Validation of Automotive Control Applications using Formal Methods and metamodeling techniques

- Simone Silvetti, Esteco Spa & University Udine
- Mariapia Marchi, Esteco Spa
MDB (Model Based Development)

❖ process aimed at designing complex systems
❖ cost reduction
❖ reduce development time
MDB (Model Based Development)
MDB (Model Based Development)
MDB (Model Based Development)
MDB (Model Based Development)
MDB (Model Based Development)
MDB (Model Based Development)
Validation Process
Validation Process

❖ Use of block diagram tools (Simulink, Gt suite)

❖ Powerful Tools but complex
Validation Process

❖ Use of block diagram tools (Simulink, Gt suite)

❖ Powerful Tools but complex

❖ Use of natural languages
❖ Involves time events...

❖ Not rigorous
❖ Not Machine interpretable
Validation Process

- Use of block diagram tools (Simulink, Gt suite)
- Powerful Tools but complex

- Use of natural languages
  - Involves time events...
- Not rigorous
  - Not Machine interpretable
Validation Process

- Use of block diagram tools (Simulink, Gt suite)

- Powerful Tools but complex

- Use of natural languages
  - Involves time events...

- Not rigorous
  - Not Machine interpretable

FORMAL METHODS!
Validation Process
Validation Process
Validation Process
Validation Process
Validation Process

“If the engine speed (w) is always less than $k_1$ then vehicle speed (v) can not exceed $k_2$ in less than $T$ sec”

$\neg (F_{[0,T]}(v \geq k_2) \land G(w \leq k_1))$
Robustness Semantics

\[ \models \varphi \]
Robustness Semantics

$\vdash \varphi$ ?

$F( f>k)$

<table>
<thead>
<tr>
<th>Boolean</th>
</tr>
</thead>
<tbody>
<tr>
<td>yes/no</td>
</tr>
</tbody>
</table>
Robustness Semantics

\[ \vdash \phi \quad ? \]

F( f>k )

<table>
<thead>
<tr>
<th>Boolean</th>
<th>Robustness</th>
</tr>
</thead>
<tbody>
<tr>
<td>yes/no</td>
<td>+30 / -30</td>
</tr>
</tbody>
</table>

More Information!
The goal

\[ f \rightarrow M \rightarrow M(f) \]
The goal

The optimization Problem

\[ R = \min_{f \in F} [M(f), \varphi] \]
The goal

The optimization Problem

\[ R = \min_{f \in F} [M(f), \varphi] \]

Counterexample

Safe!
The optimization process
The optimization process

Challenges

❖ Low number of model execution

❖ Inputs are functions (temporal series)!!
The optimization process

Challenges

- Low number of model execution
- Inputs are functions (temporal series)!!
The optimization process

**Challenges**

- Low number of model execution
- Inputs are functions (temporal series)!!

GP-UCB

Adaptive Control Point Parametrization
The Control Point Parametrization

Fix the times

interpolation
The Control Point Parametrization

Fix the times

n Control Points → n Variable to optimize
The Control Point Parametrization

Fix the times

interpolation

n Control Points -> n Variable to optimize
The **adaptive** Control Point Param.

Interpolation

n Control Points $\rightarrow 2n$ Variable to optimize
Problem

Increase the expressivity

but...

Doubled the variables
Problem

Increase the expressivity

but...

Doubled the variables

Solution

GP-UCB Optimizer
GP-UCB
GP-UCB
GP-UCB
GP-UCB

$P(x,y)$
GP-UCB

$P(x,y)$
GP-UCB

$P(x,y)$
GP-UCB

$P(x,y)$
GP-UCB

$P(x, y)$
GP-UCB

\[ P(x, y) \]
Doubled the variables
Reduce Input Space
Adaptive Idea

Input Space
Adaptive Idea
Adaptive Idea

Input Space
Adaptive Idea

Input Space
Adaptive Idea

Input Space
Adaptive Idea

Input Space
Adaptive Idea

Input Space
Adaptive Idea
Adaptive Idea

Input Space
Automatic transmission
Automatic transmission
Automatic transmission

69 blocks: 2 integrators, 3 look-up tables, 3 2D look-up tables, Stateflow Chart
## Results

### Automatic Transmission

<table>
<thead>
<tr>
<th>( \varphi )</th>
<th>Natural languages</th>
<th>MTL</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \varphi_1 )</td>
<td>The engine ((w)) and the vehicle speed ((v)) never reach (k_1) and (k_2), resp.</td>
<td>( G( (w \leq k_1) \land (v \leq k_2) ) )</td>
</tr>
<tr>
<td>( \varphi_2 )</td>
<td>If the engine speed ((w)) is always less than (k_1) then vehicle speed ((v)) can not exceed (k_2) in less then (T) sec.</td>
<td>( \neg (F_{[0,T]} (v \geq k_2) \land G(w \leq k_1)) )</td>
</tr>
<tr>
<td>( \varphi_3 )</td>
<td>Within (T) sec the vehicle speed ((v)) is above (k_2) and from that point on the engine speed ((w)) is always less then (k_1)</td>
<td>( F_{[0,T]} ((v \geq k_2) \land G(w \leq k_1)) )</td>
</tr>
<tr>
<td>( \varphi_4 )</td>
<td>A gear increase from first to fourth in under than 10 sec, ending in an engine speed ((w)) above (k_1) within 2 sec of that, should result in a vehicle speed ((v)) above (k_2).</td>
<td>( ( (g_1 \cup g_2 \cup g_3 \cup g_4) \land F_{[0,10]} (g_4 \land F_{[0,2]} (w \geq k_1)) ) ( \rightarrow G_{[0,10]} (g_4 \rightarrow X(g_4 \cup {u_{[0,1]}(v \geq k_2)}) )</td>
</tr>
</tbody>
</table>
# Results

<table>
<thead>
<tr>
<th>$\varphi$</th>
<th>$S$-TaLiro</th>
<th>$aCPP$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1,1</td>
<td>3,3</td>
</tr>
<tr>
<td>$\varphi_1$ ($k_1=4500$, $k_2=160$)</td>
<td>8.54 ± 5.72</td>
<td>10 ± 10,1</td>
</tr>
<tr>
<td>$\varphi_2$ ($k_1=4500$, $k_2=85$)</td>
<td>63.90 ± 53.20</td>
<td>124.82 ± 101.51</td>
</tr>
<tr>
<td>$\varphi_3$ ($k_1=4500$, $k_2=80$)</td>
<td>12.95 ± 7.37</td>
<td>49.8 ± 55.47</td>
</tr>
<tr>
<td>$\varphi_4$ ($k_1=4500$, $k_2=80$)</td>
<td>28.59 ± 24.15</td>
<td>32.65 ± 27.05</td>
</tr>
</tbody>
</table>

- **$aCPP$** reduces minimum number of evaluations by **50-70%**
- **GP-UCB** is slow.
Results

Time = \{\text{#Simulations}\} \times \{\text{Simulation Time}\} + \{\text{Optimizer time}\}

GP-UCB is slow
Results

\[ \text{Time} = \#\text{Simulations} \times \text{Simulation Time} + \text{Optimizer time} \]

GP-UCB is slow

Future work

- from Matlab to Java (parallelization)
- multi-objective approach
- using fmi as simulator
Acknowledges

Esteco

Luca Bortolussi

Alberto Policriti
Thank You!

....and use Formal Methods