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

1. Refining robustness
2. Time staging
3. Coverage-based falsification
Refining robustness

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

Attention
- more expressivity $\Rightarrow$ more complex algorithms
Refining robustness

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

Attention
- more expressivity $\leadsto$ more complex algorithms (here, however, only sliding-window algorithms)
Space-time robustness


Until now, robustness is spatial.

Problems:
Space-time robustness


Until now, robustness is spatial.

Problems:

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


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


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 $\rho(\sigma, \Diamond_{[a,b]} x > 0)$

\(\sim\) 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**

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

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

\[
\theta^{s}(\sigma, \neg \varphi, t) = -\theta^{s}(\sigma, \varphi, t)
\]

\[\ldots\]
Interpreting $\theta^+$ and $\theta^-$

- $\theta^+(\sigma, \varphi, t) = s > 0$: $\sigma \models \varphi$ 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 $c \in \mathbb{R}$:

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

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 $c \in \mathbb{R}$:
- $\theta^+_c(\sigma, f, t) = \theta^+_c(\chi_c(\sigma, f, t))$,
- $\theta^-_c(\sigma, f, t) = \theta^-_c(\chi_c(\sigma, f, t))$,
- $\theta^s_c(\sigma, \neg \varphi, t) = -\theta^s_c(\sigma, \varphi, t)$.
- ...

**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

\[ AP = x < r \mid x \leq r \mid x > r \mid x \geq r \]
\[ \varphi = T \mid \bot \mid AP \mid \neg \varphi \mid \varphi \lor \varphi \mid \varphi \land \varphi \mid \varphi U I \varphi \mid \varphi R I \varphi \mid \varphi \overline{U} I \varphi \mid \varphi \overline{R} I \varphi \]

Semantics

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

Robustnesses: $\rho^+, \rho^-$

- $\varphi = x \geq 0$:

- $\varphi = \overline{F_t}(x \geq 0)$:

Consequences:
- temporal aspects
- spatial aspects
Expressivity

- expeditiousness: $F_{[0,a]} \varphi$

- deadline: $F_{[0,a]} \varphi \lor \overline{F_{[a,b]} \varphi}$

- persistence: $G_{[0,a]} \varphi \land \overline{G_{[a,b]} \varphi}$
# Experimental results

<table>
<thead>
<tr>
<th>Problem 1</th>
<th>( T = 20 )</th>
<th>( T = 30 )</th>
<th>( T = 40 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specification to be falsified</td>
<td>( \diamond [0, T] (\omega \geq 2000) )</td>
<td>Succ. / 100</td>
<td>Iter. (Succ.)</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>128.8</td>
<td>20.2</td>
</tr>
<tr>
<td></td>
<td>128.8</td>
<td>128.8</td>
<td>20.2</td>
</tr>
<tr>
<td>Specification to be falsified</td>
<td>( \lozenge [0, T] (\omega \geq 2000) )</td>
<td>Succ. / 100</td>
<td>Iter. (Succ.)</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>123.9</td>
<td>22.9</td>
</tr>
<tr>
<td></td>
<td>123.9</td>
<td>123.9</td>
<td>22.9</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Problem 3</th>
<th>( T = 4 )</th>
<th>( T = 4.5 )</th>
<th>( T = 5 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specification to be falsified</td>
<td>( \square [0, T] \neg \text{gear}_4 )</td>
<td>Succ. / 20</td>
<td>Iter. (Succ.)</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>1000</td>
<td>166.9</td>
</tr>
<tr>
<td></td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Specification to be falsified</td>
<td>( \square [0, T] \neg \text{gear}_4 \land \square [T, 10] \neg \text{gear}_4 )</td>
<td>Succ. / 20</td>
<td>Iter. (Succ.)</td>
</tr>
<tr>
<td></td>
<td>17</td>
<td>570.1</td>
<td>94.0</td>
</tr>
<tr>
<td></td>
<td>494.2</td>
<td>81.8</td>
<td>250.5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Problem 5 (( \varepsilon = 0.04 ))</th>
<th>( T = 0.8 )</th>
<th>( T = 1 )</th>
<th>( T = 2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specification to be falsified</td>
<td>( \bigwedge_{i=1, \ldots, 4} \square (\neg \text{gear}_i \land \diamond [0, \varepsilon] \text{gear}_i) )</td>
<td>Succ. / 20</td>
<td>Iter. (Succ.)</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>972.5</td>
<td>402.5</td>
</tr>
<tr>
<td></td>
<td>724.5</td>
<td>297.8</td>
<td>322.9</td>
</tr>
<tr>
<td>Specification to be falsified</td>
<td>( \bigwedge_{i=1, \ldots, 4} \square (\neg \text{gear}_i \land \diamond [0, \varepsilon] \text{gear}_i) \rightarrow (\square [\varepsilon, T+\varepsilon] \text{gear}_i \land \square [T+\varepsilon, 5] \text{gear}_i) )</td>
<td>Succ. / 20</td>
<td>Iter. (Succ.)</td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>561.1</td>
<td>349.1</td>
</tr>
<tr>
<td></td>
<td>268.5</td>
<td>167.3</td>
<td>93.1</td>
</tr>
</tbody>
</table>
Time staging


