Hybrid System Falsification and Reinforcement Learning

Formal Method for Cyber-Physical Systems

Clovis Eberhart   David Sprunger

National Institute of Technology, Japan

SOKENDAI lesson, July 1, 8, and 22

Quick reminder

- Falsification:
  - method to find counterexamples to a property,
  - useful in the world of formal methods,
  - black-box method,
  - relies on optimisation algorithms.
- Hybrid system:
  - continuous and discrete parameters,
  - non-linear behaviour,
  - very expressive.
- Formulas:
  - expressed in a temporal logic,
  - boolean and robustness semantics.

1. Refining robustness

2. Time staging

3. Coverage-based falsification

Table of Contents

Refining robustness

Why?

- more expressivity (i.e., finer modelling)
- more techniques (e.g., optimisation techniques work better)

Attention

- more expressivity $\rightsquigarrow$ more complex algorithms

Refining robustness

Why?

- more expressivity (i.e., finer modelling)
- more techniques (e.g., optimisation techniques work better)

Attention

- more expressivity $\rightsquigarrow$ more complex algorithms (here, however, only sliding-window algorithms)

Space-time robustness

Donzé, A. and Maler O. *Robust satisfaction of temporal logic over real-valued signals*. FORMATS 2010.

Until now, robustness is spatial.

Problems:

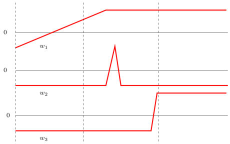
Space-time robustness

Donzé, A. and Maler O. *Robust satisfaction of temporal logic over real-valued signals*. FORMATS 2010.

Until now, robustness is spatial.

Problems:

- all these signals verify $\Diamond_{[a,b]} x > 0$ with the same robustness



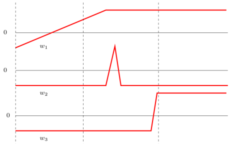
Space-time robustness

Donzé, A. and Maler O. *Robust satisfaction of temporal logic over real-valued signals*. FORMATS 2010.

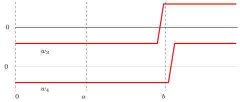
Until now, robustness is spatial.

Problems:

- all these signals verify $\Diamond_{[a,b]} x > 0$ with the same robustness



- the similarity between these two signals is lost when computing $\rho(\sigma, \Diamond_{[a,b]} x > 0)$



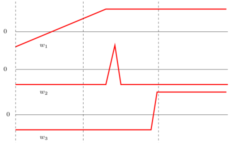
Space-time robustness

Donzé, A. and Maler O. *Robust satisfaction of temporal logic over real-valued signals*. FORMATS 2010.

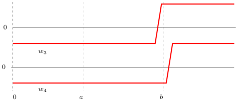
Until now, robustness is spatial.

Problems:

- all these signals verify $\Diamond_{[a,b]} x > 0$ with the same robustness



- the similarity between these two signals is lost when computing $\rho(\sigma, \Diamond_{[a,b]} x > 0)$



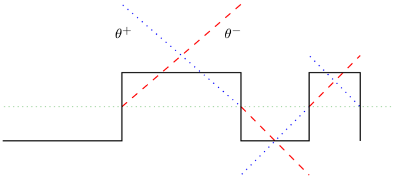
$\rightsquigarrow$ missing a temporal component

Adding time

Assumption: set $P = \{p_1, \ldots, p_n\}$ of atomic propositions.
Standard boolean semantics: $\chi(\sigma, \varphi, t)$.

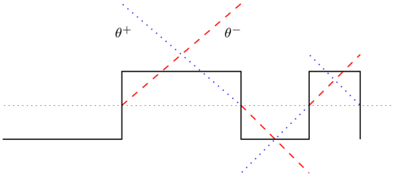
Time robustness

$\theta^-(\sigma, p, t) =$
$\chi(\sigma, p, t) \cdot \max\{d \geq 0 \mid \forall t' \in [t-d, t].\chi(\sigma, p, t') = \chi(\sigma, p, t)\}$
$\theta^+(\sigma, p, t) =$
$\chi(\sigma, p, t) \cdot \max\{d \geq 0 \mid \forall t' \in [t, t+d].\chi(\sigma, p, t') = \chi(\sigma, p, t)\}$
$\theta^s(\sigma, \neg\varphi, t) = -\theta^s(\sigma, \varphi, t)$
...



