Algorithmic Verification of Stability of Hybrid Systems

Pavithra Prabhakar
Kansas State University

Mysore Park Workshop

Joint work with Miriam Garcia Soto
(IMDEA Software Institute, Madrid)
Cyber-Physical Systems (CPS)

Systems in which software "cyber" interacts with the "physical" world

- Medical Devices
- Automotive
- Robotics
- Aeronautics
- Process control

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

Hybrid Systems

Systems with mixed discrete-continuous behaviors
Hybrid Systems
Air traffic collision avoidance protocol

\( x = (x_1, x_2) \): position of the airplane
\( d = (d_1, d_2) \): velocity of the airplane

\[
\begin{bmatrix}
\dot{x}_1 \\
\dot{x}_2 \\
\dot{d}_1 \\
\dot{d}_2
\end{bmatrix} =
\begin{bmatrix}
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 \\
0 & 0 & 0 & -\omega \\
0 & 0 & \omega & 0
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
d_1 \\
d_2
\end{bmatrix}
\]

\( \omega \): the angular velocity

\[\|x - y\| \leq p \quad c = x + \lambda d = y + \lambda e \]
\[\|x - c\| = \sqrt{3}r \quad (r\omega)^2 = \|d\|^2 \quad x^0 := x, \; d^0 := d\]

\( \omega := \ast \)

minimum separation

The aircraft maintain a minimum distance between them always

\[\|x - c\| \leq r \quad \omega := -\omega\]

\[\omega := 0 \quad x + \lambda_2 d = x^0 + \lambda_1 d^0\]
Velocity $v$ reaches $v_{ref}$

even in the presence of disturbances
Stability
Stability is a fundamental property in control system design

- It captures the notion that small perturbations in the initial state or input result in only small deviations from the nominal behavior

Cruise control

Robotic arm

Bipedal robot walking

- Set-point stability
- Stability of the periodic orbit
Small perturbations in the initial state lead to small deviations in the system behavior.
Lyapunov and asymptotic stability

Lyapunov Stability

A system is Lyapunov stable with respect to a trajectory $\tau$ if

$$\forall \varepsilon > 0, \exists \delta > 0, \forall \tau'$$

$$|\tau(0) - \tau'(0)| < \delta \Rightarrow \forall t \geq 0 \quad |\tau(t) - \tau'(t)| < \varepsilon$$

Asymptotic Stability

Asymptotic stability in addition requires convergence to the reference trajectory.
Challenges in Stability Verification for Hybrid Systems
Stability analysis

Linear dynamical systems

Stability can be determined by eigen values analysis

Stable

Linear hybrid systems

Eigen value analysis does not suffice for switched linear system

Stable

Unstable
Current techniques for Stability Verification
Lyapunov’s second method

Lyapunov function:

- Continuously differentiable
  \[ V : \mathbb{R}^n \to \mathbb{R}^+ \]
- Positive definite
  \[ V(x) \geq 0 \quad \forall x \]
- Decreases along any trajectory
  \[ \frac{\partial V(x)}{\partial x} F(x) \leq 0 \quad \forall x \]

Template based automated search

- Choose a template
- Polynomial with coefficients as parameters
- Encode (a relaxation) of the constraints as a sum-of-square programming problem
- Use existing tools for SOS

Shortcomings:

- Success depends crucially on the choice of the template
- The current methods provide no insight into the reason for the failure, when a template fails to prove stability
- No guidance regarding the choice of the next template

A CEGAR framework
Counter-example guided abstraction refinement
Abstraction

Safety Analysis

* Every trajectory corresponds to a path in the graph
* Absence of a path from green to red node implies safety
Abstraction

Safety Analysis

* Every trajectory corresponds to a path in the graph
* Absence of a path from green to red node implies safety

* The above system is safe
* The abstract graph has a counter-example
* Right abstractions are hard to find!
Refinement

Safety Analysis

- Every trajectory corresponds to a path in the graph
- Absence of a path from green to red node implies safety

- The above system is safe
- The abstract graph has a counter-example
- Right abstractions are hard to find!

