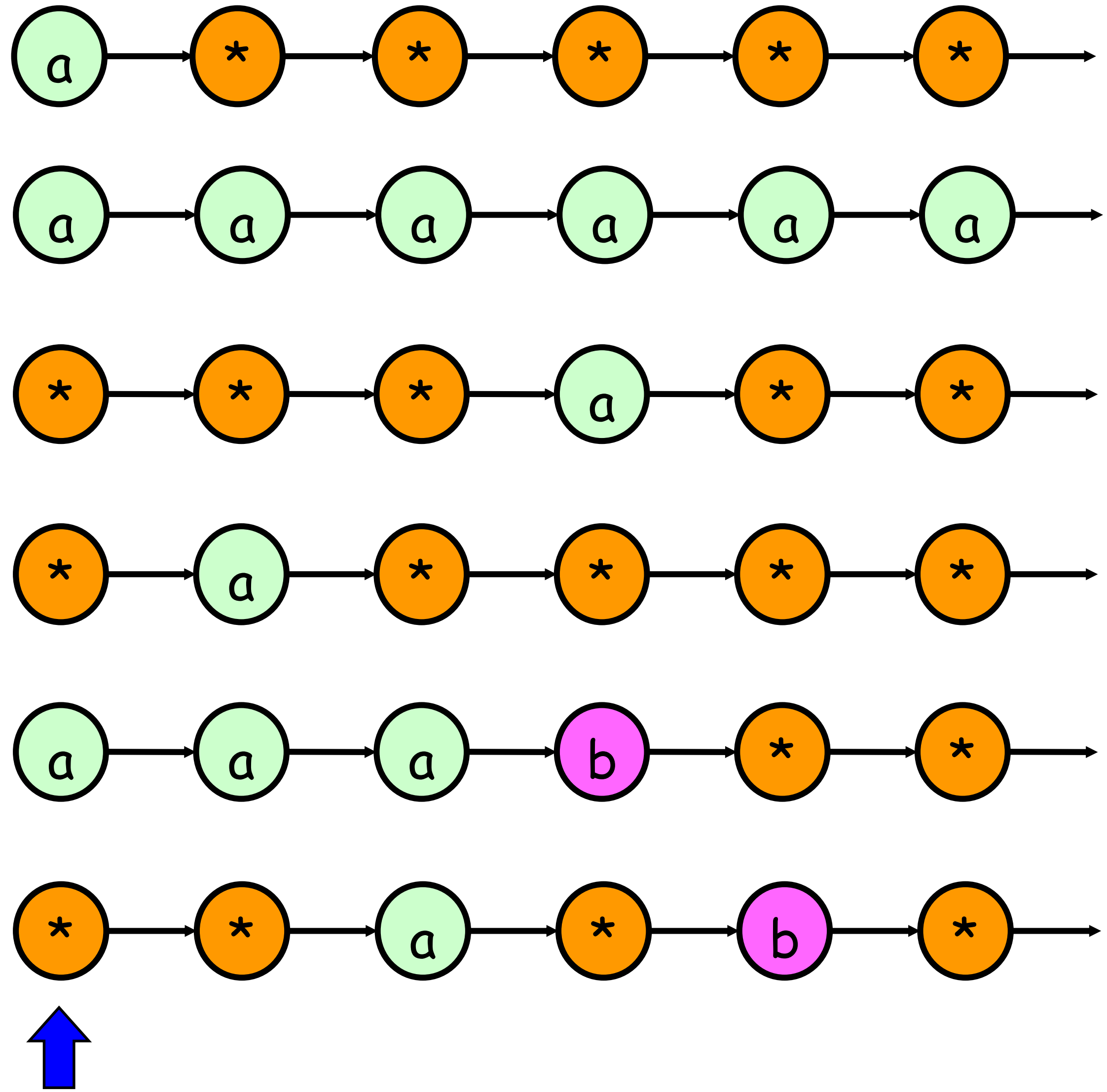
□ a - always a

◇ a – eventually a

○ a – next state a

a **U** b – a until b

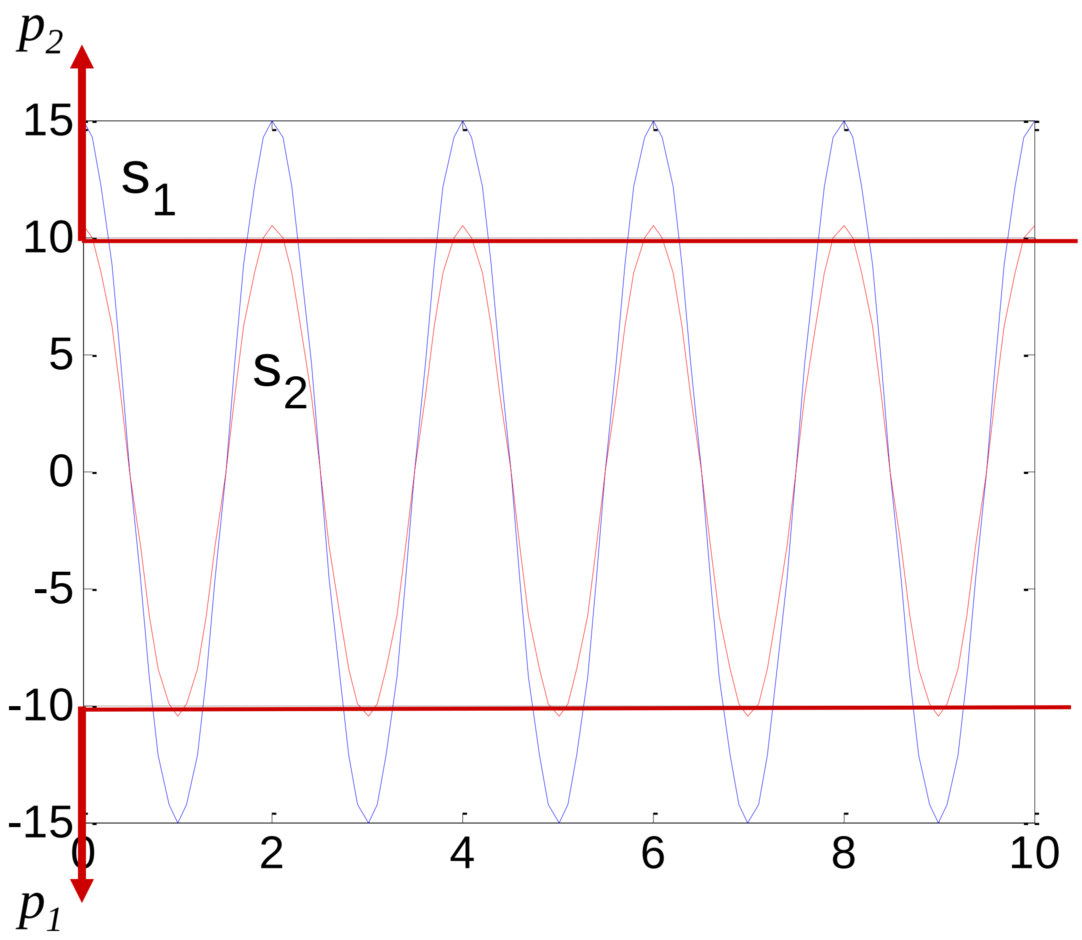
a **B** b – a before b



Two signals that satisfy the same spec, but …

MTL Spec:

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



… the same concepts applies to time robustness as well

Discrete-time Robust Semantics for MTL

$$
\begin{aligned}
[\![c]\!]_D(\mu,i) &:= c \\
[\![p]\!]_D(\mu,i) &:= \mathbf{Dist}_d(\sigma(i),\mathcal{O}(p)) \\
[\![\neg\phi_1]\!]_D(\mu,i) &:= -[\![\phi_1]\!]_D(\mu,i) \\
[\![\phi_1 \vee \phi_2]\!]_D(\mu,i) &:= [\![\phi_1]\!]_D(\mu,i) \sqcup [\![\phi_2]\!]_D(\mu,i) \\
[\![\phi_1 \mathcal{U}_{\mathcal{I}} \phi_2]\!]_D(\mu,i) &:= \bigsqcup_{j\in\tau^{-1}(\tau(i)+_R\mathcal{I})} \Big([\![\phi_2]\!]_D(\mu,j) \sqcap \bigsqcap_{i\le k<j} [\![\phi_1]\!]_D(\mu,k)\Big)
\end{aligned}
$$

timed trace
μ = (σ,τ)