Interpreting θ^+ and θ^-

- $\theta^+(\sigma, \varphi, t) = s > 0$: $\sigma \models \varphi$ for at least time s
- $\theta^+(\sigma, \varphi, t) = s < 0$: $\sigma \not\models \varphi$ for at least time s
- $\theta^-(\sigma, \varphi, t) = s > 0$: $\sigma \models \varphi$ since at least time s
- $\theta^-(\sigma, \varphi, t) = s < 0$: $\sigma \not\models \varphi$ since at least time s



Space-time Robustness

Assumption: atomic propositions are functions (e.g., $x^2 + y^2$).
Standard robustness semantics: $\rho(\sigma, \varphi, t)$.

Space-time robustness

For any $c \in \mathbb{R}$:

- $\theta_c^+(\sigma, f, t) = \theta^+(\chi_c(\sigma, f, t))$,
- $\theta_c^-(\sigma, f, t) = \theta^-(\chi_c(\sigma, f, t))$,
- $\theta_c^s(\sigma, \neg\varphi, t) = -\theta_c^s(\sigma, \varphi, t)$.
- ...

Interpretation:

- $\theta_c^+(\sigma, \varphi, t) = s > 0$: $\rho(\sigma, \varphi, t) > c$ for at least time s,
- ...

Space-time Robustness

Assumption: atomic propositions are functions (e.g., $x^2 + y^2$).
Standard robustness semantics: $\rho(\sigma, \varphi, t)$.

Space-time robustness

For any $c \in \mathbb{R}$:

- $\theta_c^+(\sigma, f, t) = \theta^+(\chi_c(\sigma, f, t))$,
- $\theta_c^-(\sigma, f, t) = \theta^-(\chi_c(\sigma, f, t))$,
- $\theta_c^s(\sigma, \neg\varphi, t) = -\theta_c^s(\sigma, \varphi, t)$.
- ...

Interpretation:

- $\theta_c^+(\sigma, \varphi, t) = s > 0$: $\rho(\sigma, \varphi, t) > c$ for at least time s,
- ...

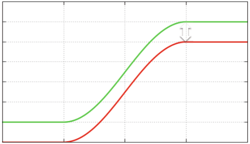
Remarks:

- hopefully more efficient
- how to choose c?
- not more expressive

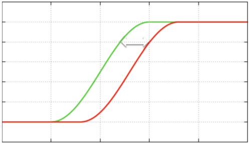
More flexibility

Akazaki T. and Hasuo I. *Time robustness in MTL and expressivity in hybrid system falsification*. CAV 2015.

- Spatial robustness:



- Temporal robustness:



AvSTL

Syntax

$$AP = x < r \mid x \leq r \mid x > r \mid x \geq r$$

$$\varphi = \top \mid \bot \mid AP \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \, \mathsf{U}_I \, \varphi \mid \varphi \, \mathsf{R}_I \, \varphi \mid \varphi \, \overline{\mathsf{U}_I} \, \varphi \mid \varphi \, \overline{\mathsf{R}_I} \, \varphi$$

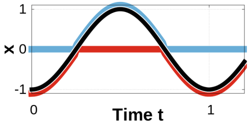
Semantics

- $\rho^+(\sigma, x < r, t) = \max\{0, r - \sigma(x)(t)\}$
- $\rho^-(\sigma, x < r, t) = \min\{0, r - \sigma(x)(t)\}$
- ...
- $\rho^+(\sigma, \neg\varphi, t) = \rho^-(\sigma, \varphi, t)$
- $\rho^+(\sigma, \varphi \, \overline{\mathsf{U}_{[a,b]}} \, \psi, t) = \frac{1}{b-a} \int_a^b \rho(\sigma, \varphi \, \mathsf{U}_{[a,b] \cap [0,\tau]} \, \psi, t) d\tau$
- ...