- Refine by analyzing the abstract counter-example
Counter-example guided abstraction refinement

Concrete System → Abstract System → Model-Check → Property

- Abstraction Relation
- Systematically iterate over the abstract systems
- Returns a counter-example in the case that the abstraction fails
- The counter-example can be used to guide the choice of the next abstraction

Abstract System → Model-Check

- No → Abstract Counter-example
- Yes → Property violated

Refine

- No Analysis Results
- Yes Validate

Template based search

- Success depends crucially on the choice of the template
- No insight into the reason for the failure, when a template fails to prove stability
- No guidance regarding the choice of the next template

CEGAR for discrete systems [Kurshan et al. 93, Clarke et al. 00, Ball et al. 02]

CEGAR for hybrid systems safety verification [Alur et al 03, Clarke et al 03, Prabhakar et al 13]

CEGAR framework
What are the ingredients for CEGAR?
CEGAR questions

- What pre-orders preserve stability?
- How do we construct abstractions/refinement?
Simulations and Bisimulations

Simulation between $\mathcal{T}_1$ and $\mathcal{T}_2$ is a binary relation $R \subseteq S_1 \times S_2$

- Every path of the first system has a matching path in the second system
- Bisimulations preserve several discrete-time properties [Timed automata, Multi-rate automata, O-minimal automata]
Stability is not bisimulation invariant!

Uniformly continuous (bi)-simulations

\begin{align*}
R & \text{ is a uniformly continuous simulation from } \mathcal{T}_1 \text{ to } \mathcal{T}_2 \text{ if} \\
& \quad \text{1. } R \text{ is a simulation and} \\
& \quad \text{2. } R \text{ is uniformly continuous.} \\
& \quad \forall \epsilon > 0, \exists \delta > 0 \text{ such that } \forall x \in \text{Dom}(R), \\
& \quad R(B_\delta(x)) \subseteq B_\epsilon(R(x))
\end{align*}

\textbf{Theorem}

Let \( R \) be a uniformly continuous simulation from \( \mathcal{T}_1 \) to \( \mathcal{T}_2 \), and be consistent with \( \tau_1 \) and \( \tau_2 \). \( \mathcal{T}_2 \) is stable with respect to \( \tau_2 \) implies \( \mathcal{T}_1 \) is stable with respect to \( \tau_1 \)

\begin{itemize}
  \item Continuous simulations suffice for stability with respect to an equilibrium point
  \item Classical stability analysis techniques —— Lyapunov’s second method and Linearization —— are instances of stability analysis based on uniformly continuous simulations
\end{itemize}
Abstraction based Analysis

- What pre-orders preserve stability?
- How do we construct abstractions?
\[ \exists \gamma > 0, \forall \epsilon \in (0, \gamma] \]
\[ \forall \epsilon > 0, \exists \delta > 0, [(\tau(0) \in B_\delta(0)) \Rightarrow \forall t(\tau(t) \in B_\epsilon(0))] \]

- Special structure in a small neighborhood
- Homogenous linear constraints matter
PCD examples

Lyapunov stable but Not asymptotically stable

Both Lyapunov stable and asymptotically stable

Unstable

**Theorem**

Verifying Lyapunov / Asymptotic Stability is undecidable in 5 dimensions for PCDs, but is decidable in 2 dimension for a more general class of systems.
Weights capture information about distance to the origin along the executions.
Weighted Graph Construction
A remark on weight computation

\[ w(e) = \frac{|d_2|}{|d_1|} \]

\[ \sup \frac{|v_2|}{|v_1|} \]

\[ t \geq 0, v_1 \in f_1, v_2 \in f_2, v_2 = v_1 + \varphi t \]

\[ \vec{a} \rightarrow \vec{b} \text{ implies } \alpha \vec{a} \rightarrow \alpha \vec{b} \]
Theorem

The piecewise constant derivative system is Lyapunov stable if

- there are no edges with infinite weights and
- the weighted graph does not contain any cycles with product of weights on the edges greater than 1

Abstraction based model-checking of stability of hybrid systems. P. Prabhakar, M. G. Soto. CAV’13
Quantitative Predicate Abstraction

Let \( \mathcal{H} \) be a hybrid system.

Let \( \mathcal{P} = \{P_1, \ldots, P_k\} \) a finite partition of its state-space

Construct a weighted graph \( \mathcal{G} = (V, E, W) \), where:

- \( V = \mathcal{P} \)
- \( (P_1, P_2) \in E \) if there exists \( P \) such that \( \text{Reach}(P_1, P, P_2) \neq \emptyset \)
- \( W(e) = \sup\{ \|y\| \mid (x, y) \in \text{Reach}(P_1, P, P_2) \} \), where \( e = (P_1, P_2) \)

\[
\text{Reach}(P_1, P, P_2) = \{ (s_1, s_2) \mid s_1 \in P_1, s_2 \in P_2, s_1 \xrightarrow{P} s_2 \}
\]

Soundness holds under certain finite variability conditions on the dynamics with respect to the partition