Algorithm I (m-function fw_taliro)

- Based on formula re-writing
- Suitable for runtime monitoring algorithms
- Details Fainekos & Pappas, TCS 2009

Algorithm II (m-function dp_taliro)

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

Algorithms adopted and adapted from prior results by Thati, Rosu and Havelund

Discrete-time Robust Semantics for MTL

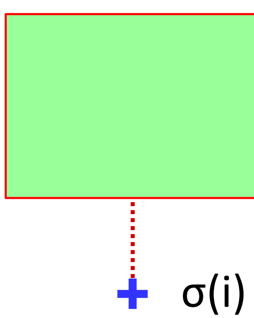
$$[\![p]\!]_D(\mu, i) := \mathbf{Dist}_d(\sigma(i), \mathcal{O}(p))$$

timed trace μ = (σ,τ)

More accurate →

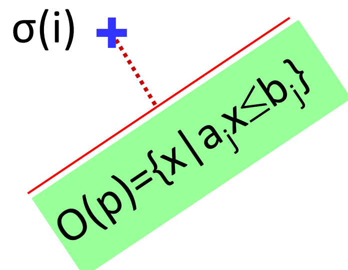
← Faster computations

Box sets & metric in $\mathbb{R}^n$

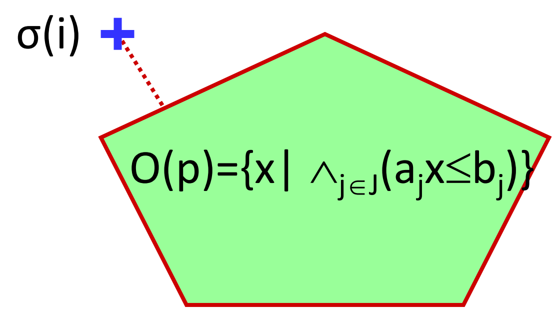


Subtractions & comparisons

Polyhedral sets & Euclidean metric in $\mathbb{R}^n$



Analytical solution



Solve Quadratic Program:

$\min (x-y)^T(x-y)$

s.t. $\wedge_j a_j x \leq b_j$

$\mathrm{Dist}(y, O(p)) = -\mathrm{minvalQP}$

TaLiRo Usage

- Operator correspondence

¬	∨	∧	→	↔	
!	\/	/\	->	<->	
○	⊙	□, G	◇, F	U	R
X	W	[]	<>	U	R

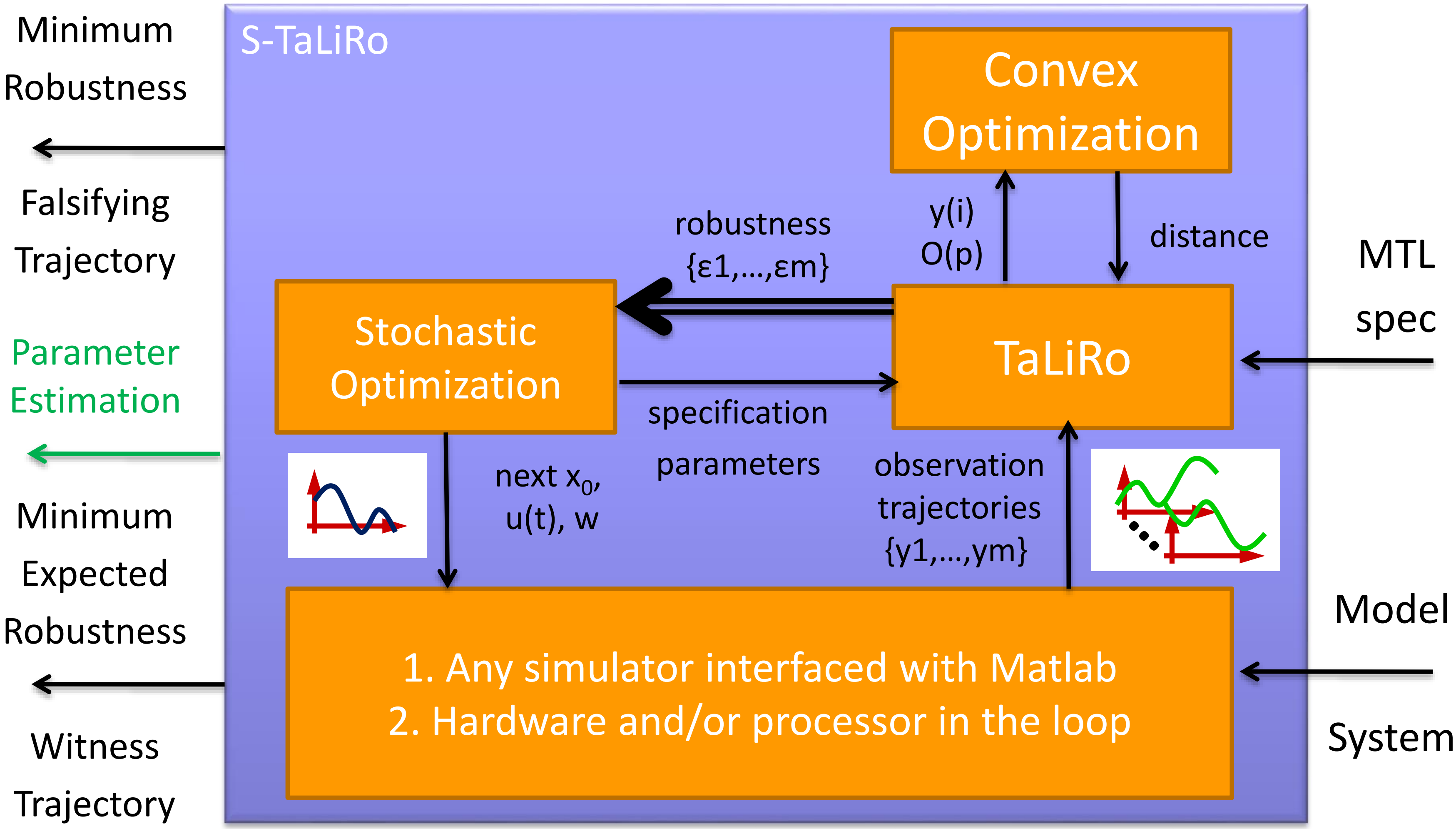
- Timing constraints are indicated by _<a,b> where $< \in \{(,[\}$ and $> \in \{),]\}$ and $a,b \in \mathbb{R}_+ \cup \{inf\}$

- Sets are given as Matlab objects with members
  - loc: the set of discrete modes or locations
  - A: the array that contains the vectors a_j
  - b: the array that contains the scalars b_j
  - str: the name of the atomic predicate