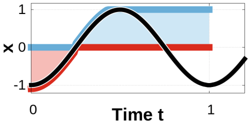
Example

Robustnesses: ρ^+, ρ^-

- $\varphi = x \geq 0$:



- $\varphi = \overline{\mathsf{F}_I}(x \geq 0)$:

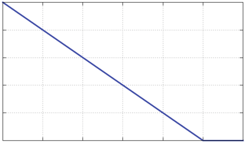


Consequences:

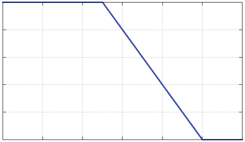
- temporal aspects
- spatial aspects

Expressivity

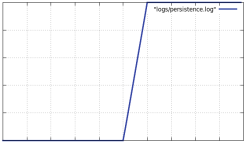
- expeditiousness: $\overline{\mathsf{F}_{[0,a]}}\,\varphi$



- deadline: $\mathsf{F}_{[0,a]}\,\varphi \vee \overline{\mathsf{F}_{[a,b]}}\,\varphi$



- persistence: $\mathsf{G}_{[0,a]}\,\varphi \wedge \overline{\mathsf{G}_{[a,b]}}\,\varphi$



Experimental results

Problem 1	$T=20$			$T=30$			$T=40$		
Specification to be falsified	Succ. $/\mathbf{100}$	Iter. (Succ.)	Time (Succ.)	Succ. $/\mathbf{100}$	Iter. (Succ.)	Time (Succ.)	Succ. $/\mathbf{100}$	Iter. (Succ.)	Time (Succ.)
$\Diamond_{[0,T]}(\omega \geq 2000)$	100	128.8	20.2	81	440.9	82.5	32	834.3	162.9
		128.8	20.2		309.7	59.0		482.2	94.4
$\overline{\Diamond}_{[0,T]}(\omega \geq 2000)$	100	123.9	22.9	98	249.8	46.1	81	539.6	110.9
		123.9	22.9		234.5	43.4		431.6	89.2

Problem 3	$T=4$			$T=4.5$			$T=5$		
Specification to be falsified	Succ. $/\mathbf{20}$	Iter. (Succ.)	Time (Succ.)	Succ. $/\mathbf{20}$	Iter. (Succ.)	Time (Succ.)	Succ. $/\mathbf{20}$	Iter. (Succ.)	Time (Succ.)
$\Box_{[0,T]}\neg\mathtt{gear}_4$	0	1000	166.9	11	742.8	122.9	18	449.0	71.8
		–	–		532.3	87.5		387.7	61.9
$\Box_{[0,T]}\neg\mathtt{gear}_4$	17	570.1	94.0	20	250.5	40.3	20	107.5	17.6
$\wedge\overline{\Box}_{[T,10]}\neg\mathtt{gear}_4$		494.2	81.8		250.5	40.3		107.5	17.6

Problem 5 ($\varepsilon = 0.04$)	$T=0.8$			$T=1$			$T=2$		
Specification to be falsified	Succ. $/\mathbf{20}$	Iter. (Succ.)	Time (Succ.)	Succ. $/\mathbf{20}$	Iter. (Succ.)	Time (Succ.)	Succ. $/\mathbf{20}$	Iter. (Succ.)	Time (Succ.)
$\bigwedge_{i=1,\dots,4}\Box\Big((\neg\mathtt{gear}_i \wedge \Diamond_{[0,\varepsilon]}\mathtt{gear}_i)$	2	972.5	402.5	19	356.8	155.6	20	27.4	11.8
$\to (\Box_{[\varepsilon,T+\varepsilon]}\mathtt{gear}_i)\Big)$		724.5	297.8		322.9	140.9		27.4	11.8
$\bigwedge_{i=1,\dots,4}\Box\Big((\neg\mathtt{gear}_i \wedge \Diamond_{[0,\varepsilon]}\mathtt{gear}_i)$	12	561.1	349.1	20	93.1	57.8	20	42.7	26.9
$\to (\Box_{[\varepsilon,T+\varepsilon]}\mathtt{gear}_i \wedge \overline{\Box}_{[T+\varepsilon,5]}\mathtt{gear}_i)\Big)$		268.5	167.3		93.1	57.8		42.7	26.9

Table of Contents

Time staging

Zhang, Z., Ernst, G., Sedwards, S., Arcaini, P., and Hasuo, I. *Two-Layered Falsification of Hybrid Systems Guided by Monte Carlo Tree Search*. EMSOFT 2018.