*Foundations for Quantative predicate abstraction for stability analysis of hybrid systems.* P. Prabhakar, M. G. Soto. **VMCAI’15**
Rectangular and polyhedral dynamics

$$\text{Reach}(P_1, P, P_2) = \{(s_1, s_2) \mid s_1 \in P_1, s_2 \in P_2, s_1 \overset{P}{\sim} s_2\}$$

Constant derivative $\dot{x} = \varphi$

$$\sup \frac{|v_2|}{|v_1|}$$

$$t \geq 0, v_1 \in f_1, v_2 \in f_2, v_2 = v_1 + \varphi t \quad \varphi \in P$$

Polyhedral dynamics $\dot{x} \in P$, $P$ is a polyhedral set
Polyhedral switched systems

• Overlapping guards and invariants

\[
\begin{align*}
\dot{x} &= -1 \\
y &= 1
\end{align*}
\]
\begin{align*}
q_1 & \quad x \geq 0 \\
q_2 & \quad x - y < 0
\end{align*}
\begin{align*}
\dot{x} &= 1 \\
y &= -2
\end{align*}
\begin{align*}
q_3 & \quad y \geq 0 \\
q_4 & \quad x \leq 0
\end{align*}

• The number of switchings is not bounded

• Compute the reachability relation for a strongly connected component

An algorithmic approach to stability verification of polyhedral switched systems. P. Prabhakar, M. G. Soto. ACC'14
Polyhedral switched systems contd.

Strongly connected component

\[ x_2 = x_1 + a_1 t_1 + a_2 t_2 + a_1 t_3 + a_2 t_4 + \ldots \]
\[ x_1 \in R, x_1 + a_1 t_1 \in R, x_1 + a_1 t_1 + a_2 t_2 \in R, \ldots \]

\[ x_2 = x_1 + a_1 t'_1 + a_2 t'_2 \]
\[ x_1 \in R, x_2 \in R \]
Polyhedral switched systems contd.

Strongly connected component

\[ x_2 = x_1 + a_1 t_1 + a_2 t_2 + a_1 t_3 + a_2 t_4 + \ldots \]
\[ x_1 \in R, x_1 + a_1 t_1 \in R, x_1 + a_1 t_1 + a_2 t_2 \in R, \ldots \]

\[ x_2 = x_1 + a_1 t_1' + a_2 t_2' \]
\[ x_1 \in R, x_2 \in R \]
Summary

- Can compute abstractions for PCD and polyhedral hybrid systems
- Quantitative predicate abstraction is sound for a general class of hybrid systems
- What happens if the abstraction fails to deduce stability?
- It returns a counter-example!
Validation: Counter-example Analysis
If the abstract system fails to prove stability, then it returns a counter-example.

A cycle with product of weight greater than 1

Need to check if it is spurious — can the system follow the cycle to exhibit trajectories which diverge

Validation — checking spuriousness — is not a bounded model-checking problem
Validation

**Theorem**

A counter example $p_1 \rightarrow p_2 \rightarrow p_3 \rightarrow \cdots \rightarrow p_1$ is valid if and only if

$$\exists \alpha > 1 : x_1 \xRightarrow{P_1} x_2 \xRightarrow{P_2} x_3 \cdots \xRightarrow{P_k} x_k \land x_k = \alpha x_1$$
}\exists \alpha > 1 : x_1 \xrightarrow{P_1} x_2 \xrightarrow{P_2} x_3 \ldots \xrightarrow{P_k} x_k \land x_k = \alpha x_1

\rightarrow \quad \alpha \rightarrow \quad \alpha^2 \rightarrow \quad \alpha^3

y_1 \quad \alpha y_1 \quad \alpha^2 y_1 \quad \alpha^3 y_1
Validation

$y_1 \rightarrow y_2 \rightarrow y_3 \rightarrow y_4$

Need not have a pair $y_{i+1} = \alpha y_i$ for $\alpha > 1$
Validation

\[ G : S \rightarrow 2^S \]

\[ y \mapsto Y \]

\[ \alpha y^* \in G(y^*) \]

Has some similarity with fix point

\[ y^* \mapsto y^* \]
Let $S$ be a non-empty, compact and convex set. Let $H$ be a set-valued function $S$ to $S$ such that
- its graph is a closed set
- $H(s)$ is non-empty and convex for all $s$ in $S$
Then $H$ has a fixpoint

$G : S \rightarrow 2^S$
$y \mapsto Y$

$G' : S \rightarrow 2^S$ given by $s \mapsto \frac{G(s)}{w}

S'$ with states from which there are infinite executions following the cycle
Validation and Refinement Summary

- If the counter-example is spurious, perform a backward propagation along the weights on the edges to compute the point of refinement.

