CSE 591: Theoretical Aspects of CPS

Robust testing and Testing robustness

Instructor: Georgios E. Fainekos

School of Computing, Informatics and Decision System Engineering

Arizona State University

fainekos at asu edu

http://www.public.asu.edu/~gfaineko
Motivation - Transmission lines

At low frequencies
Motivation - Transmission lines

Operating frequencies

Feature size

Transmission lines become as important as the devices!

Distributed RLC circuit

Analysis of interconnect effects is a necessity!
Motivation - A study of transient dynamics

Desired Performance Characteristics
1. Overshoot
2. Rise time
3. Delay time
4. Settling time
5. Constraints on input/states
6. Response sensitivity

Use Linear or Metric Temporal Logic
Example: Verifying a transmission line

System:
\[ \dot{x}(t) = A x(t) + b u_{in}(t) \]
\[ u_{out}(t) = C x(t) \]

Step input \((t > 0)\):
\[ u_{in}(t) = 1 \]

Steady state at \(t = 0\):
\[ x(0) = -A^{-1} b u_{in}(0) \]

Property:
\[ \Phi = G p_1 \land F_{[0,0.85]} G p_2 \]
\[ \Theta(p_1) = [-1.5,1.5] \]
\[ \Theta(p_2) = [0.8,1.2] \]

Initial conditions:
\[ u_{in}(0) \in [-0.2,0.2] \]

Uncertain parameters
e.g. \(C \in [a_1,a_2]\)
What can we verify?

• Verifying **Cyber systems** is a **decidable problem**
  - The Turing award this year was given to the founders of model checking: E. Clark, A. Emerson and J. Sifakis

  ![Diagram]

• Applications in verification of software, hardware, protocols etc.
• Many software toolboxes: SPIN, SMV etc
What about hybrid (embedded) systems?

• In general, verifying a hybrid system is **undecidable**
What can be done?

• Previous approaches to the undecidability problem:
  • identifying decidable classes:
    ♦ Alur, Henzinger, Pappas, Lafferriere, ...
  • semi-decidable algorithms:
    ♦ Krogh, Alur, Henzinger, Dang, Ivančić, Girard, Mitchell, Tomlin, Maler, ...
  • barrier certificates:
    ♦ Prajna, Jadbabaie, ...
  • systematic simulations / model based testing:
    ♦ Krogh, Maler, Dang, Lee, Sokolsky, Kumar, Esposito, LaValle, Vardi, ...

• In Practice: **Simulations !**
Advantages:

- A finite number of simulations
- Coverage guarantees
- Scales as well as simulation scales
- More complicated specifications than safety
- Almost no parameters to set besides the simulation parameters

- New tools: we need metrics
Problem 1: Robust Temporal Logic Testing

Closed-loop system $\Sigma$

\[
\begin{align*}
\dot{x} &= f(x, p, u) \\
y &= g(x, p, u)
\end{align*}
\]

$X_0 \subseteq X$

$LTL / MTL$

\[
\Phi = G p_1 \land F_{[0,T]} G p_2
\]

Robust Tester

$L(\Sigma) \subseteq L(\Phi)$

Fainekos, Girard and Pappas, *Temporal logic verification using simulation*, FORMATS 2006


Fainekos, Pappas, *MTL Robust Testing and Verification for LPV Systems*, ACC 2009
What are the challenges in robust TL testing?

1. What is the specification language?

2. How can we define neighborhoods of signals with the same temporal properties?

3. Can two signals have approximately similar behavior?

4. How can we verify a system for an infinite set of parameters?
Overview

Motivation

Robust Testing

- Problem - Challenges
- Specification language (MTL)
  - Robust Temporal Logic Testing
  - Analog system robust testing / verification
  - Hybrid system robust testing

Testing Robustness
Metric Temporal Logic (MTL)

Syntax:

\[ \Phi ::= T \mid \bot \mid p \mid \neg p \mid \Phi_1 \lor \Phi_2 \mid \Phi_1 \land \Phi_2 \mid \Phi_1 \mathcal{U}_t \Phi_2 \mid \Phi_1 \mathcal{R}_t \Phi_2 \]

*I* can be of any bounded or unbounded interval of \( \mathbb{R}^+ \), but \( I \neq \emptyset \)

i.e. \( I = [0, +\infty) \), \( I = [2.5, 9.8] \)

Derived operators:

- **Eventually (in the future)** \( F_I \Phi := T \mathcal{U}_I \Phi \)
- **Always (globally)** \( G_I \Phi := \bot \mathcal{R}_I \Phi \)

**Koymans ’90, Specifying real-time properties with metric temporal logic**
LTL intuition

- **G a** - always a
- **F a** - eventually a
- **X a** - next state a
- **a U b** - a until b
- **a B b** - a before b
MTL: An example for signals

Boolean abstraction

$F_I a$
MTL: More complicated examples

Boolean abstraction

$F_t a$

$G_{t'} \neg a$

$(\neg a) U_t b$
Temporal Logic Testing

Truth Value \{\bot, T\}

LTL / MTL
\[ \Phi = G p_1 \land F_{[0,T]} G p_2 \]

A/D \rightarrow Boolean abstraction

Monitoring Algorithm

[Maler and Nickovic '04]
[Thati and Rosu '04]
[Rosu and Havelund '05]
[Geilen '01]
others ...
Overview

Motivation

Robust Testing

- Problem - Challenges
- Specification language (MTL)
- Robust Temporal Logic Testing
- Analog system robust testing / verification
- Hybrid system robust testing

Testing Robustness
Two signals that satisfy the same spec, but ...