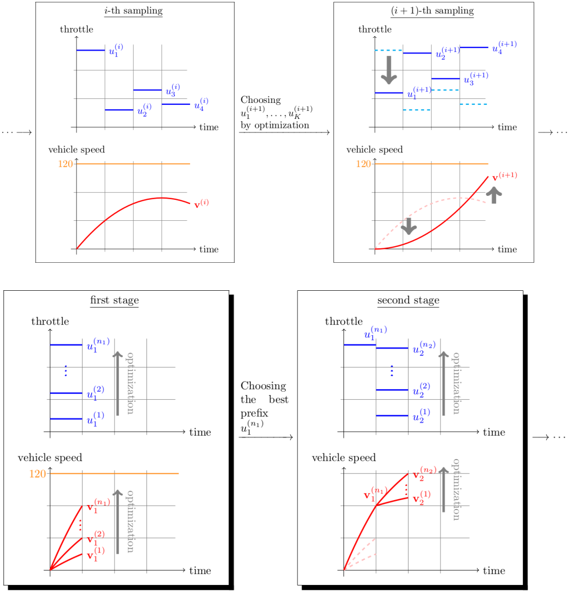
Ernst, G., Sedwards, S., Zhang, Z., and Hasuo, I. *Fast Falsification of Hybrid Systems using Probabilistically Adaptive Input*. QEST 2019.

Idea

- σ_{out} causally dependent on σ_{in}
- optimisation methods blind to this dependence

$\rightsquigarrow$ modify the algorithm to take it into account

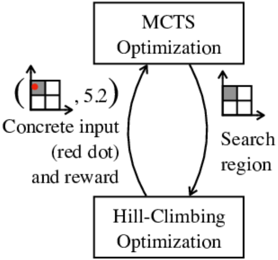
A picture is worth a thousand words



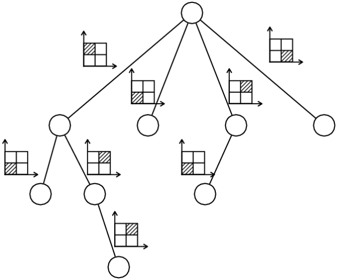
High-Level Algorithm

Alternate between:

- Monte-Carlo Tree Search to find a good zone,
- hill-climbing to find a good point in the zone.



Monte-Carlo Tree Search



Each node equipped with:

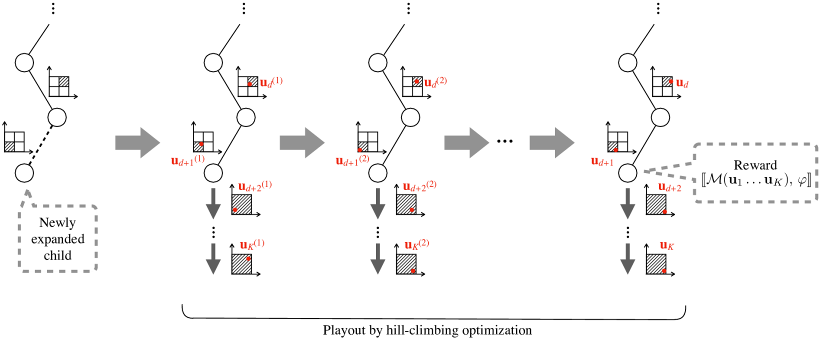
- robustness estimate,
- number of visits.

To choose a node, balance between:

- an exploitation score (bigger with smaller robustness estimates),
- an exploration score (bigger with fewer visits to the node).

