Verification from Simulations and Modular Annotations

Zhenqi Huang, Yu Wang, Sayan Mitra and Geir Dullerud
Coordinated Science Lab
University of Illinois at Urbana-Champaign
Composed Safety-critical CPS
Safety under Adversary

\[ \dot{u}_c = \text{ctr}(u_c, x) \]

\[ \dot{x} = f(x, u_c, u_a) \]

\[ u_a = \text{adv}(x) \]
Invariant Verification

Computing reach set exactly is undecidable [Henzinger]
  • Over-approximations
  • Bounded time

• Static analysis and symbolic approaches
  • E.g. HyTech[Henzinger97], CheckMate[Silva00], d/dt[Dang98],
    SpaceEx[Frehse11], flow*[Chen13]

• Dynamic+Static analysis using numerical simulations
  • E.g. Breach[Donzé10], S-TaLiRo[Annapureddy11], C2E2[Duggirala13]
Simulation-Based Bounded Reachability

\[ \dot{x} = f(x), \Theta \subseteq \mathbb{R}^n \]

- Finite cover of \( \Theta \)
- Simulate from the center of each cover
- Bloat the simulation with some factor, such that the bloated tube contains all trajectories starting from the cover
- Union of all tubes gives an over-approximation of reach set

The bloating factor can be computed using sensitivity analysis[Donzé07], or given as an annotation for the model[Duggirala13,Huang14].
We assume the network is annotated by the user per automaton per mode.
Definition [Duggirala13, Huang14]. IS discrepancy is defined by $\beta$ and $\gamma$ such that for any initial states $\theta, \theta'$ and any inputs $u, u'$,

$$|x(t) - x'(t)| \leq \beta(|\theta - \theta'|, t) + \int_{0}^{t} \gamma(|u(s) - u'(s)|) ds$$

- $\beta \to 0$ as $\theta \to \theta'$, and $\gamma \to 0$ as $u \to u'$
- Linear $f()$: found automatically
- Nonlinear $f()$: several heuristics were proposed
Bloating a Trajectory with IS Discrepancy

\[ \dot{m}_1 = \beta_1(\delta, t) + \gamma_1(m_2, m_3) \]

\[ \dot{m}_2 = \beta_2(\delta, t) + \gamma_2(m_1, m_3) \]

\[ \dot{m}_3 = \beta_3(\delta, t) + \gamma_3(m_1, m_2) \]

• The bloated tube contains all trajectories start from the \( \delta \)-ball of \( \theta \).
• The over-approximation can be computed arbitrarily precise.
Simulation & Modular Annotation $\implies$ Proof

\[
\dot{x}_1 = f_1(x_1, u_1) \\
\dot{x}_2 = f_2(x_2, u_2) \\
\dot{x}_3 = f_3(x_3, u_3)
\]

Reduced Model

Simulation Engine

Reach set over-approximation

Trajectory

Bloating factor

Refinement

Sat Inv?

Counter Example

Proof

HSCC 2014, Berlin
Soundness and Relative Completeness

• **Definition.** $c$-perturb($A$) is the set of all HA $A'$, such that $A'$ and $A$ are identical except that
  • The initial sets: $d_H(\Theta_A, \Theta_{A'}) \leq c$, and
  • The differential equations in every module: $d_\infty(f_A, f_{A'}) \leq c$

• **Definition.** $A$ Robustly satisfies (violates) $Inv$ iff there exists $c > 0$ such that all $c$-perturb($A$) satisfy (violate) $Inv$.

• **Theorem:** the algorithm is sound and relatively complete.
  • i.e. the algorithm terminates if $A$ robustly satisfies (violates) $Inv$. 
## Experiments

<table>
<thead>
<tr>
<th>Network</th>
<th># Variables</th>
<th># Modes</th>
<th># Sims</th>
<th>Run Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 cells (FH)</td>
<td>16</td>
<td>1</td>
<td>24</td>
<td>33</td>
</tr>
<tr>
<td>Lin. Sync</td>
<td>24</td>
<td>6</td>
<td>128</td>
<td>135.1</td>
</tr>
<tr>
<td>Nonli. WT</td>
<td>30</td>
<td>6</td>
<td>128</td>
<td>140.0</td>
</tr>
<tr>
<td>5 cells</td>
<td>20</td>
<td>$2.1 \times 10^7$</td>
<td>170</td>
<td>945</td>
</tr>
<tr>
<td>8 cells</td>
<td>32</td>
<td>$5.0 \times 10^{10}$</td>
<td>73</td>
<td>2377</td>
</tr>
</tbody>
</table>
Discussion

• A scalable technique to verify nonlinear hybrid automata networks using annotations
  • IS discrepancies are used to construct a reduced model of the overall network whose trajectory gives the bloating factor
  • Both original network and the reduced model
  • Sound and relatively complete algorithm

• Cardiac cell networks upto 8 cells, 32 var. and $29^8$ modes are verified using 29 annotations
Ongoing: Synthesis

Controller
\[ u_c = ctr(u_c, x) \]

Vehicle
\[ \dot{x} = f(x, u_c, u_a) \]

Adversary
\[ u_a = adv(x) \]

Candidate Controllers
\[ \{ \hat{u}_t \} \]

Synthesizer

Verifier

Synthesizer

Verifier

Synthesis Fails

Synthesis Success