MTL Spec:
\[ G(p_1YF_{\leq} p_2) \]
LTL to motion planning

F(p_2 \land F(p_3 \land F(p_4 \land \neg(p_3 \lor p_4) \cup p_1)))
Robustness of Temporal Logics

$LTL / MTL$

$\Phi = G(p_1 Y F \leq 2 p_2)$

Monitor/Tester

Robustness parameter

$\varepsilon \in \mathbb{R} \cup \{\pm \infty\}$


Fainekos and Pappas, *Robustness of temporal logic specifications for continuous-time signals*, TCS, 2009
Metric Spaces

- A **metric space** \((X, d)\) is a set \(X\) with a metric \(d\)
- A **metric** on a set \(X\) is a positive function \(d : X \times X \rightarrow \mathbb{R}^+\), such that the three following properties hold:
  - for all \(x_1, x_2, x_3 \in X\) it is \(d(x_1, x_3) \leq d(x_1, x_2) + d(x_2, x_3)\)
  - for all \(x_1, x_2 \in X\) it is \(d(x_1, x_2) = 0\) iff \(x_1 = x_2\)
  - for all \(x_1, x_2 \in X\) it is \(d(x_1, x_2) = d(x_2, x_1)\)
- Given a metric \(d\), a radius \(\varepsilon \in \mathbb{R}^+\) and a point \(x \in X\), then the **open \(\varepsilon\)-ball** centered at \(x\) is defined as
  \[
  B_d(x, \varepsilon) = \{ y \in X \mid d(x, y) < \varepsilon \}
  \]
Let $x \in X$ be a point, $C \subseteq X$ be a set and $d$ be a metric. Then we define

$dist_d(x, C) := \inf \{ d(x, y) \mid y \in cl(C) \}$

$depth_d(x, C) := dist_d(x, X \setminus C)$

$Dist_d(x, C) := \begin{cases} 
- dist_d(x, C) & \text{if } x \not\in C \\
depth_d(x, C) & \text{if } x \in C
\end{cases}$
Definition of robustness for signals

Given a signal $s$, we can define the *robustness degree* as

$$\varepsilon := \text{Dist}_\rho(s, \mathcal{L}(\Phi))$$

$$\rho(s, s') = \sup \{d(s(t), s'(t)) \mid t \in \mathbb{R}\}$$
**Main result**

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

---


Simple example
Software toolbox: TaLiRo

Available at: http://www.public.asu.edu/~gfaineko/taliro.html

Input:
- Discrete time signal
- LTL / MTL formula & Observation map

Monitor/Tester

Output:
\[ \varepsilon \in \mathbb{R} \cup \{\pm \infty\} \]
Discrete-time Robust Semantics for MTL

\[
\begin{align*}
\llbracket c \rrbracket_D(\mu, i) & := c \\
\llbracket p \rrbracket_D(\mu, i) & := \text{Dist}_d(\sigma(i), \mathcal{O}(p)) \\
\llbracket \neg \phi_1 \rrbracket_D(\mu, i) & := \neg \llbracket \phi_1 \rrbracket_D(\mu, i) \\
\llbracket \phi_1 \lor \phi_2 \rrbracket_D(\mu, i) & := \llbracket \phi_1 \rrbracket_D(\mu, i) \cup \llbracket \phi_2 \rrbracket_D(\mu, i) \\
\llbracket \phi_1 \mathcal{U} \mathcal{I} \phi_2 \rrbracket_D(\mu, i) & := \bigsqcup_{j \in \tau^{-1}(\tau(i) + R\mathcal{I})} \left( \llbracket \phi_2 \rrbracket_D(\mu, j) \cap \bigsqcap_{i \leq k < j} \llbracket \phi_1 \rrbracket_D(\mu, k) \right)
\end{align*}
\]

Based on formula re-writing, we can derive a monitoring/testing algorithm

\[
\llbracket \phi_1 \mathcal{U} \mathcal{I} \phi_2 \rrbracket_D(\mu, i) = \begin{cases} 
(K_\infty^\infty(0, \mathcal{I}) \cap \llbracket \phi_2 \rrbracket_D(\mu, i)) \cup \\
\quad \cup \left( \llbracket \phi_1 \rrbracket_D(\mu, i) \cap \llbracket \phi_1 \mathcal{U} \mathcal{I}^{-\delta_\tau(i)} \phi_2 \rrbracket_D(\mu, i + 1) \right) & \text{if } i < \max N \\
K_\infty^\infty(0, \mathcal{I}) \cap \llbracket \phi_2 \rrbracket_D(\mu, i) & \text{otherwise}
\end{cases}
\]
Main results