Robustness estimates

To get robustness estimates: complete the signal by pure hill-climbing.
For example, for a newly-expanded node:



Experimental results

		Parameters			AT model										AFC model				FFR model	
					S1		S2		S3		S4		S5		Sbasic		Sstable		Strap	
	Algorithm	M_b	TO_{po}	c	succ.	time	succ.	time	succ.	time	succ.	time	succ.	time	succ.	time	succ.	time	succ.	time
	Random				10/10	108.9	10/10	289.1	1/10	301.1	0/10	-	0/10	-	6/10	278.7	10/10	242.6	4/10	409.3
CMA-ES	Breach				10/10	21.9	6/10	30.3	**10/10**	**193.9**	4/10	208.8	3/10	75.5	**10/10**	**111.7**	3/10	256.3	**10/10**	**119.8**
CMA-ES	Basic	40	15	0.20	10/10	15.8	10/10	108.5	10/10	697.1	7/10	786.8	9/10	384.4	10/10	182.0	7/10	336.9	10/10	338.0
CMA-ES	P.W.	40	15	0.20	**10/10**	**10.8**	**10/10**	**65.7**	10/10	728.6	**7/10**	**767.8**	**10/10**	**648.1**	10/10	177.1	**8/10**	**272.9**	10/10	473.9
GNM	Breach				**10/10**	**5.4**	**10/10**	**151.4**	0/10	-	0/10	-	0/10	-	**10/10**	**171.4**	0/10	-	0/10	-
GNM	Basic	20	5	0.20	10/10	12.4	10/10	162.3	**10/10**	**185.6**	7/10	261.9	7/10	163.7	10/10	227.1	2/10	378.5	**10/10**	**162.2**
GNM	P.W.	20	5	0.05	10/10	60.8	9/10	110.7	8/10	211.2	**8/10**	**313.0**	**10/10**	**178.7**	10/10	252.0	**6/10**	**153.2**	6/10	197.4
SA	Breach				**10/10**	**160.1**	0/10	-	3/10	383.7	0/10	-	3/10	80.4	0/10	-	6/10	307.0	3/10	92.8
SA	Basic	20	15	0.05	10/10	264.8	9/10	236.1	**8/10**	**385.6**	**8/10**	**505.3**	7/10	341.2	5/10	391.3	**8/10**	**273.8**	**10/10**	**273.2**
SA	P.W.	40	15	0.20	10/10	208.7	**10/10**	**377.6**	8/10	666.0	7/10	795.4	**10/10**	**624.2**	**8/10**	**665.7**	6/10	293.7	10/10	390.9

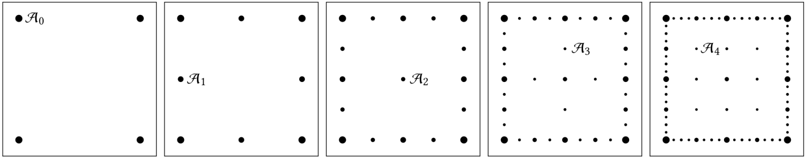
Interpretation: MTCS explores more, so:

- better results on hard problems
- slower on simple problems

Adaptive Las Vegas Tree Search

To build signal σ incrementally:

- randomly choose a level l of "granularity" (initially, low granularity is favoured),
- choose $\sigma' = \mathcal{D}_l(\sigma)$, where $\mathcal{D}_l$ chooses "finer" signals for large l (shorter time, more precise value),
- adapt $\mathcal{D}_l$ according to $\rho(\sigma\sigma', \varphi, t)$.