**Idea**

- $\sigma_{\text{out}}$ causally dependent on $\sigma_{\text{in}}$
- optimisation methods blind to this dependence

$\mapsto$ 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

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Parameters</th>
<th>AT model</th>
<th>AFC model</th>
<th>FFR model</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>M_b TOpo c</td>
<td>S1 succ. time</td>
<td>S2 succ. time</td>
<td>S3 succ. time</td>
</tr>
<tr>
<td>Random</td>
<td></td>
<td>10/10 108.9</td>
<td>10/10 289.1</td>
<td>1/10 301.1</td>
</tr>
<tr>
<td>CMA-ES Breach</td>
<td>40 15 0.20</td>
<td>10/10 193.9</td>
<td>4/10 208.8</td>
<td>3/10 75.5</td>
</tr>
<tr>
<td>Basic</td>
<td></td>
<td>10/10 15.8</td>
<td>10/10 108.5</td>
<td>10/10 697.1</td>
</tr>
<tr>
<td>P.W.</td>
<td></td>
<td>10/10 10.8</td>
<td>10/10 65.7</td>
<td>10/10 728.6</td>
</tr>
<tr>
<td>GNM Breach</td>
<td>20 5 0.20</td>
<td>10/10 5.4</td>
<td>10/10 151.4</td>
<td>0/10 -</td>
</tr>
<tr>
<td>Basic</td>
<td></td>
<td>10/10 12.4</td>
<td>10/10 162.3</td>
<td>10/10 185.6</td>
</tr>
<tr>
<td>P.W.</td>
<td></td>
<td>10/10 60.8</td>
<td>9/10 110.7</td>
<td>8/10 211.2</td>
</tr>
<tr>
<td>SA Breach</td>
<td>20 15 0.05</td>
<td>10/10 160.1</td>
<td>0/10 -</td>
<td>3/10 383.7</td>
</tr>
<tr>
<td>Basic</td>
<td></td>
<td>10/10 264.8</td>
<td>9/10 236.1</td>
<td>8/10 385.6</td>
</tr>
<tr>
<td>P.W.</td>
<td></td>
<td>10/10 208.7</td>
<td>10/10 377.6</td>
<td>8/10 666.0</td>
</tr>
</tbody>
</table>

**Interpretation:** MTCS explores more, so:
- **better results on hard problems**
- **slower on simple problems**
Adaptive Las Vegas Tree Search

To build signal $\sigma$ 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 $\rho(\sigma\sigma', \varphi, t)$. 

\[ \begin{align*}
\mathcal{A}_0 & \quad \mathcal{A}_1 \\
\mathcal{A}_2 & \quad \mathcal{A}_3 \\
\mathcal{A}_4 &
\end{align*} \]
Experimental results