Theorem: Let $\Phi$ be an MTL formula and $\mu = (\sigma, \tau)$ be a TSS, then

$$-\text{dist}_\rho(\sigma, L^T_i(\phi, \mathcal{O})) \leq [\phi, \mathcal{O}]_D(\mu, i) \leq \text{depth}_\rho(\sigma, L^T_i(\phi, \mathcal{O}))$$

$$|[\phi, \mathcal{O}]_D(\mu)| \leq |\text{Dist}_\rho(\mu^{(1)}, L^{\mu^{(2)}}(\phi, \mathcal{O}))|$$

Extension to continuous time:

Fainekos and Pappas, Robust Sampling for MITL specifications, FORMATS 2007
Fainekos and Pappas, Robustness of temporal logic specifications for continuous-time signals, Theoretical Computer Science, 2009
Computing Distances

For computational reasons we allow only convex or concave sets which are usually described using intersections OR unions of halfspaces.

\[ S = \{ x \mid \bigwedge_{j \in J} (a_j \cdot x \leq b_j) \} \]

Assume \( S \) is convex if \( y \notin S \)

Solve QP:
\[
\begin{align*}
\min & \quad (x-y)^T(x-y) \\
\text{s.t.} & \quad \bigwedge_{j \in J} a_j \cdot x \leq b_j
\end{align*}
\]

\( \text{Dist}(y, S) = -\text{minvalQP} \)

else

for \( j \in J \)
\[
d_j = \frac{|b_j - a_j x|}{\| a_j \|}
\]
end for

\( \text{Dist}(y, S) = \min \{ d_j \}_{j \in J} \)
end if
Running TaLiRo on a Dell PowerEdge 1650

\[ s(i) = \sin \tau(i) + \sin 2\tau(i) \]
\[ \tau(i) = 0.2i \]

\[ O(p_1) = [1.5, +\infty) \]

\[ G(p_1 \rightarrow F_{(0,0,1.0) \neg p_1}) \]

<table>
<thead>
<tr>
<th>signal's time domain</th>
<th>number of samples</th>
<th>computation time (sec)</th>
<th>robustness</th>
</tr>
</thead>
<tbody>
<tr>
<td>[0, 21.99]</td>
<td>110</td>
<td>0.00</td>
<td>0.097603</td>
</tr>
<tr>
<td>[0, 188.49]</td>
<td>943</td>
<td>0.03</td>
<td>0.097603</td>
</tr>
<tr>
<td>[0, 6283.2]</td>
<td>31,416</td>
<td>1.05</td>
<td>0.092065</td>
</tr>
<tr>
<td>[0, 219911.48]</td>
<td>1,099,558</td>
<td>37.61</td>
<td>0.091793</td>
</tr>
</tbody>
</table>
Related Research

Robustness in temporal logics

- **WRT time:**
  - Huang, Voeten, Geilen '03, *Real-time property preservation in approximations of timed systems*
  - Henzinger, Majumdar, Prabhu '05, *Quantifying similarities between timed systems*
  - ...

- **WRT state:**
  - Huth, Kwiatkowska '97, *Quantitative analysis and model checking*
  - Lamine, Kabanza '00, *Using fuzzy TL for monitoring behavior-based mobile robots*
  - de Alfaro, Faella, Stoelinga '04, *Linear and Branching Metrics for Quantitative Transition Systems*
  - de Alfaro, Faella, Henzinger, Majumdar, Stoelinga '04, *Model Checking Discounted Temporal Properties*
  - Rizk, Batt, Fages, Soliman '08 *On a continuous degree of satisfaction of temporal logic formulae with applications to systems biology*
Overview

Motivation

Robust Testing
  - Problem - Challenges
  - Specification language (MTL)
  - Robust Temporal Logic Testing
  - Analog system robust testing / verification
  - Hybrid system robust testing

Testing Robustness
Example: a study of transient dynamics

System (dim 81):
\[
\dot{x}(t) = Ax(t) + bU_{in}(t) \\
U_{out}(t) = Cx(t)
\]

Step input \((t > 0)\):
\[
U_{in}(t) = 1
\]

Steady state at \(t = 0\):
\[
x(0) = -A^{-1}bU_{in}(0)
\]

Property:
\[
\Phi = Gp_1 \land F_{[0,T]}Gp_2 \\
\Theta(p_1) = [-1.5, 1.5] \\
\Theta(p_2) = [0.8, 1.2]
\]

Initial conditions:
\[
U_{in}(0) \in [-0.2, 0.2]
\]
Robust TL testing of analog systems