Experimental results

Formula	Random succ. /50	Random iter. M	Random iter. SD	Breach: CMA-ES succ. /50	Breach: CMA-ES iter. M	Breach: CMA-ES iter. SD	FalStar: aLVTS succ. /50	FalStar: aLVTS iter. M	FalStar: aLVTS iter. SD
AT1	43	106.6	83.9	50	39.7	23.6	50	8.5	6.7
AT2 ($i = 3$)	50	41.0	36.7	50	13.2	9.1	50	33.4	27.5
AT2 ($i = 4$)	49	67.0	60.8	6	17.8	15.9	50	23.4	22.5
AT3	19	151.1	98.1	50	145.2	63.0	50	86.3	52.1
AT4 (a)	36	117.3	71.8	50	97.0	47.7	50	22.8	10.6
AT4 (b)	2	117.7	9.2	49	46.7	58.0	50	47.6	23.5
Summary AT	199	95.3	47.9	255	42.8	29.0	300	29.2	19.4
AFC27	15	129.1	90.8	41	121.0	49.3	50	3.9	4.3

Interpretation:

- falsifying signals are often coarse, or slight variations of such, so explored very fast by this algorithm,
- robustness scores that concern discrete variables are hard to manipulate for optimisation algorithm (not continuous)

Table of Contents

Idea

Adimoolam, A., Dang, T., Donzé, A., Kapinski, J., and Jin, X. *Classification and coverage-based falsification for embedded control systems*. CAV 2017.

Trade-off between:

- define a coverage metric of the input space,
- alternate between:
  - a global search to classify the search space into zones,
  - local searches on the promising zones to converge to a minimum.

High-level algorithm

Input: $t_{\max}$
Output: a u such that $\mathcal{M}(u) \not\models \varphi$
S = sample N points at random;
R = zones(S);
while $t < t_{max}$ **do**
    subdivide(R);
    S += biased-sampling(R);
    S += singularity-sampling(R);
    S += local-search(R);
end
for u *in* S **do**
    if $\rho(u) < 0$ **then**
        return u
    end
end
return None

Subdivision

Goal: divide the search space into rectangles with different average robustnesses.

Input: R a list of rectangles, S a list of sampled points, K a threshold

Output: a list of subdivided rectangles

for r *in* R **do**
  pop(R, r);
  if $|S \cap r| > K$ **then**
    $H = \text{argmin}(\Gamma_H(R, S), H \text{ hyperplane})$;
    push(R,$r \cap H^-$,$r \cap H^+$);
  end
end

$$\Gamma_{(d,r,p)}(R, S) = \sum_{x \in S \cap R} e_{(d,r,p)}(x)$$
$$e_{(d,r,p)}(x) = \max\{p(\rho(x) - \mu)(x_d - r), 0\}$$

Samplings

Biased sampling

Goal: increase coverage and decrease robustness.

Idea: sample according to a weighted sum of two distributions:

- P_c^i: proportional to the numbers of unoccupied cells in rectangle R_i,
- P_r^i: takes into consideration how the robustness of sampled points varies from the average.

Singularity sampling

Goal: sample more in rectangles with "singular" samples (robustness much lower than average in rectangle).

Local search

Goal: converge to a minimum faster by using local search with a good seed.

Experimental results

Solver	Seed	Computation time (secs)		Falsification	
		PTC	Aut. Trans	PTC	Aut. Trans
Hyperplane classification + CMA-ES-Breach	0	2891	996	✓	✓
	5000	2364	1382	✓	✓
	10000	2101	1720	✓	✓
	15000	2271	1355	✓	✓
CMA-ES-Breach	0	T.O. (5000)	T.O. (2000)		
	5000	T.O. (5000)	1302		✓
	10000	T.O. (5000)	T.O. (2000)		
	15000	T.O. (5000)	1325		✓
Grid based random sampling	0	T.O. (5000)	T.O. (2000)		
	5000	T.O. (5000)	T.O. (2000)		
	10000	3766	T.O. (2000)	✓	
	15000	268	T.O. (2000)	✓	
S-TaLiRo (Simulated Annealing)		4481	T.O. (3000)	✓	
S-TaLiRo (Simulated Annealing)		4481	Default stopping (3300)	✓	

Interpretation: other methods got caught in local minima.

Conclusion

- different notions of robustness:
  - can be more expressive
  - can make algorithms more efficient
- time staging:
  - explores more
  - hence can resolve harder problems
- coverage-based falsification:
  - theoretical result (if there exists an ε-robust counterexample, there is a grid size such that will find it)
  - coverage helps falsification by exploring more, thus avoiding local minima