S-Taliro

Run setup_staliro.m

S-TaLiRo: Systems TemporAl LogIc RObustness

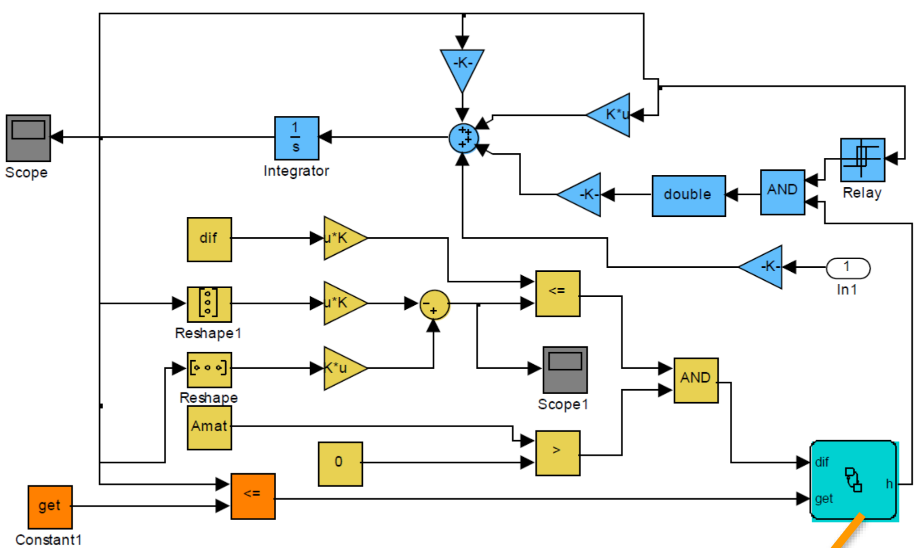


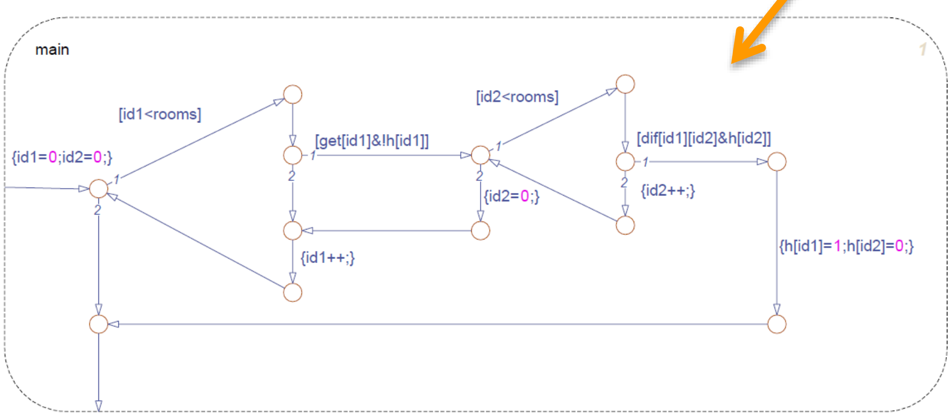
https://sites.google.com/a/asu.edu/s-taliro/

Example 0

see demos\staliro_heat_bench_demo_01.m

Room heating benchmark





- Room heating benchmark HEAT30 from Fehnker and Ivancic 2004

- A hybrid system with 10 continuous variables (10 rooms) and 3360 discrete locations (all possible combinations of 4 heaters in the 10 rooms).

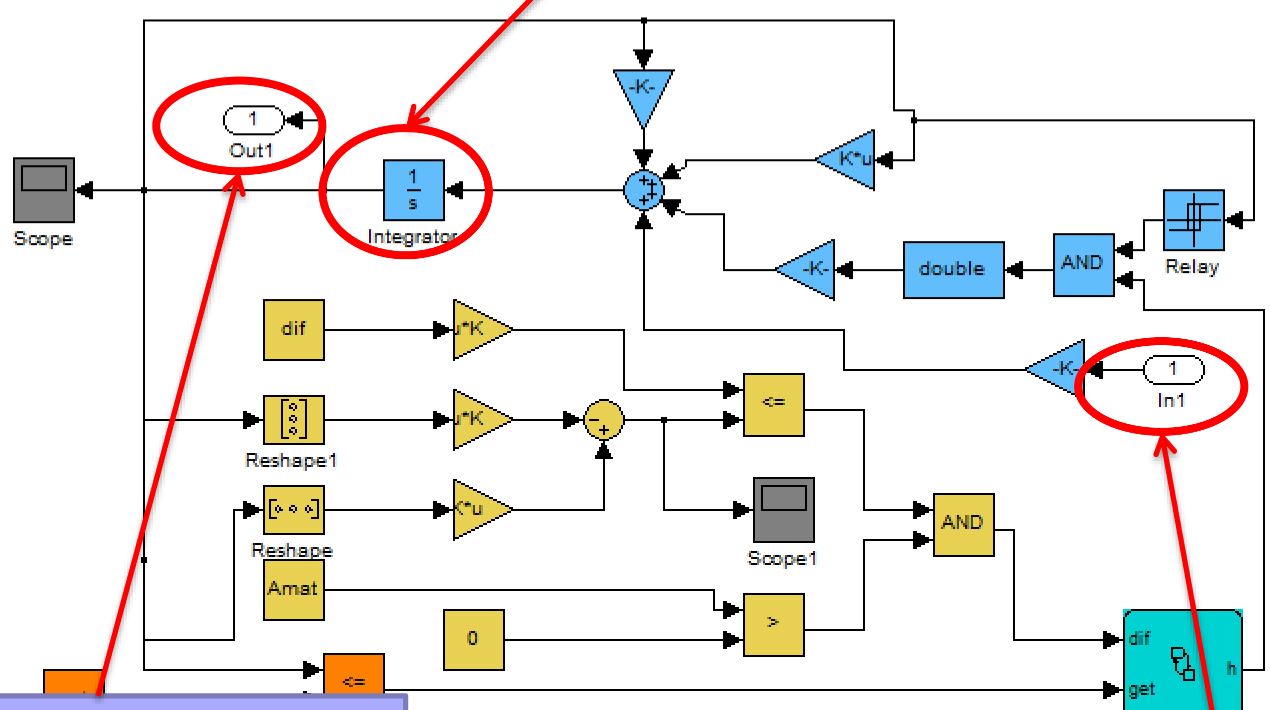
- The set of initial conditions is $[17, 18]^{10}$ and input signal u can range in [1, 2].

- The goal is to verify that no room temperature drops below [14.50 14.50 13.50 14.00 13.00 14.00 14.00 13.00 13.50 14:00]

Room Heating Benchmark

Initial condition ranges of the integrators are given as hypercubes

- If the initial conditions are not of interest, then they can be a single point or the default Simulink model values



It is recommended that specifications are expressed over output signals

If input signals are required, then input ports must be used.

Room Heating Benchmark

```
model = 'heat25830_staliro_01';
load heat30;
time = 24;
cp_array = 4;
input_range = [1 2];
X0 = [17*ones(10,1) 18*ones(10,1)];
phi = '[]p';
pred.str = 'p';
pred.A = -eye(10);
pred.b = -[14.50; 14.50; 13.50; 14.00; 13.00; 14.00; 14.00; 13.00; 13.50; 14.00];

opt = staliro_options();
opt.runs = 1;

results = staliro(model,X0,input_range,cp_array,phi,pred,time,opt);

figure(1)
clf
[T1,XT1,YT1,IT1] = SimSimulinkMdl(model,X0,input_range,cp_array,results.run(1).bestSample,time,opt);
subplot(2,1,1)
plot(T1,XT1)
title('State trajectories')
subplot(2,1,2)
plot(IT1(:,1),IT1(:,2))
title('Input Signal')
```