Closed-loop system $\Sigma$
\[
\begin{align*}
\dot{x} &= f(x) \\
y &= g(x)
\end{align*}
\]
\[X_0 \subseteq X\]

LTL / MTL
\[
\Phi = \mathcal{G} p_1 \land \mathcal{F}_{[0,T]} \mathcal{G} p_2
\]

Robust Tester

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

Fainekos, Girard and Pappas, *Temporal logic verification using simulation*, FORMATS 2006
Fainekos, Pappas, *MTL Robust Testing and Verification for LPV Systems*, ACC 2009
Main idea

Closed-loop system $\Sigma$:
\[
\begin{align*}
\dot{x} &= f(x) \\
y &= g(x)
\end{align*}
\]

$X_0 \subseteq X$

Specification $\Phi$

$L(\Sigma) \subseteq L(\Phi)$

$Proj_0 (P^\Phi \cap L(\Sigma))$

$\varepsilon$ robustness parameter $B_\rho(\sigma,|\varepsilon|)$
Main idea

Closed-loop system \( \Sigma \):
\[
\begin{align*}
\dot{x} &= f(x) \\
y &= g(x)
\end{align*}
\]
\( X_0 \subseteq X \)

Specification \( \Phi \)

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

\( \text{Proj}_0 (P^\Phi \cap \mathcal{L}(\Sigma)) \)

\( \varepsilon \) robustness parameter

\( B_\rho (\sigma, |\varepsilon|) \)
Main idea

Closed-loop system $\Sigma$:

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

$X_0 \subseteq X$

Specification $\Phi$

$L(\Sigma) \subseteq L(\Phi)$

$Proj_0 (P^\Phi \cap L(\Sigma))$

$\varepsilon$ robustness parameter $B_\rho(\sigma,|\varepsilon|)$
Achieving coverage I

Closed-loop system $\Sigma$:
\[
\begin{align*}
\dot{x} &= f(x) \quad X_0 \subseteq X \\
y &= g(x)
\end{align*}
\]

Specification $\Phi$

\[\mathcal{L}(\Sigma) \subseteq \mathcal{L}(\Phi)\]

$Proj_0(P^\Phi \cap \mathcal{L}(\Sigma))$

Good news!
Coverage with a finite number of simulations
Computing bisimulation functions

Quadratic Bisimulation Functions for Deterministic Linear Systems

\[ \dot{x} = Ax \]
\[ y = Cx \]

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

is a bisimulation function if

\[ M \geq C^T C \]
\[ A^T M + M A \leq 0 \]

Approximate Bisimulations for Constrained Linear Systems
Antoine Girard and George J. Pappas

Bisimulation Functions using Sum Of Squares Relaxation

\[ \dot{x} = f(x) \]
\[ y = g(x) \]

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

is a bisimulation function if

\[ q(x_1, x_2) - \|g_1(x_1) - g_2(x_2)\|^2 \]

is SOS

\[ \left( - \frac{\partial q(x_1, x_2)}{\partial x_1} f_1(x_1) - \frac{\partial q(x_1, x_2)}{\partial x_2} f_2(x_2) \right) \]

is SOS

Approximate Bisimulations for Nonlinear Dynamical Systems
Antoine Girard and George J. Pappas
Achieving coverage II

Closed-loop system $\Sigma$: 

\[
\begin{align*}
\dot{x} &= f(x) \\
y &= g(x)
\end{align*}
\]

$X_0 \subseteq X$

Specification $\Phi$

\[
\mathcal{L}(\Sigma) \subseteq \mathcal{L}(\Phi)
\]

Even better news! It is possible to verify the system with just one simulation.
Quick falsification

Closed-loop system $\Sigma$:
\[
\begin{align*}
\dot{x} &= f(x) \\
y &= g(x)
\end{align*}
\]
$X_0 \subseteq X$

Specification $\Phi$

\[
\mathcal{L}(\Sigma) \subseteq \mathcal{L}(\Phi)
\]

Observation!
A robust system with respect to the property requires less simulations

\[
Proj_0 (P\Phi \cap \mathcal{L}(\Sigma))
\]

\[
X_0 \subseteq X
\]
Coverage certificates

Closed-loop system $\Sigma$:

\[
\begin{align*}
\dot{x} &= f(x) \\
y &= g(x)
\end{align*}
\]

$X_0 \subseteq X$

Specification $\Phi$

\[
L(\Sigma) \subseteq L(\Phi)
\]

Good news! We get coverage guarantees after $K$ iterations.
Main results

**Theorem:** Let $V$ be a bisimulation function, for $i = 1, 2$, let $(x_i, y_i)$ be trajectories of $\Sigma$ and $\varepsilon_i$ be the robustness parameter of $\Phi$ wrt $y_i$, then

$$\exists i \in \{1, 2\}. V(x_1(0), x_2(0)) < \varepsilon_i \text{ implies } y_1 \models \Phi \text{ iff } y_2 \models \Phi$$