- Refine as before by splitting the region at the point of refinement.

- Some improvements:
  - If an infinite execution (not necessarily diverging) does not exist, then can try to “eliminate” the cycle.
  - If infinite executions exist, but no diverging executions, then reduce the weight on some edge of the cycle.
Linear Hybrid Systems
Linear dynamical systems

\[
\begin{pmatrix}
\dot{x} \\
\dot{y}
\end{pmatrix} =
\begin{pmatrix}
a & b \\
c & d
\end{pmatrix}
\begin{pmatrix}
x \\
y
\end{pmatrix}
\]

A very important class of control system

\[
\text{Reach}(P_1, P, P_2) = \{(s_1, s_2) \mid s_1 \in P_1, s_2 \in P_2, s_1 \xrightarrow{P} s_2\}
\]

- Solution is an exponential function
- Need a representation on which optimization can be performed
- Approximation methods [Girard et al., Frehse et al., PP]
Switched Linear Systems

Figure 1: Phase portraits

Figure 2: Sample trajectories

Figure 3: Quantitative Abstractions

Figure 4: Annotated Abstract Graph for System M

Figure 5: Annotated Abstract Graph for System G

Figure 6: Graphical representation of the SpaceX
Arbitrary switching example

\[ V(x) = 19.576x_1^6 + 11.627x_1^5x_2 + 15.267x_1^4x_2 + 3.0857x_1^3x_2^3 + 8.9471x_1^2x_2^4 - 1.3629x_1x_2^5 + 1.0539x_2^6. \]
Hybridization for stability

\[ P = \{ Ax \mid x \in R \} \]

- Conical partitions do not ensure bounded error approximation of the reachability relation
- However, they ensure bounded error approximation of the scaling
Completeness for linear systems

Theorem
For every linear dynamical system that is asymptotically stable, there exists a polyhedral hybrid system abstraction that is asymptotically stable.

Proof Idea

\[ \dot{x} \in f(x) \quad \dot{x} \in g(x) \]

\[ d(f(x), g(x)) < \varepsilon \text{ implies } d(\sigma_f(x_0, t), \sigma_g(x_0, t)) \leq m(\varepsilon, T) \text{ for a time bound } T \]

Inspired from a classical result from differential inclusions theory, that states that if the Hausdorff distance between two differential inclusions is bounded by \( g \), then the solutions within time \( T \) are bounded by some exponential function of \( (g, T) \)
Completeness for linear systems

Theorem
For every linear dynamical system that is asymptotically stable, there exists a polyhedral hybrid system abstraction that is asymptotically stable.

Proof Idea
\[ \dot{x} \in f(x), \quad \dot{x} \in g(x) \]

\[ d(f(x), g(x)) < \epsilon \text{ implies } d(\sigma_f(x_0, t), \sigma_g(x_0, t)) \leq m(\epsilon, T) \text{ for a time bound } T \]

Polyhedral
\[ \dot{x} = Ax, \quad \dot{x} \in P \]
\[ P = \{ Ax \mid x \in R \} \]

Polyhedral-like
\[ \dot{x} = Ax, \quad \dot{x} \in P \| x \| \]
\[ d(f(x), g(x)) < \epsilon \| x \| \]
Proof continued ....

\[ \dot{x} = Ax \]
\[ \dot{x} \in P \]
\[ P = \{ Ax \mid x \in R \} \]

Polyhedral

Polyhedral-like

Polyhedral system stable iff Polyhedral-like system stable

If the linear system is asymptotically stable, then there exists a polyhedral-like system that is stable
If the linear system is asymptotically stable, then there exists a polyhedral-like system that is stable. Asymptotically stable linear systems are uniformly converging — choose the $\epsilon$ such that the error in the solutions between polyhedral-like and linear systems is bounded by $1/4$ for the time $T$ it takes for the trajectories of the linear system to be $1/2$ the distance where they started.
AVERIST: Algorithmic VERifier for STability

- Hybridization
- Polyhedral Hybrid Automaton
- Quantitative Predicate Abstraction
  - Model-Checking
  - Validation
  - Refinement

- Linear Hybrid Automaton
- Linear Optimization Solver (GLPK)
- Graph analyzer (NetworkX)
- SMT Solver (Z3)
- Parma Polyhedral Library
Experiments