model name

simulation time

input signals

initial conditions

Always $x_1 \geq 14.5$ and … and $x_{10} \geq 14$

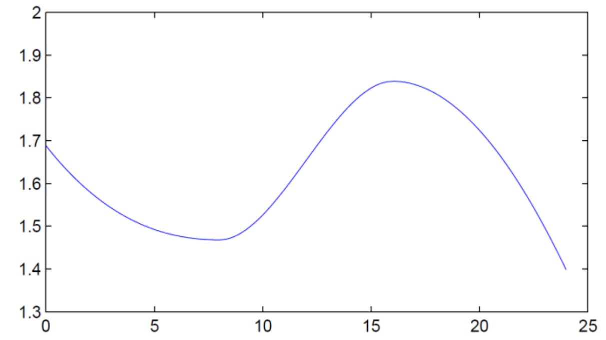
MTL spec

Other options

output/input signal visualization

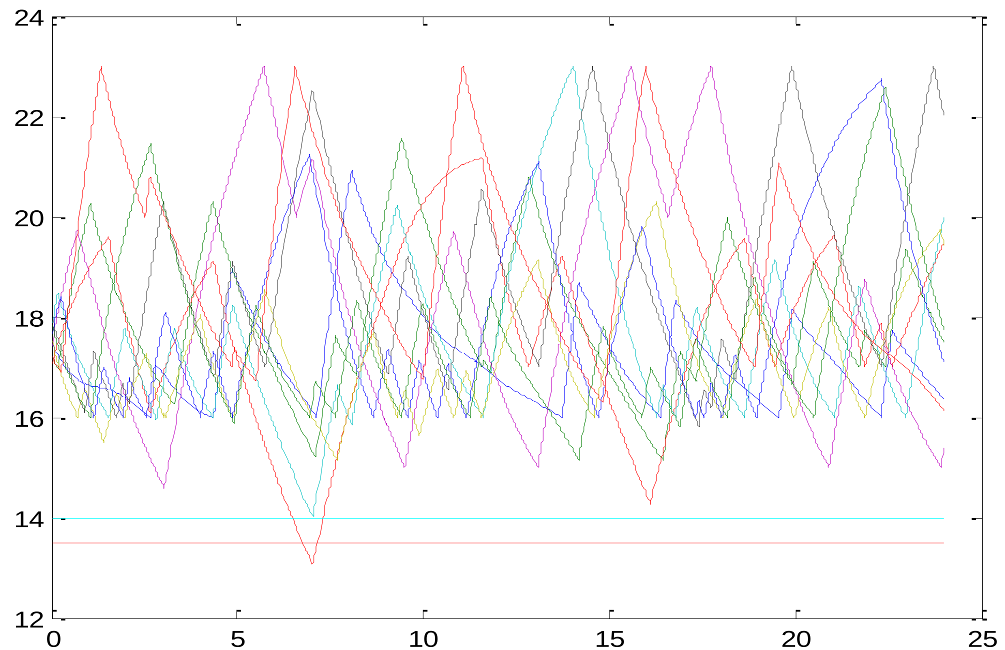
Room Heating Benchmark

Falsifying input signal and initial conditions



x_0 = [17.4705 17.2197 17.0643 17.8663 17.4316 17.5354 17.9900 17.6599 17.8402 17.2036 $]^T$

Resulting trajectories



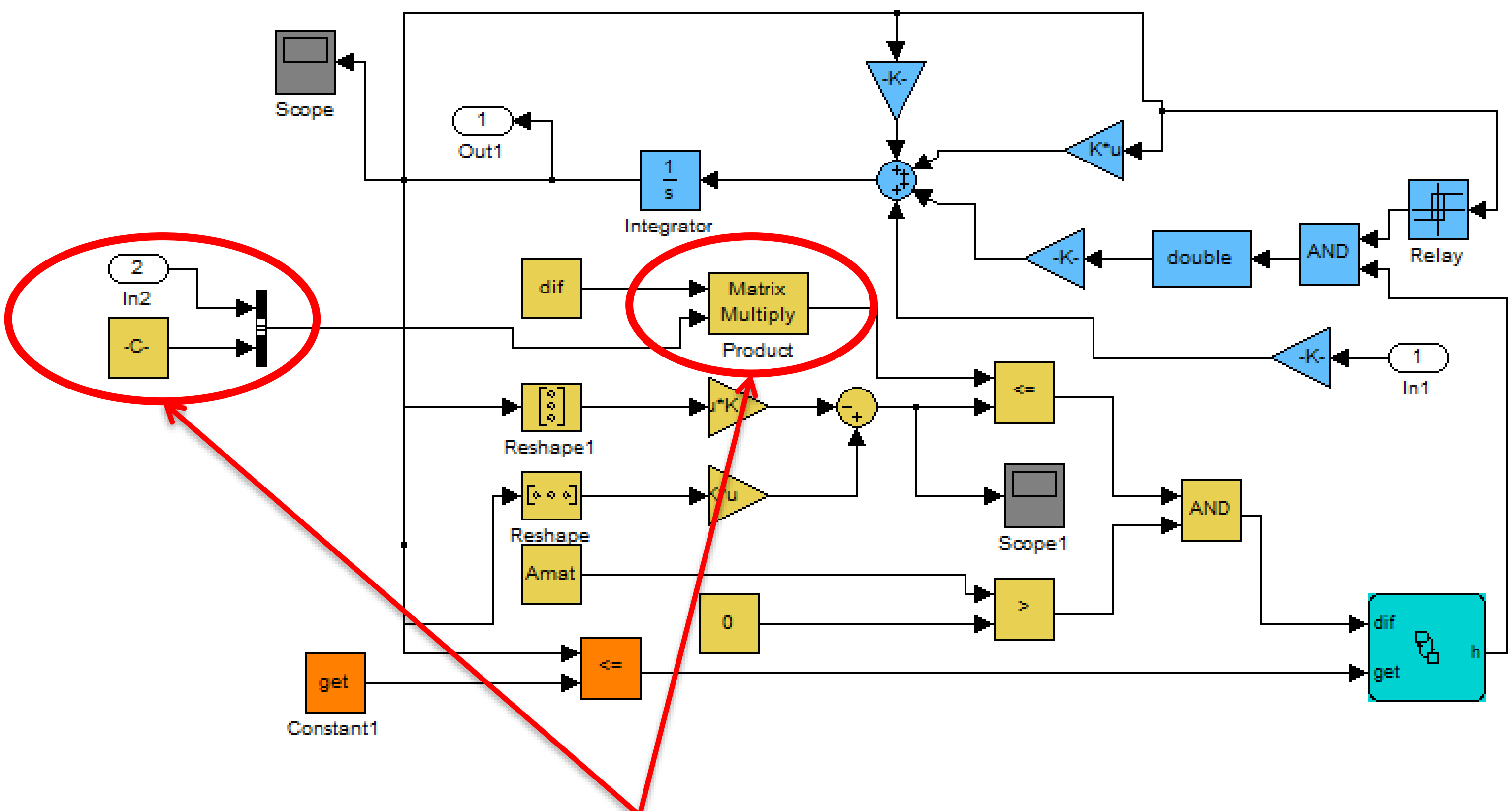
Robustness

-0.429

Example 1

see demos\staliro_heat_bench_demo_02.m

Simulink / Stateflow models



Input ports and data storage blocks can be used to modify system parameters. Use 'const' input signals with 1 control point for constant unknown parameters or any other interpolation function for time varying parameters.

Room Heating Benchmark

```
model = 'heat25830_staliro_02';
load heat30;
time = 24;
cp_array = [4 1];
input_range = [1 2; 0.8 1.2];
X0 = [17*ones(10,1) 18*ones(10,1)];
phi = '[]p';
pred.str = 'p';
pred.A = -eye(10);
pred.b = -[14.50; 14.50; 13.50; 14.00; 13.00; 14.00; 14.00; 13.00; 13.50; 14.00];

opt = staliro_options();
opt.runs = 1;
opt.interpolationtype = {'pchip', 'const'};

results = staliro(model,X0,input_range,cp_array,phi,pred

figure(1)
clf
[T1,XT1,YT1,IT1] = SimSimulinkMdl(model,X0,input_range,cp_array,results.run(1).bestSample,time,opt);
subplot(2,1,1)
plot(T1,XT1)
title('State trajectories')
subplot(2,1,2)
plot(IT1(:,1),IT1(:,2))
title('Input Signal')
```

Defining function interpolation through staliro_options

Example 2

see demos\staliro_demo_sa_nonlinear.m

Example of MTL Robustness Landscape

System:

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

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

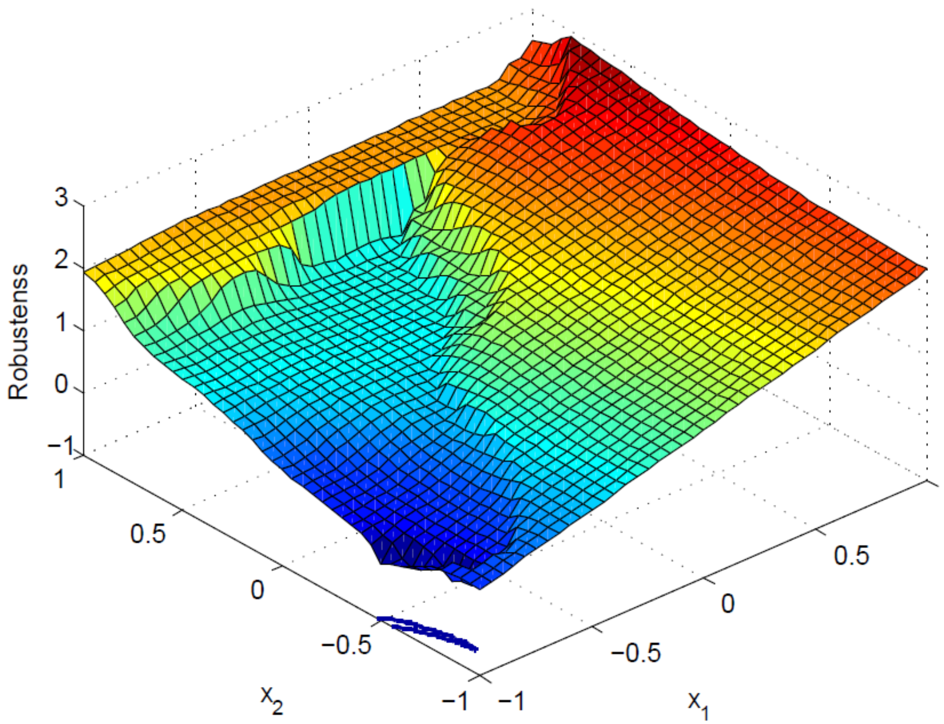
Initial conditions:

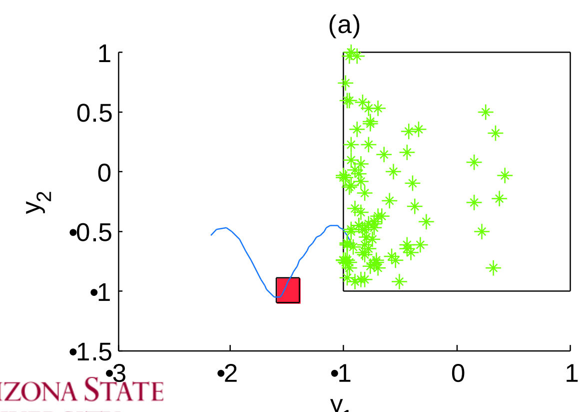
$$[-1,1]\times[-1,1]$$

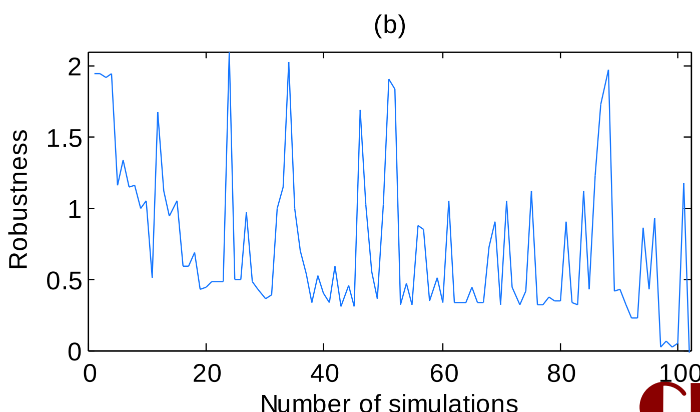
Specification:

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

where $O(a) = [-1.6,-1.4]\times[-1.1,-.9]$







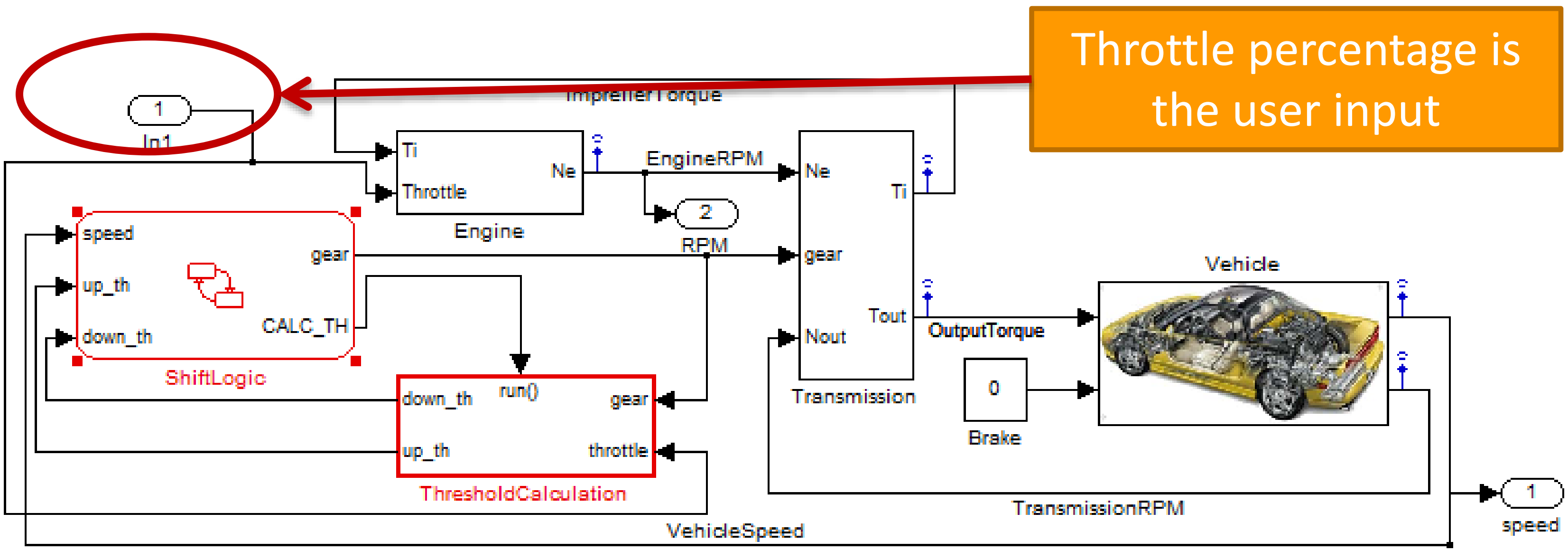
ARIZONA STATE UNIVERSITY

CPS Lab

Example 3

see staliro_demo_autotrans_01.m

Automatic Transmission Simulink Demo



Specification: The following 2 conditions should not occur during the first 60 sec of system operation:

1. the vehicle speed v exceeds 120km/h, and
2. the engine speed ω exceeds 4500RPM