**Proposition:** Let $V$ be a bisimulation function. For any compact set of initial conditions $X_0 \subseteq \mathbb{R}$, for all $\delta > 0$, there exists a finite set of points $\{x_1, ..., x_r\} \subseteq X_0$ such that

for all $x \in X_0$, there exists $x_i$, such that $V(x, x_i) \leq \delta$

**Theorem:** Let $(x_1, y_1), ..., (x_r, y_r)$ be trajectories of $\Sigma$ such that $\text{Disc}(X_0, \delta) = \{x_1(0), ..., x_r(0)\}$. Let $\varepsilon_i$ be the robustness parameter of $\Phi$ wrt $y_i$. Then,

$$\forall i \in \{1, ..., r\} \cdot \varepsilon_i > \delta \implies \forall y \in \mathcal{L}(\Sigma) \cdot y \not\models \Phi$$
Experimental Results
(MATLAB toolbox)

Property:

\[ \Phi = \mathcal{G} p_1 \wedge \mathcal{F}_{[0,T]} \mathcal{G} p_2 \]

\[ \mathcal{O}(p_1) = [-\theta, \theta], \quad \mathcal{O}(p_2) = [0.8, 1.2] \]

<table>
<thead>
<tr>
<th></th>
<th>T=0.8</th>
<th>T=1.2</th>
<th>T=1.6</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \theta = 1.4 )</td>
<td>False</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>( \theta = 1.5 )</td>
<td>False</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>( \theta = 1.6 )</td>
<td>False</td>
<td>True</td>
<td>True</td>
</tr>
</tbody>
</table>

from Zhi Han's PhD Thesis 2005
Experimental Results (MATLAB toolbox)

Property:

\[ \Phi = G_{p_1} \land F_{[0,T]} G_{p_2} \]

\[ \Theta(p_1) = [-\theta,\theta], \quad \Theta(p_2) = [0.8,1.2] \]

T=0.8 | T=1.2 | T=1.6
---|---|---
\( \theta=1.4 \) | False | False | False
\( \theta=1.5 \) | False | False | False
\( \theta=1.6 \) | False | True | True

-0.1526

from Zhi Han's PhD Thesis 2005
Experimental Results

(MATLAB toolbox)

Property:

\[ \Phi = G p_1 \land F_{[0,T]} G p_2 \]

\[ \Theta(p_1) = [-\theta, \theta], \Theta(p_2) = [0.8, 1.2] \]

<table>
<thead>
<tr>
<th>T</th>
<th>T=0.8</th>
<th>T=1.2</th>
<th>T=1.6</th>
</tr>
</thead>
<tbody>
<tr>
<td>θ=1.4</td>
<td>False</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>θ=1.5</td>
<td>False</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>θ=1.6</td>
<td>False</td>
<td>True</td>
<td>True</td>
</tr>
</tbody>
</table>

from Zhi Han's PhD Thesis 2005
Experimental Results
(MATLAB toolbox)

Property:

$$\Phi = G_{p_1} \land F_{[0,T]} G_{p_2}$$

$$\mathcal{O}(p_1) = [-\theta, \theta], \mathcal{O}(p_2) = [0.8, 1.2]$$

<table>
<thead>
<tr>
<th>T=0.8</th>
<th>T=1.2</th>
<th>T=1.6</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\theta=1.4)</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>(\theta=1.5)</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>(\theta=1.6)</td>
<td>False</td>
<td>True</td>
</tr>
</tbody>
</table>

from Zhi Han’s PhD Thesis 2005
Experimental Results
(MATLAB toolbox)

Property:

\[ \Phi = G_{p_1} \land F_{[0,T]} G_{p_2} \]

\[ \Theta(p_1) = [-\theta,\theta], \Theta(p_2) = [0.8,1.2] \]

<table>
<thead>
<tr>
<th>( \theta )</th>
<th>( T=0.8 )</th>
<th>( T=1.2 )</th>
<th>( T=1.6 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.4</td>
<td>False</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>1.5</td>
<td>False</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>1.6</td>
<td>False</td>
<td>True</td>
<td>True</td>
</tr>
</tbody>
</table>

from Zhi Han's PhD Thesis 2005
Experimental Results

(MATLAB toolbox)

Property:

\[ \Phi = G_{p_1} \land F_{[0,T]} G_{p_2} \]

\( \Theta(p_i) = [\theta_{i-1}, \theta_i] \)

\( \Theta(p_i) = [0.8, 1.2] \)

from Zhi Han's PhD Thesis 2005
Extension to systems with uncertain parameters

Closed-loop system $\Sigma$
\[
\begin{align*}
\frac{dx}{dt} &= A(p(t)) \ x(t) & X_0 \subseteq X \\
y(t) &= C \ x(t) & p(t) \in P
\end{align*}
\]

$LTL / MTL$
\[
\Phi = G \pi_1 \land F_{[0,T]} G \pi_2
\]
Observation map $O$

$L(\Sigma) \subseteq L(\Phi,O)$

$\delta$-approximately bisimilar

Closed-loop system $\Sigma'$
\[
\begin{align*}
\frac{dx}{dt} &= A' \ x(t) \\
y(t) &= C \ x(t)
\end{align*}
\]