Lyapunov’s method suffers from numerical instability

- 6th degree polynomial returned, but no 8th degree polynomial
- LF found for arbitrary switched system, but not for restricted switched system
- Common LF found, but no multiple LF

AVERIST

- Prove stability in many more cases than Stabhyli
- The verification time increases slower with respect to the number of regions as compared to the degree of the polynomial
- Abstraction computation is parallelizable
- Stabhyli can handle non-linear hybrid systems

<table>
<thead>
<tr>
<th>Dimension / name</th>
<th>Regions</th>
<th>Runtime</th>
<th>Proved Stability</th>
<th>Degree</th>
<th>LF found</th>
<th>Runtime</th>
</tr>
</thead>
<tbody>
<tr>
<td>2D</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AS1</td>
<td>129</td>
<td>31</td>
<td>Yes</td>
<td>6</td>
<td>Yes</td>
<td>8</td>
</tr>
<tr>
<td>SS4 1</td>
<td>9</td>
<td>&lt;1</td>
<td>Yes</td>
<td>8</td>
<td>–</td>
<td>452</td>
</tr>
<tr>
<td>SS8 1</td>
<td>17</td>
<td>&lt;1</td>
<td>Yes</td>
<td>6</td>
<td>–</td>
<td>443</td>
</tr>
<tr>
<td>SS16 1</td>
<td>33</td>
<td>1</td>
<td>Yes</td>
<td>4</td>
<td>–</td>
<td>177</td>
</tr>
<tr>
<td>3D</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AS 4</td>
<td>147</td>
<td>194</td>
<td>Yes</td>
<td>6</td>
<td>–</td>
<td>410</td>
</tr>
<tr>
<td>SS4 4</td>
<td>771</td>
<td>484</td>
<td>Yes</td>
<td>2</td>
<td>Yes</td>
<td>75</td>
</tr>
<tr>
<td>SS8 4</td>
<td>771</td>
<td>470</td>
<td>Yes</td>
<td>2</td>
<td>Yes</td>
<td>15</td>
</tr>
<tr>
<td>SS16 4</td>
<td>771</td>
<td>568</td>
<td>Yes</td>
<td>2</td>
<td>Yes</td>
<td>138</td>
</tr>
<tr>
<td>4D</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AS 7</td>
<td>81</td>
<td>625</td>
<td>Yes</td>
<td>2</td>
<td>–</td>
<td>12</td>
</tr>
<tr>
<td>SS4 7</td>
<td>81</td>
<td>119</td>
<td>Yes</td>
<td>2</td>
<td>–</td>
<td>101</td>
</tr>
<tr>
<td>SS8 7</td>
<td>153</td>
<td>234</td>
<td>Yes</td>
<td>2</td>
<td>–</td>
<td>1071</td>
</tr>
<tr>
<td>SS16 7</td>
<td>297</td>
<td>533</td>
<td>Yes</td>
<td>2</td>
<td>–</td>
<td>339</td>
</tr>
<tr>
<td>AS 9</td>
<td>–</td>
<td>out</td>
<td>No</td>
<td>4</td>
<td>Yes</td>
<td>34</td>
</tr>
<tr>
<td>SS4 9</td>
<td>81</td>
<td>125</td>
<td>Yes</td>
<td>4</td>
<td>–</td>
<td>105</td>
</tr>
<tr>
<td>SS8 9</td>
<td>153</td>
<td>247</td>
<td>Yes</td>
<td>2</td>
<td>–</td>
<td>16</td>
</tr>
</tbody>
</table>
Conclusion

- An algorithmic verification method for stability analysis based on abstraction-refinement and hybridization
- Works for polyhedral and linear hybrid systems
- Future Work: Non-linear systems and case studies
References

- Pre-orders for reasoning about stability.  
P. Prabhakar, G. E. Dullerud, M. Viswanathan. HSCC’12

- Pre-orders for reasoning about stability properties with respect to inputs of hybrid systems.  
P. Prabhakar, J. Liu, R. M. Murray. EMSOFT’13

- On the decidability of stability of hybrid systems.  
P. Prabhakar, M. Viswanathan. HSCC’13

- Abstraction based model-checking of stability of hybrid systems.  
P. Prabhakar, M. G. Soto. CAV’13

- An algorithmic approach to stability verification of polyhedral switched systems.  
P. Prabhakar, M. G. Soto. ACC’14

- Foundations for Quantative predicate abstraction for stability analysis of hybrid systems.  
P. Prabhakar, M. G. Soto. VMCAI’15

- Hybridization for stability analysis of switched linear systems.  
P. Prabhakar, M. G. Soto. HSCC’16