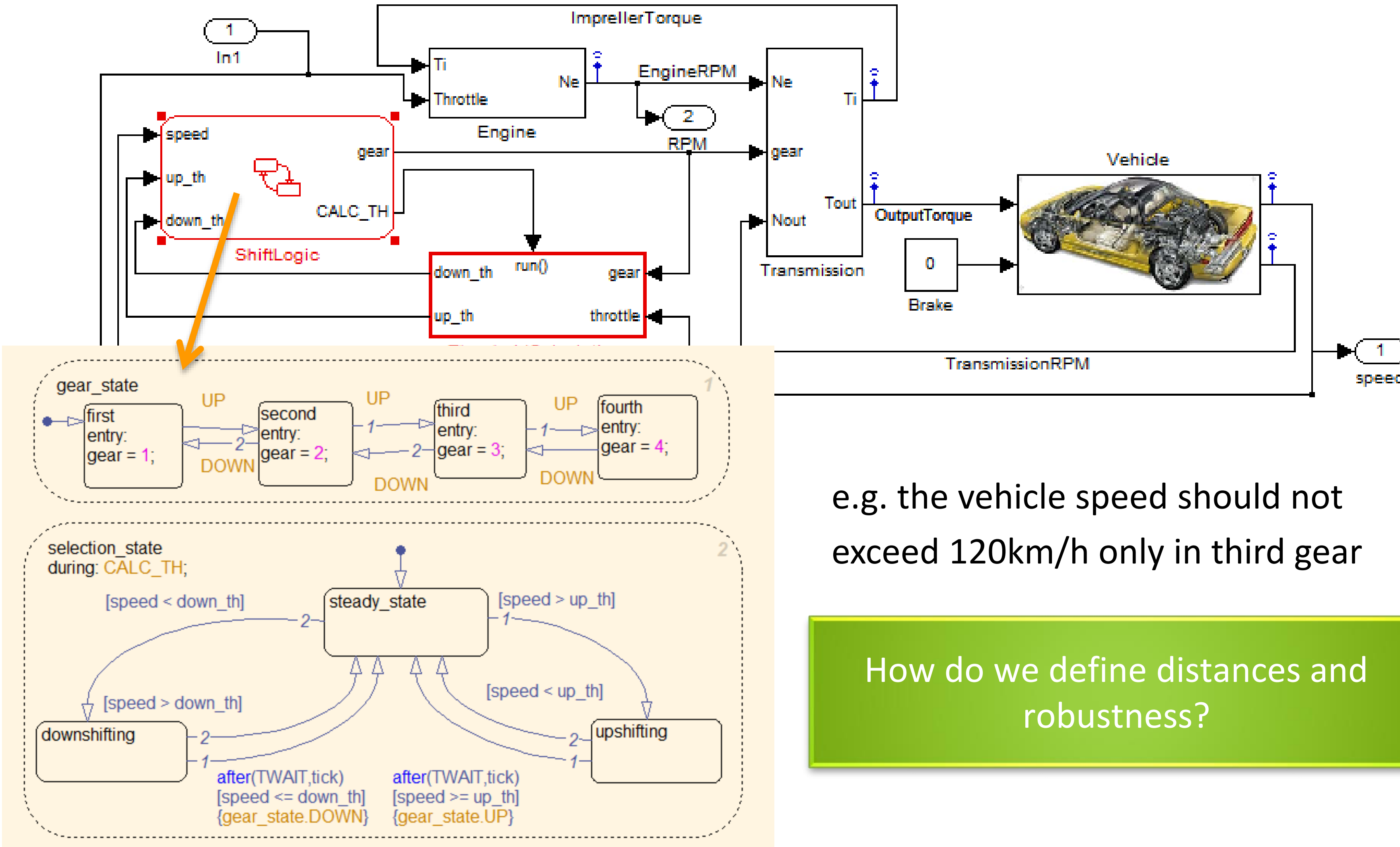
Example 4

see staliro_demo_autotrans_02.m

Matlab package MatlabBGL is required:

http://www.mathworks.com/matlabcentral/fileexchange/10922

But the requirements are not only on the continuous state space …

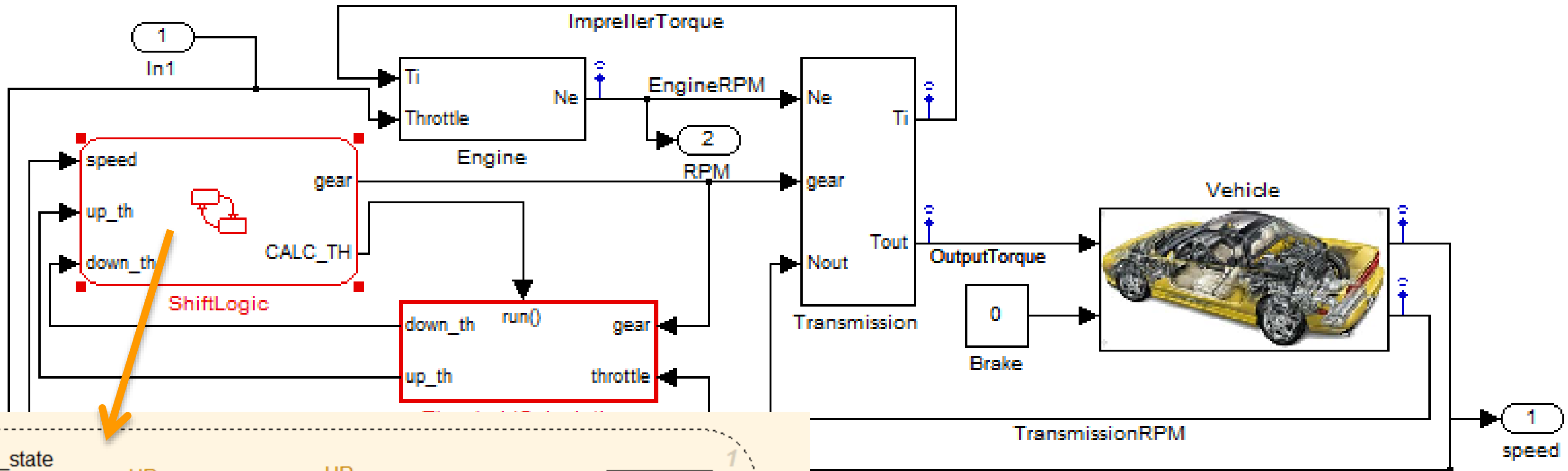


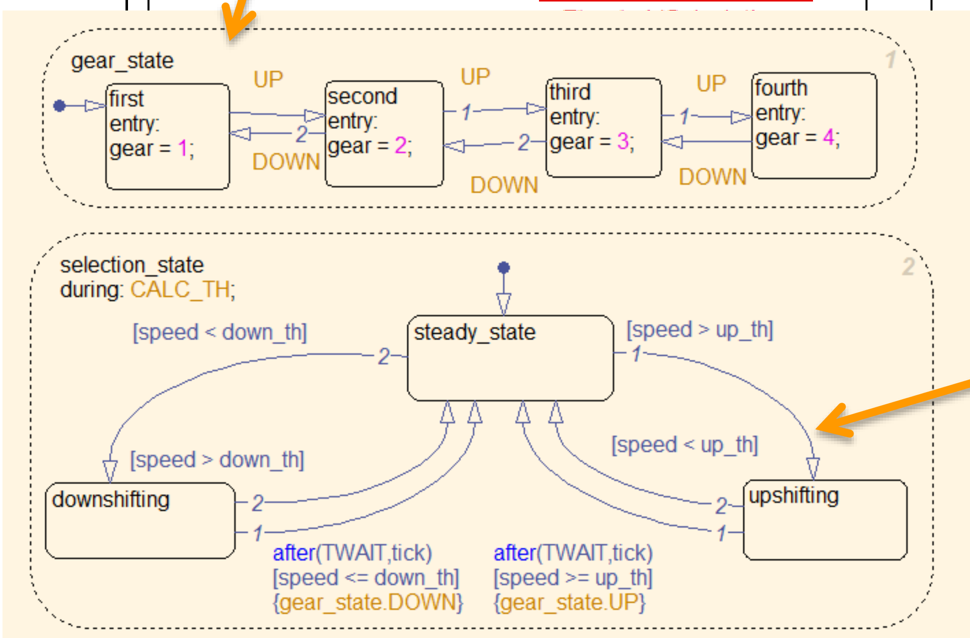
e.g. the vehicle speed should not exceed 120km/h only in third gear

How do we define distances and robustness?

Concurrent FSM

But the requirements are not only on the continuous state space …



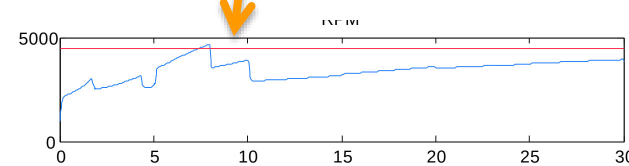


Define "hybrid" robustness values:

(ε, δ)

Path distance* on the graph

Distance on the continuous state



*the shortest path to reach the set of modes

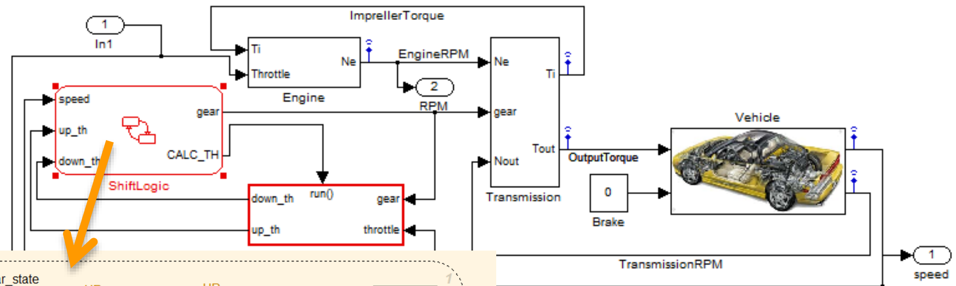
Example 5

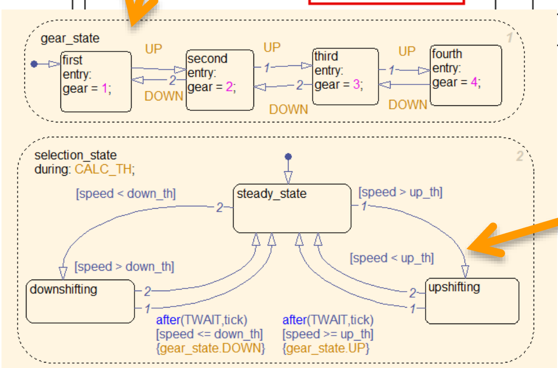
see staliro_demo_autotrans_03.m

Matlab package MatlabBGL is required:

http://www.mathworks.com/matlabcentral/fileexchange/10922

In case the stochastic optimizer does not support "hybrid" robustness value search…