$LTL / MTL$
\[
\Phi = G \pi_1 \land F_{[0,T]} G \pi_2
\]
Observation map $O_{\delta}$

$L(\Sigma') \subseteq L(\Phi,O_{\delta})$ ?
Theorem: Let $\Sigma$ be an LPV system and $p' \in P$. If there exists a positive semidefinite matrix $M$ such that

$$M \geq \begin{bmatrix} C^T \xi C & -C^T C \\ -C^T C & C^T \xi C \end{bmatrix}$$

$$\forall p \in EP(P). \begin{bmatrix} A^T(p) & 0 \\ 0 & A^T(p') \end{bmatrix} M + M \begin{bmatrix} A(p) & 0 \\ 0 & A(p') \end{bmatrix} \leq 0$$

Then $F(x_0, x'_0) = \sqrt{\begin{bmatrix} x_0^T & x_0'^T \end{bmatrix} M \begin{bmatrix} x_0^T \\ x_0'^T \end{bmatrix}^T}$ is a bisimulation function between $\Sigma$ and $\Sigma'$. 
Approximating an LPV with an LTI System

**Proposition:** Let \( \Sigma \) be an LPV system, \( \rho' \in \mathcal{P} \) and let

\[
F(x, x') = \sqrt{V(x, x')} = \sqrt{\left[x^T \ x'^T \right] M \left[x^T \ x'^T \right]^T}
\]

be a bisimulation function between \( \Sigma \) and \( \Sigma(\rho') \). Then, the solution of the static games

\[
\max \left( \sup_{x \in E P(X_0)} \inf_{x' \in X_0} F(x, x'), \sup_{x' \in E P(X_0)} \inf_{x \in X_0} F(x, x') \right)
\]

computes the optimal points \( x \) and \( x' \) which provide an upper bound \( \delta = \mathcal{R}(x, x') \) for the approximate bisimulation relation.
Proposition:
Consider an MTL formula $\varphi$ and a map $O : \Pi \to P(X)$. If systems $\Sigma$ and $\Sigma'$ are $\delta$-approximately bisimilar and $B_\delta(L(\Sigma')) \subseteq L(\Phi, O)$, then $L(\Sigma) \subseteq L(\Phi, O)$.

Corollary:
Consider an MTL formula $\varphi$ and a map $O : \Pi \to P(X)$. If systems $\Sigma$ and $\Sigma'$ are $\delta$-approximately bisimilar and $L(\Sigma') \subseteq L(\Phi, O_\delta)$, then $L(\Sigma) \subseteq L(\Phi, O)$. 
Related Research

- TL Verification/Robust Testing of Analog Systems
  - Ghosh, Vemuri '99, Formal Verification of Synthesized Analog Designs
  - Hartong, Hedrich, Barke '02, On Discrete Modeling and Model Checking for Nonlinear Analog Systems
  - Gupta, Krogh, Rutenbar '04, Towards Formal Verification of Analog Designs
  - Frehse, Krogh, Rutenbar, Maler '05, Time Domain Verification of Oscillator Circuit Properties
  - Donze, Maler '07, Systematic Simulation using Sensitivity Analysis
  - Rizk, Batt, Fages, Soliman '08, On a continuous degree of satisfaction of temporal logic formulae with applications to systems biology
  - Julius and Pappas '09, Trajectory based verification using local finite-time invariance
  - ...
Overview

Motivation

Robust Testing

- Problem - Challenges
- Specification language (MTL)
- Robust Temporal Logic Testing
- Analog system robust testing / verification
- Hybrid system robust testing

Testing Robustness
Navigation benchmark


Hybrid system with 3x3 locations.

The dynamics:

\[
x = [x_1 \ x_2 \ v_1 \ v_2]^T, \quad \dot{x} = Ax - Bu(i, j)
\]

\[
A = \begin{bmatrix}
0 & 0 & 1 & 0 \\
0 & 0 & -1.2 & 1.1 \\
0 & 0 & 0.1 & -1.2
\end{bmatrix}
\quad \text{and} \quad
B = \begin{bmatrix}
0 & 0 \\
0 & 0 \\
-1.2 & 0.1 \\
0.1 & -1.2
\end{bmatrix}
\]

\[
u(i, j) = [\sin(\pi C(i, j)/4) \ \cos(\pi C(i, j)/4)]^T
\]

\[
C_1 = \begin{bmatrix}
U & 2 & 4 \\
4 & 3 & 4 \\
2 & 2 & U
\end{bmatrix} \quad
C_2 = \begin{bmatrix}
2 & 3 & 6 \\
3 & 3 & G \\
2 & 2 & U
\end{bmatrix} \quad
C_3 = \begin{bmatrix}
U & 2 & 4 \\
2 & 2 & 4 \\
1 & 1 & G
\end{bmatrix}
\]
Navigation benchmark
Robust testing of hybrid systems