<table>
<thead>
<tr>
<th>Formula</th>
<th>Random</th>
<th></th>
<th></th>
<th>Breach: CMA-ES</th>
<th></th>
<th></th>
<th>FalSTAR: aLVTS</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>succ.</td>
<td>iter.</td>
<td>succ.</td>
<td>iter.</td>
<td>succ.</td>
<td>iter.</td>
<td>succ.</td>
<td>iter.</td>
</tr>
<tr>
<td>AT1</td>
<td>43</td>
<td>106.6</td>
<td>83.9</td>
<td>50</td>
<td>39.7</td>
<td>23.6</td>
<td>50</td>
<td>8.5</td>
</tr>
<tr>
<td>AT2 (i = 3)</td>
<td>50</td>
<td>41.0</td>
<td>36.7</td>
<td>50</td>
<td>13.2</td>
<td>9.1</td>
<td>50</td>
<td>33.4</td>
</tr>
<tr>
<td>AT2 (i = 4)</td>
<td>49</td>
<td>67.0</td>
<td>60.8</td>
<td>6</td>
<td>17.8</td>
<td>15.9</td>
<td>50</td>
<td>23.4</td>
</tr>
<tr>
<td>AT3</td>
<td>19</td>
<td>151.1</td>
<td>98.1</td>
<td>50</td>
<td>145.2</td>
<td>63.0</td>
<td>50</td>
<td>86.3</td>
</tr>
<tr>
<td>AT4 (a)</td>
<td>36</td>
<td>117.3</td>
<td>71.8</td>
<td>50</td>
<td>97.0</td>
<td>47.7</td>
<td>50</td>
<td>22.8</td>
</tr>
<tr>
<td>AT4 (b)</td>
<td>2</td>
<td>117.7</td>
<td>9.2</td>
<td>49</td>
<td>46.7</td>
<td>58.0</td>
<td>50</td>
<td>47.6</td>
</tr>
<tr>
<td>Summary AT</td>
<td>199</td>
<td>95.3</td>
<td>47.9</td>
<td>255</td>
<td>42.8</td>
<td>29.0</td>
<td>300</td>
<td>29.2</td>
</tr>
<tr>
<td>AFC27</td>
<td>15</td>
<td>129.1</td>
<td>90.8</td>
<td>41</td>
<td>121.0</td>
<td>49.3</td>
<td>50</td>
<td>3.9</td>
</tr>
</tbody>
</table>

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

1 Refining robustness
2 Time staging
3 Coverage-based falsification
Idea


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:** \( t_{\text{max}} \)

**Output:** a \( u \) such that \( \mathcal{M}(u) \not\equiv \varphi \)

\( S = \) sample \( N \) points at random;

\( R = \text{zones}( S ); \)

**while** \( t < t_{\text{max}} \) **do**

- **subdivide**( \( R \) );
- \( S += \text{biased-sampling}( R ) \);
- \( S += \text{singularity-sampling}( R ) \);
- \( S += \text{local-search}( R ) \);

**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
  - $H = \arg\min(\Gamma_H(R, S), H$ hyperplane$)$;
  - push($R, r \cap H^-, r \cap H^+$);
- end

end

$$\Gamma_{(d,r,p)}(R, S) = \sum_{x \in S \cap R} e_{(d,r,p)}(x)$$

$$e_{(d,r,p)}(x) = \max\{p(\rho(x) - \mu)(x_d - r), 0\}$$
**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

Interpretation: other methods got caught in local minima.

<table>
<thead>
<tr>
<th>Solver</th>
<th>Seed</th>
<th>Computation time (secs)</th>
<th>Falsification</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>PTC</td>
<td>Aut. Trans</td>
</tr>
<tr>
<td>Hyperplane classification + CMA-ES-Breach</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>2891</td>
<td>996</td>
<td></td>
</tr>
<tr>
<td>5000</td>
<td>2364</td>
<td>1382</td>
<td></td>
</tr>
<tr>
<td>10000</td>
<td>2101</td>
<td>1720</td>
<td></td>
</tr>
<tr>
<td>15000</td>
<td>2271</td>
<td>1355</td>
<td></td>
</tr>
<tr>
<td>CMA-ES-Breach</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5000</td>
<td>T.O. (5000)</td>
<td>1302</td>
<td></td>
</tr>
<tr>
<td>15000</td>
<td>T.O. (5000)</td>
<td>1325</td>
<td></td>
</tr>
<tr>
<td>Grid based random sampling</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S-TaLiRo (Simulated Annealing)</td>
<td>4481</td>
<td>T.O. (3000)</td>
<td></td>
</tr>
<tr>
<td>S-TaLiRo (Simulated Annealing)</td>
<td>4481</td>
<td>Default stopping (3300)</td>
<td>✓</td>
</tr>
</tbody>
</table>
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 $\epsilon$-robust counterexample, there is a grid size such that will find it)
  - coverage helps falsification by exploring more, thus avoiding local minima