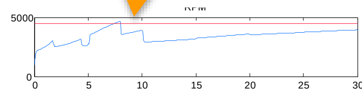


Define "hybrid" robustness values:

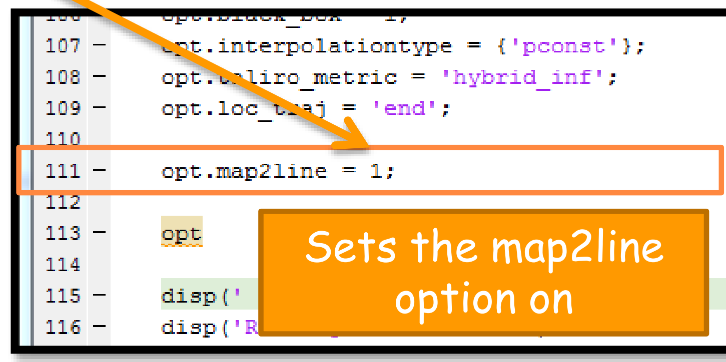
(ε, δ)

Path distance* on the graph

Distance on the continuous state



ADD



Sets the map2line option on

map2line - maps a "hybrid" robustness value to the real line

```
Running S-TaLiRo ...

Run number 1
Best ==> <0,342.3044>
Best ==> <0,218.3047>
Best ==> <0,-4.6151>
FALSIFIED!
```

```
Running S-TaLiRo ...

Run number 1
Best ==> 0.93683
Best ==> 0.79743
Best ==> -0.023071
FALSIFIED!
```

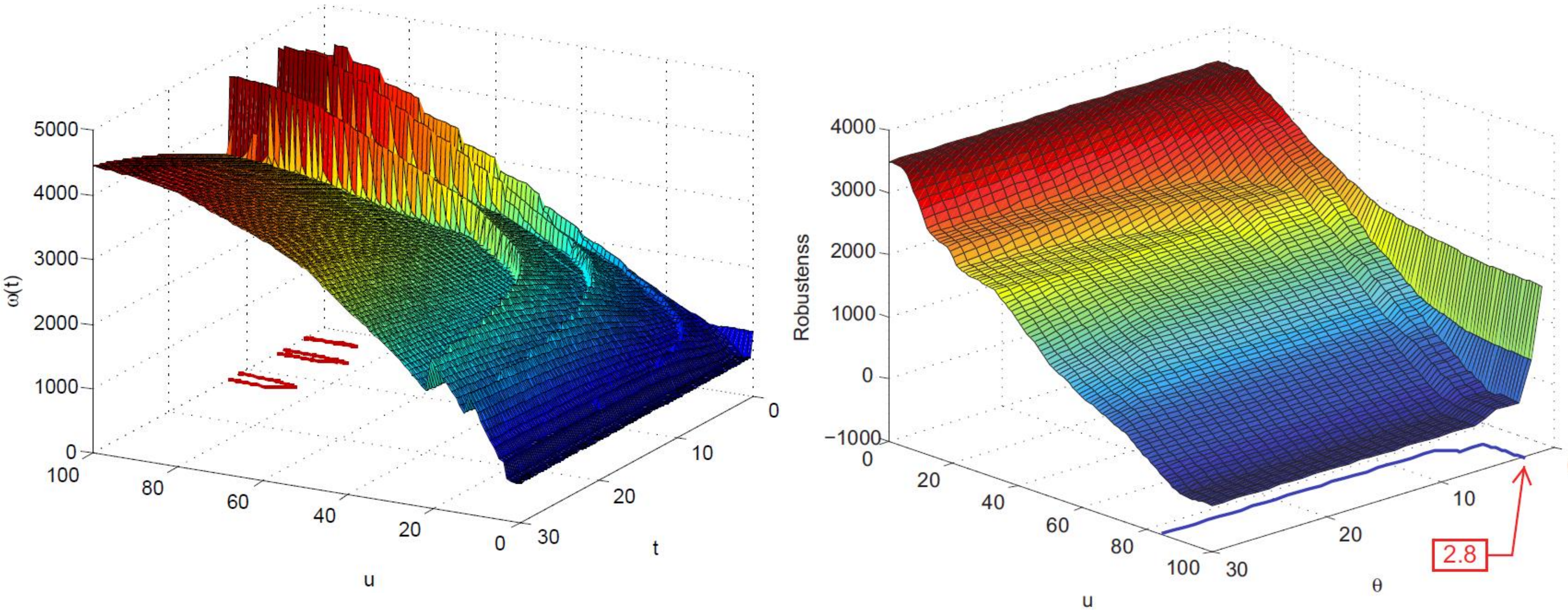
Example 6

see staliro_demo_autotrans_parameter.m

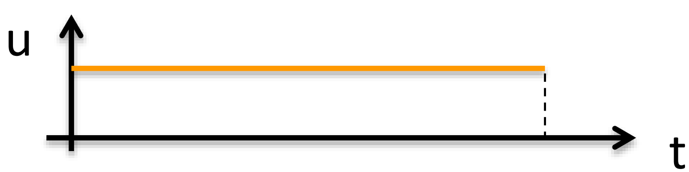
Example for parameter estimation

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

I.e., find the parameter θ such that no matter what the input u is ω is always below 4500



Throttle % parameterization with 1 control variable for easy visualization



Example 7

see staliro_demo_quadrotor_2D.m

Matlab packages required:

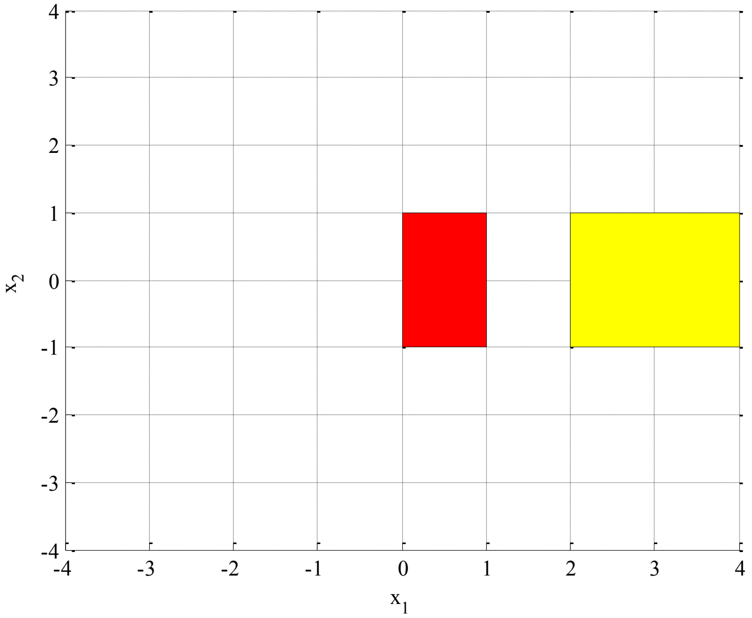
Robotics Toolbox

http://www.petercorke.com/Robotics_Toolbox.html

Multi-Parametric Toolbox (MPT)

http://control.ee.ethz.ch/~mpt/

Potential application: MTL Path Planning



Specifications:

1. eventually yellow and always not red
2. eventually always yellow and always for time between 4 and 5 not red

New Features in S-TaLiRo 1.5

New Features in S-TaLiRo 1.5

Parallel Simulations

The new version of S-Taliro utilizes the Matlab parallel computing toolbox to enable parallel simulations for the Cross Entropy (CE_Taliro) and Uniform Random (UR_Taliro) optimization solvers. Ex. To run 4 simulations in parallel set opt.n_workers = 4.