Hybrid automaton
\[ H = (X, Q, F, H_0, \rightarrow, R, I, G) \]

Safety Specification
\[ H_U \subseteq H \]

Simulator / Tester

\[ \text{Reach}(H) \cap H_U = \emptyset \]

Julius, Fainekos, Anand, Lee, Pappas,
Robust Test Generation and Coverage for Hybrid Systems, HSCC 2007
How robust is a hybrid test trajectory?
How robust is a hybrid test trajectory?
How robust is a hybrid test trajectory?
How robust is a hybrid test trajectory?
How robust is a hybrid test trajectory?
Increasing robustness

\[ g_1^{\text{act}} \quad g_2^{\text{act}} \]

\[ g_1 \]

\[ \xi(t, x_0) \]

\[ \xi(\tau, x_0) \]

Unsafe

Unsafe

\[ x_0 \]
Timing guarantees

\[ \xi(\tau + \varepsilon, x_0) \]
\[ \xi(\tau - \varepsilon, x_0) \]
\[ B_\phi(x_0, d_{\text{min}}) \]
Computing distances

linear projections
(least squares when we consider the location dynamics)

quadratic programming

semidefinite programming

Unsafe
Overview of algorithm

1. **Hybrid Automaton**
   - Includes initial conditions

2. **Input Parameters**
   - Includes max simulation time, max number of tests, etc.

3. **Pick a point in the parameter space**

4. **Simulate trajectory**

5. **Safe?**
   - Yes: **System Unsafe**
   - No: **Dangerous**

6. **System Unsafe**
   - **Update parameter space**
     - Remove computed ellipsoid from initial conditions
     - Includes the computation of bisimulation functions

7. **Compute Robustness**

8. **Output Results**
   - Yes
   - No: **Stopping criterion?**
$$\mathcal{X}_0 = [1, 2] \times [1, 2] \times \{-0.2\} \times \{0\}$$
Navigation benchmark 1

With 25 runs, we cover >48% of the initial set.
Notice that there is a clear divide in the initial set, due to different transitions.
Benchmark problem 2

Initial condition:
\[ [2.2, 2.8] \times [1.2, 1.8] \times \{-0.2\} \times \{0\} \]

Verified to be safe with CHARON
Benchmark problem 2

Safety verified after 9 tests!
(All traces have the same qualitative behavior and the system is robust wrt to the unsafe set. Termination guaranteed similar to Girard & Pappas HSCC’06, Fainekos et al FORMATS 2006)

*Numerically, we compute a coverage estimate of 72%.*
Navigation benchmark 3

\[ X_0 = [0, 1] \times [2, 3] \times [-1, 1] \times [-1, 1] \]
Navigation benchmark 3

- Test generation using Voronoi with weights
- We proved unsafety with 10 tests.
Related Research

- Robust Testing of hybrid systems
  - Girard and Pappas '06, *Verification using simulation*
  - Dang, Donze, Maler, Shalev, '08, *Sensitive state-space exploration*
  - Lerda, Kapinski, Clarke, and Krogh, '08, *Verification of supervisory control software using state proximity and merging*
Overview

Motivation

Robust Testing

- Problem - Challenges
- Specification language (MTL)
- Robust Temporal Logic Testing
- Analog system robust testing / verification
- Hybrid system robust testing

Testing Robustness
Problem 2: Testing Temporal Logic Robustness

Closed-loop system $\Sigma$

$\dot{x} = f(x, p, u) \quad x_0 \subseteq X$

$y = g(x, p, u)$

LTL / MTL

$\Phi = G p_1 \land F_{[0,T]} G p_2$

Tester

$\inf \{ \text{Dist}_\rho (\sigma, \mathcal{L}(\Phi)) \mid \sigma \in \Sigma \} = ?$
• We need to solve an optimization problem:
  \[ \min \text{Dist}_\rho (y, \mathcal{L}(\Phi)) \]
  \[ y = g(x, u, p) \text{ and } x \text{ is a solution of } \frac{dx}{dt} = f(x, u, p) \text{ with } x_0 \in X_0, \ u \in U, \ p \in P \]

• Challenges:
  • Non-linear system dynamics
  • Unknown input signals
  • Unknown system parameters
  • Non-smooth cost function
    • not known in closed form

• Classical solution to difficult engineering problems:
  • Stochastic optimization algorithms

• Goal: Falsify the system or find non-robust behaviors
Monte Carlo Sampling (Simulated Annealing)

We adapt $\beta$ so that the acceptance / rejection ratio remains close to 1.
Example 1

System:
\[
\frac{dx}{dt} = x - y + 0.1t \\
\frac{dy}{dt} = y\cos(2\pi y) - x\sin(2\pi x) + 0.1t
\]

Initial conditions:
\([-1,1] \times [-1,1]\)

Specification:
\[ G_{[0,2]} \neg a \]
where \( O(a) = [-1.6, -1.4] \times [-1.1, -0.9] \)
Example 2: Aircraft model

\[
\dot{x} = \begin{bmatrix}
  - \frac{S \rho B_0}{2m} x_1^2 - g \sin(x_2) \\
  \frac{S \rho C_0}{2m} x_1 - g \frac{\cos(x_2)}{x_1} \\
  - \frac{S \rho C_1}{2m} \frac{x_2^2}{x_1} (B_1 u_1 + B_2 u_2) \\
  \end{bmatrix} + \begin{bmatrix}
  \frac{u_1}{m} \\
  0 \\
  0 \\
  \end{bmatrix} + \begin{bmatrix}
  - \frac{S \rho C_1}{2m} \frac{x_2^2}{x_1} \left( B_1 u_1 + B_2 u_2 u_2 \right) \\
  \frac{S \rho C_1}{2m} \frac{x_2^2}{x_1} u_2 \\
  0 \\
  \end{bmatrix}
\]

\[X_0 = [200, 260] \times [-10, 10] \times [120, 150]\]
\[U = [34386, 53973] \times [0, 16]\]

\[\varphi = [1, 1.5] (x_1 \in [250, 260]) \Rightarrow [3, 4] (x_1 \notin [230, 240])\]
Example 3: Navigation Benchmark

Specification:

\((\neg \text{Cyan}) \cup_{[0,25]} \text{(White)}\)

Not falsified, but robustness close to 0!
### Experimental Results

<table>
<thead>
<tr>
<th>$\psi$</th>
<th>#Run</th>
<th>#Iter. per run</th>
<th>#Fals.</th>
<th>MinRob. $\langle\text{min, avg, max}\rangle$</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>MC</td>
<td>UR</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>avg</td>
<td>avg</td>
</tr>
<tr>
<td>Aircraft Example 3.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\square_{[.5,1.5]} a \land \diamond_{[3,4]} b$</td>
<td>100</td>
<td>500</td>
<td>88</td>
<td>100</td>
<td>$\langle 0, 1.2, 18.6 \rangle$</td>
</tr>
<tr>
<td>$\square_{[0,4]} c \land \diamond_{[3.5,4]} d$</td>
<td>100</td>
<td>1000</td>
<td>100</td>
<td>66</td>
<td>$\langle 0, 0, 0 \rangle$</td>
</tr>
<tr>
<td>$\diamond_{[1,3]} e$</td>
<td>100</td>
<td>2000</td>
<td>81</td>
<td>16</td>
<td>$\langle 0, .9, 40 \rangle$</td>
</tr>
<tr>
<td>$\diamond_{[.5,1]} f \land \square_{[3,4]} g$</td>
<td>100</td>
<td>2500</td>
<td>0</td>
<td>0</td>
<td>$\langle 9.5, 9.7, 10.1 \rangle$</td>
</tr>
<tr>
<td>$\square_{[0,.5]} h$</td>
<td>100</td>
<td>2500</td>
<td>100</td>
<td>100</td>
<td>$\langle 0, 0, 0 \rangle$</td>
</tr>
<tr>
<td>$\square_{[2,2.5]} i$</td>
<td>100</td>
<td>2500</td>
<td>99</td>
<td>51</td>
<td>$\langle 0, 0, 1.0 \rangle$</td>
</tr>
<tr>
<td>Vehicle Benchmark 4.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$(\neg b) U_{[0,25.0]} c$</td>
<td>35</td>
<td>1000</td>
<td>11</td>
<td>8</td>
<td>$\langle 0, .02, .04 \rangle$</td>
</tr>
</tbody>
</table>
Related Research

- **Probabilistic/Stochastic testing**
  - Kapinski, Krogh, Maler, Stursberg ’03, *On systematic simulation of open continuous systems*
  - Tan, Kim, Sokolsky, Lee ’04, *Model-based testing and monitoring for hybrid embedded systems*
  - Bhatia, Frazzoli, ’04, Incremental search methods for reachability analysis of continuous and hybrid systems.
  - Bensalem, Bozga, Krichen, Tripakis ’05, *Testing conformance of real time applications with automatic generation of observers*
  - Esposito, Kim, Kumar ’05, *Adaptive RRTs for validating hybrid robotic control systems*
  - Branicky, Curtiss, Levine, Morgan, ’06, *Sampling-based planning, control and verification of hybrid systems.*
  - Younes, Simmons,’06. *Statistical probabilistic model checking with a focus on time-bounded properties*
  - Plaku, Kavraki, Vardi ’07, *Hybrid systems: From verification to falsification*
  - Nahhal, Dang ’07, *Test Coverage for Continuous and Hybrid Systems*
  - Plaku, Kavraki, Vardi, ’09, *Falsification of ltl safety properties in hybrid systems*
  - Clarke, Donze, Legay, ’09 *Statistical model checking of analog mixed-signal circuits with an application to a third order δ-σ modulator*
  - ...