Example: ***demos/ver1_5/demo_new_features_1_5_beta_02.m***

Output Dimension Projection

S_Taliro now can consider only a subset of the systems' outputs. Ex. If the system has 5 outputs $[y_1 \ldots y_5]$ and we only need to focus on $[y_1\ y_2]$ then we can set the option opt.dim_proj = $[1\ 2]$. This simplifies the predicate definitions since they are defined only over the outputs 1 and 2.

Example: ***demos/ver1_5/demo_new_features_1_5_beta_01.m***

Output Under sampling for Robustness Computation

In case the output array is too large and is causing robustness metric computation to take too long you can under sample the output sequence. Ex. opt.taliro_undersampling_factor = 50 will consider every 50th value in the output sequence when calculating the robustness metric.

Example: ***demos/ver1_5/demo_new_features_1_5_beta_01.m***

New Features in S-TaLiRo 1.5

Added support for hybrid distance metric for CE_Taliro

Previously, to use the hybrid distance metric with CE_Taliro, it was necessary to set opt.map2line = 1 in order to convert the hybrid distance value to the real line. This is not necessary anymore since CE_Taliro now supports the hybrid distance metric.

Example: ***demos/ver1_5/demo_new_features_1_5_beta_02.m***

Random Number Generator Seed for S-Taliro

The option to set the seed for the random number generator is added. This enables the user to reproduce simulation results. Ex. opt.seed = 1 sets the rng seed to 1.

Example: ***demos/ver1_5/demo_new_features_1_5_beta_01.m***

Save Intermediate S-Taliro Results

This option enables the user to save intermediate results after each run. This is especially useful in the case when runs take a long time. To use this option set opt.save_intermediate_results = 1.

Example: ***demos/ver1_5/demo_new_features_1_5_beta_01.m***

New Features in S-TaLiRo 1.5

Varying Time Control Points

Added a feature to expand the search space to include varying time control points with the simulated annealing optimization engine. If opt.varying_cp_times = 0, the control points for the input signal are equally distributed in simulation time. If opt.varying_cp_times = 1, the placement of the control points for the input signal is included in the search space.

Example: ***demos/ver1_5/demo_new_features_1_5_beta_01.m***

Troubleshooting & Tips

Tips

- **Predicates:**
  - A polyhedral set S can be defined either as a constraint of the form S_P : $Ax \le b$, where A is an array and b is a vector, or as a conjunction of halfspaces $S_H : \wedge_i\, a_i x \le b_i$.
  - Even though S_P and S_H define the same set, i.e., $A = [a_1; \ldots; a_n]$ and $b = [b_1, \ldots, b_n]$, computing the distance to S_P or S_H is performed under different algorithms:
    1. The distance of a point x to S_P is computed by solving a quadratic program
    2. The distance of a point x to SH is computed analytically by computing the distance of x to each halspace and, then, taking the maximum.
  - 1 is more accurate, but is 1-2 orders of magnitude slower than 2.
  - Thus, we highly recommend replacing a predicate that represents a polyhedron by a conjunction of predicates representing the corresponding half-spaces.

Tips

- **Use output ports instead of state space variables**
  For Simulink models, we recommend using output ports to define your specifications even if you need to analyze all the continuous state variables of the system. *State variables can change order when compiling a Simulink model.*

Troubleshooting

- **Robustness value is not what you expect**
  When developing a new Simulink model, do not forget to remove the option “Limit data points to last:”. If this option is set then the analysis is going to be based only on the last samples created by the model
  - Simulation > Configuration Parameters > Data Import/Export > Limit data points to last
- **Error message that Simulink model outputs are empty or the robustness value is not what you expect**
  After Matlab version 2011a make sure that the signal logging format is set to ModelDataLogs.
  - Simulation > Configuration Parameters > Data Import/Export > Signal logging format

Troubleshooting

- **Error returned when number of workers set is greater than 1 and input model type is set to blackbox**
  To fix this issue you should compile the model first. That can be done using the `modelname([],[],[],'compile')` command.

Other functionality included in S-Taliro …

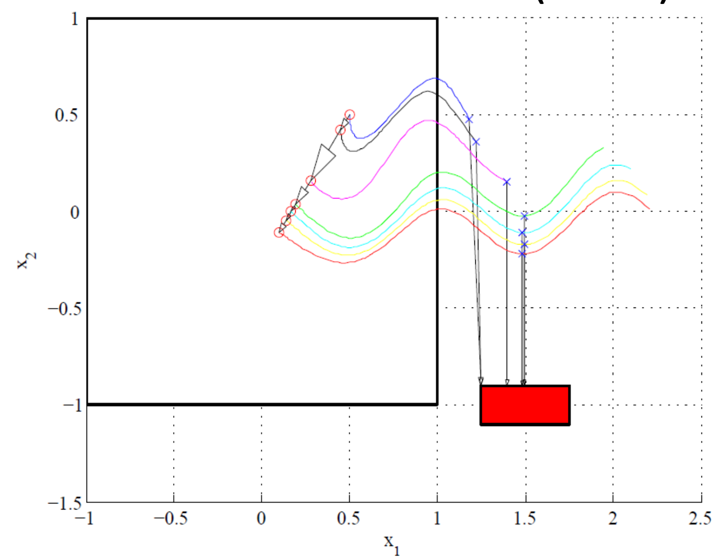
Other functionality

- Expected best/worst robustness
  - *See demo demos/ver1_5/staliro_demo_stoch_cold_chain.m*
- Robust test generation using auto-bisimulation functions
  - *Under the folder ha_robust_tester*
- Time robustness computation (*dp_t_taliro*)
  - *Included only as alpha version. It is not fully tested.*
- Conformance testing
  - *See demo demos/ver1_5/staliro_demo_conformance.m*

Other functionality

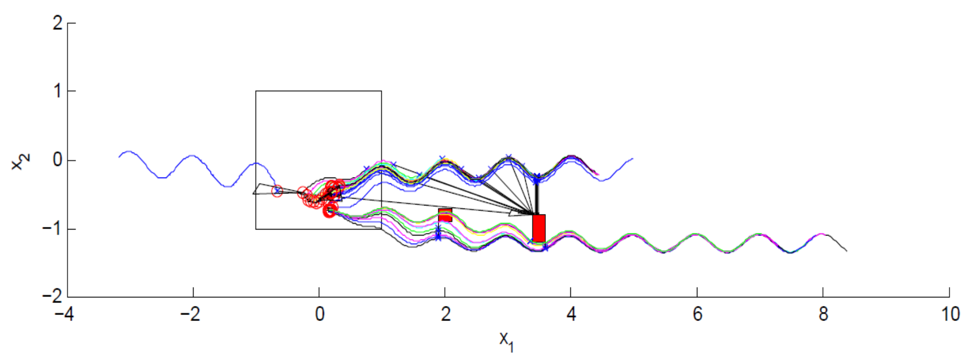
- dp_taliro returns the optimizing sample point and predicate.
  - This is useful for implementing gradient descent optimization algorithms and for user debugging

$G(\neg red)$



- Descent directions for linear hybrid automata and smooth non-linear systems
  - *Not included in version 1.5*
  - *See Abbas & Fainekos, ACC 13*

$G(small\_red \Rightarrow G_{[0,1]} \neg big\_red)$



Acknowledgements

Students
- Houssam Abbas – PhD (ECEE)
- Adel Dokhanchi – PhD (CIDSE)
- Bardh Hoxha – PhD (CIDSE)

Former Students
- Y. Annapureddy - MS (CIDSE)
- Hengyi Yang – MS (CIDSE)

Main Academic Collaborators
- Sriram Sankaranarayanan (U of Colorado)

Academic Collaborators
- Truong Nghiem (UPenn)
- George Pappas (UPenn)

Industry Collaborators
- Jyotirmoy V. Deshmukh (Toyota)
- Aarti Gupta (NEC Labs)
- Franjo Ivancic (NEC Labs)
- James Kapinski (Toyota)
- Koichi Ueda (Toyota)
- Hakan Yazarel (Toyota)

Sponsors

NSF

CNS-1017074
CNS-1116136
CNS-1319560

TOYOTA

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

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